

# 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 v1.3, last revised 2011/08/06.

# Contents

<b>1 Identification</b>	<b>4</b>
<b>2 Generic helpers</b>	<b>4</b>
<b>3 Information, Warnings, and Errors</b>	<b>7</b>
<b>4 Using the keyval package</b>	<b>8</b>
<b>5 Margin resp. width</b>	<b>13</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>19</b>
<b>10 Label separators</b>	<b>20</b>
<b>11 Text formats</b>	<b>21</b>
<b>12 Fonts</b>	<b>21</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>25</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 . . . . .	28
19.3 The beamer class (Part one) . . . . .	30
19.4 The KOMA-Script classes . . . . .	31
19.5 The NTG Dutch classes . . . . .	32
19.6 The thesis class . . . . .	32

19.7	The frenchb Babel option . . . . .	33
19.8	The frenchle/pro package . . . . .	33
19.9	The hungarian and magyar Babel option . . . . .	34
19.10	Unknown document class (or package) . . . . .	34
<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>35</b>
<b>23</b>	<b>Types &amp; sub-types</b>	<b>40</b>
<b>24</b>	<b>subfig package adaption</b>	<b>50</b>

# 1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2011/08/06 v1.3 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@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.

```
20 \newcommand*\caption@ifinlist{%
21   \@expandtwoargs\caption@@ifinlist}
22 \newcommand*\caption@@ifinlist[2]{%
23   \begingroup
24   \def\@tempa##1, #1, ##2\@nil{%
25     \endgroup
26     \ifx\relax##2\relax
27       \expandafter\@secondoftwo
28     \else
29       \expandafter\@firstoftwo
30     \fi}%
31   \@tempa, #2, #1, \@nil}%
```

```

\caption@ifin@list \caption@ifin@list{\langle cmd\rangle}{\langle list entry\rangle}{\langle yes\rangle}{\langle no\rangle}
32 \newcommand*\caption@ifin@list[2]{%
33 \caption@ifempty@list#1%
34 {\@secondoftwo}%
35 {\@expandtwoargs\caption@@ifinlist{#2}{#1}}}

\caption@g@addto@list \caption@g@addto@list{\langle cmd\rangle}{\langle list entry\rangle}
36 \newcommand*\caption@g@addto@list[2]{%
37 \caption@ifempty@list#1{\gdef#1{#2}}{\g@addto@macro#1{, #2}}}

\caption@l@addto@list \caption@l@addto@list{\langle cmd\rangle}{\langle list entry\rangle}
38 \newcommand*\caption@l@addto@list[2]{%
39 \caption@ifempty@list#1{\def#1{#2}}{\l@addto@macro#1{, #2}}}

\caption@g@removefrom@list \caption@g@removefrom@list{\langle cmd\rangle}{\langle list entry\rangle}
40 \newcommand*\caption@g@removefrom@list[2]{%
41 \caption@l@removefrom@list#1{#2}%
42 \global\let#1#1}

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

\caption@for@list \caption@for@list{\langle cmd\rangle}{\langle code with #1\rangle}
45 \newcommand*\caption@for@list[2]{%
46 \caption@ifempty@list#1{}{%
47 \def\caption@tempb##1{#2}%
48 \@for\caption@tempa:=#1\do{%
49 \expandafter\caption@tempb\expandafter{\caption@tempa}}}}

\caption@ifempty@list \caption@ifempty@list{\langle cmd\rangle}{\langle true\rangle}{\langle false\rangle}
50 \newcommand*\caption@ifempty@list[1]{%
51 \ifx#1\@undefined
52 \expandafter\@firstoftwo
53 \else\ifx#1\relax
54 \expandafter\expandafter\expandafter\@firstoftwo
55 \else\ifx#1\empty
56 \expandafter\expandafter\expandafter\expandafter
57 \expandafter\expandafter\expandafter\expandafter\@firstoftwo
58 \else
59 \expandafter\expandafter\expandafter\expandafter
60 \expandafter\expandafter\expandafter\expandafter\@secondoftwo
61 \fi\fi\fi}

\caption@setbool For setting and testing boolean options we offer these three helper macros:
\caption@set@bool \caption@setbool{\langle name\rangle}{\langle value\rangle}
\caption@ifbool (with value = false/true/no/yes/off/on/0/1)
\caption@undefbool \caption@ifbool{\langle name\rangle}{\langle if-clause\rangle}{\langle else-clause\rangle}
\caption@undefbool{\langle name\rangle}

```

```

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

64 \newcommand*\caption@set@bool[2]{%
65   \caption@ifinlist{#2}{1,true,yes,on}{%
66     \let#1\@firstoftwo
67   }{\caption@ifinlist{#2}{0,false,no,off}{%
68     \let#1\@secondoftwo
69   }}%
70   \caption@Error{Undefined boolean value `#2'}%
71 }

72 \newcommand*\caption@ifbool[1]{\@nameuse{caption@if#1}}
73 \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>}
74 \newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}
75 \newcommand*\caption@teststar@[3]{%
76   \@ifstar{#1{#2}}{\caption@ifatletter{#1{#2}}{#1{#3}}}}
77 \AtBeginDocument{\let\caption@teststar@\caption@teststar}

78 \newcommand*\caption@ifatletter{%
79   \ifnum\the\catcode`\@=11
80     \expandafter\@firstoftwo
81   \else
82     \expandafter\@secondoftwo
83   \fi}
84 \AtBeginDocument{\let\caption@ifatletter\@secondoftwo}

\caption@withoptargs \caption@withoptargs{<cmd>}
85 \newcommand*\caption@withoptargs[1]{%
86   \@ifstar
87   {\def\caption@tempa{*}\caption@@withoptargs#1}%
88   {\def\caption@tempa{}\caption@@withoptargs#1}}
89 \def\caption@@withoptargs#1{%
90   \@ifnextchar[%]
91   {\caption@@@withoptargs#1}%
92   {\caption@@@withoptargs#1}}
93 \def\caption@@@withoptargs#1[#2]{%
94   \l@addto@macro\caption@tempa{[#2]}%
95   \caption@@withoptargs#1}
96 \def\caption@@@withoptargs#1{%
97   \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.)

```

```

98 \newcommand\caption@DoCheckCommand[2]{%
99   \begingroup
100     \let\@tempa#1%
101     #2%
102     \ifx\@tempa#1%
103       \endgroup
104       \let\caption@CheckCommand\@gobbletwo
105     \else
106       \endgroup
107     \fi}
108 \@onlypreamble\caption@DoCheckCommand

109 \let\caption@CheckCommand\caption@DoCheckCommand
110 \@onlypreamble\caption@CheckCommand

111 \newcommand*\caption@IfCheckCommand{%
112   \ifx\caption@CheckCommand\@gobbletwo
113     \let\caption@CheckCommand\caption@DoCheckCommand
114     \expandafter\@firstoftwo
115   \else
116     \expandafter\@secondoftwo
117   \fi}
118 \@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.

```

119 \let\caption@begindocumenthook\@empty
120 \let\caption@@begindocumenthook\@empty

121 \def\caption@AtBeginDocument{%
122   \caption@teststar@g@addto@macro
123   \caption@@begindocumenthook\caption@begindocumenthook}
124 %\@onlypreamble\caption@AtBeginDocument

125 \AtBeginDocument{%
126   \caption@InfoNoLine{Begin \noexpand\AtBeginDocument code}%

127   \def\caption@AtBeginDocument{%
128     \@ifstar{\g@addto@macro\caption@@begindocumenthook}\@firstofone}%
129   \caption@begindocumenthook
130   \let\caption@begindocumenthook\relax

131   \def\caption@AtBeginDocument{%
132     \@ifstar\@firstofone\@firstofone}%
133   \caption@@begindocumenthook
134   \let\caption@@begindocumenthook\relax

135   \caption@InfoNoLine{End \noexpand\AtBeginDocument code}}

```

### 3 Information, Warnings, and Errors

`\caption@Info`

`\caption@Info{<message>}`

```

136 \newcommand*\caption@Info[1]{%
137   \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.

```
138 \newcommand*\caption@InfoNoLine[1]{%
139   \PackageInfo{caption}{#1\@gobble}}
```

```
\caption@Warning \caption@Warning{message}
```

```
140 \newcommand*\caption@Warning[1]{%
141   \caption@WarningNoLine{#1\on@line}}
```

```
\caption@WarningNoLine \caption@WarningNoLine{message}
```

```
142 \newcommand*\caption@WarningNoLine[1]{%
143   \PackageWarning{caption}{#1.^J\caption@wh\@gobbletwo}}
144 \newcommand*\caption@wh{%
145   See the caption package documentation for explanation.}
```

```
\caption@Error \caption@Error{message}
```

```
146 \newcommand*\caption@Error[1]{%
147   \PackageError{caption}{#1}\caption@eh}
148 \newcommand*\caption@eh{%
149   If you do not understand this error, please take a closer look\MessageBreak
150   at the documentation of the 'caption' package, especially the\MessageBreak
151   section about errors.\MessageBreak\@ehc}
```

```
\caption@KV@err
```

```
152 \let\caption@KV@err\caption@Error
```

## 4 Using the keyval package

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

```
153 \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.

```
154 \providecommand*\undefine@key[2]{%
155   \@nameundef{KV@#1@#2}\@nameundef{KV@#1@#2@default}}
```

```
\@onlypreamble@key \onlypreamble@key{family}{key}
```

Analogous to \@onlypreamble from L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

```
156 \providecommand*\@preamble@keys{}
157 \providecommand*\@onlypreamble@key[2]{\@cons\@preamble@keys{{#1}{#2}}}
158 \@onlypreamble\@onlypreamble@key
159 \@onlypreamble\@preamble@keys
160 \providecommand*\@notprerr@key[1]{\KV@err{Can be used only in preamble}}
161 \caption@AtBeginDocument*{%
162   \def\@elt#1#2{\expandafter\let\csname KV@#1@#2\endcsname\@notprerr@key}%
163   \@preamble@keys
164   \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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>), so the `keyval` package is used. The starred form makes the option available during the lifetime of the current package only, so they can be used with `\usepackage`, but *not* with `\captionsetup` later on.

```

165 \newcommand*\DeclareCaptionOption{%
166   \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
167 \@onlypreamble\DeclareCaptionOption

168 \newcommand*\caption@declareoption[2]{%
169   #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
170 \@onlypreamble\caption@declareoption

```

```

\DeclareCaptionOptionNoValue \DeclareCaptionOptionNoValue{<option>}{<code>}
\DeclareCaptionOptionNoValue*{<option>}{<code>}

```

Same as `\DeclareCaptionOption` but issues an error if a value is given.

```

171 \newcommand*\DeclareCaptionOptionNoValue{%
172   \caption@teststar\caption@declareoption@novalue\AtEndOfPackage\@gobble}
173 \@onlypreamble\DeclareCaptionOptionNoValue

174 \newcommand\caption@declareoption@novalue[3]{%
175   \caption@declareoption{#1}{#2}[\KV@err]{%
176     \caption@option@novalue{#2}{#1}{#3}}
177 \@onlypreamble\caption@declareoption@novalue

178 \newcommand*\caption@option@novalue[2]{%
179   \ifx\KV@err#2%
180     \expandafter\@firstofone
181   \else
182     \KV@err{No value allowed for #1}%
183     \expandafter\@gobble
184   \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`.)

```

185 \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.

```

186 \newcommand*\captionsetup{%
187   \caption@teststar\@captionsetup\@gobble\@firstofone}

188 \newcommand*\@captionsetup[1]{%
189   \captionsetup@startrue#1\captionsetup@starfalse
190   \@ifnextchar[\caption@setup@options\caption@setup}

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

192 \def\caption@setup@options[#1]#2{%
193   \@bsphack

```

```

194 \ifcaptionsetup@star\captionsetup@starfalse\else\caption@addtooptlist{#1}\fi
195 \expandafter\caption@l@addto@list\csname caption@opt@#1\endcsname{#2}%
196 \@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.

```

197 \newcommand*\clearcaptionsetup{%
198   \caption@teststar@{\@clearcaptionsetup\@gobble\@firstofone}
199 \newcommand*\@clearcaptionsetup[1]{%
200   \let\caption@tempa#1%
201   \@testopt\@clearcaptionsetup{}}
202 \def\@clearcaptionsetup[#1]#2{%
203   \@bsphack
204   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
205   {\caption@tempa{\caption@Warning{Option list `#2' undefined}}}%
206   {\ifx,#1,%
207     \caption@clearsetup{#2}%
208     \else
209     \caption@@removefromsetup{#1}{#2}%
210     \fi}%
211   \@esphack}
212 \newcommand*\caption@clearsetup[1]{%
213   \caption@removefromoptlist{#1}%
214   \@nameundef{caption@opt@#1}}
215 \newcommand*\caption@removefromsetup{%
216   \let\caption@tempa\@gobble
217   \caption@@removefromsetup}
218 \newcommand*\caption@@removefromsetup[2]{%
219   \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
220   \expandafter\let\csname caption@opt@#2\endcsname\@undefined
221   \def\@tempb##1=##2\@nil{##1}%
222   \edef\@tempc{#1}%
223   \@for\@tempa:=\@tempa\do{%
224     \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
225     \ifx\@tempd\@tempc
226       \let\caption@tempa\@gobble
227     \else
228       \expandafter\expandafter\expandafter\caption@l@addto@list
229       \expandafter\csname caption@opt@#2\expandafter\endcsname
230       \expandafter{\@tempa}%
231     \fi}%
232   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
233   {\caption@removefromoptlist{#2}}}%
234   \caption@tempa{\caption@Warning{%
235     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>*.

```

236 \newcommand*\showcaptionsetup[2][\@firstofone]{%
237   \@bsphack
238   \GenericWarning{ }{%
239     #1 Caption Info: Option list on `#2'\MessageBreak
240     #1 Caption Data: \@ifundefined{caption@opt@#2}{%
241       -none-%
242     }{%
243       {\expandafter\expandafter\expandafter\strip@prefix
244         \expandafter\meaning\csname caption@opt@#2\endcsname}%
245     }%
246   \@esphack}

247 \DeclareCaptionOption{options}{\caption@setoptions{#1}}
248 \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.)

249 \newcommand*\caption@setoptions{%
250   \caption@teststar\caption@set@options\@gobble\@firstofone}

251 \newcommand*\caption@set@options[2]{%
252   \caption@Debug{options=#2}%
253   \expandafter\let\expandafter\caption@opt\csname caption@opt@#2\endcsname
254   \ifx\caption@opt\relax \else
255     \caption@xsetup\caption@opt
256     #1{\caption@clearsetup{#2}}% #1 = \@firstofone -or- \@gobble
257   \fi}

258 \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").

259 \newcommand*\caption@addtooptlist[1]{%
260   \@ifundefined{caption@opt@#1@lineno}{%
261     \caption@dooptlist\caption@g@addto@list{#1}%
262     \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
263   }{}}

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

267 \newcommand*\caption@dooptlist[2]{%
268   \begingroup
269     \edef\@tempa{#2}\@onelevel@sanitize\@tempa
270     \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
271   \endgroup}

272 \AtEndDocument{%
273   \caption@for@list\caption@optlist{%

```

```

274 \caption@WarningNoLine{%
275 Unused \string\captionsetup[#1]
276 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.
277 \newcommand*\caption@setkeys{\@dblarg\caption@@setkeys}

278 \long\def\caption@@setkeys[#1]#2#3{%
279 \@bsphack

280 \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
281 \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
282 \expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
283 \ifx\KV@err\relax
284 \def\KV@err##1{\PackageError{#1}{##1}{%
285 See the #1 package documentation for explanation.}}%
286 \fi
287 \let\KV@errx\KV@err
288 \edef\caption@keydepth{\caption@keydepth i}%
289 \caption@Debug{\protect\setkeys{#2}{#3}}%
290 \setkeys{#2}{#3}%
291 \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
292 \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
293 \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
294 \ifx\caption@keydepth\@empty \captionsetup@starfalse \fi
295 \@esphack}

296 \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 LATEX 2ε.
297 \newcommand*\caption@ExecuteOptions[2]{%
298 \expandafter\@expandtwoargs\csname caption@setkeys@#1\endcsname{#1}{#2}}%
299 \@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 LATEX 2ε. The starred variant do not process the global
options. (This code was taken from the hyperref package[2] v6.74 and improved.)
300 \newcommand*\caption@ProcessOptions{%
301 \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
302 \@onlypreamble\caption@ProcessOptions

303 \newcommand\caption@@ProcessOptions[2]{%
304 \let\@tempc\relax
305 \let\caption@tempa\@empty
306 #1{% \@firstofone -or- \@gobble
307 \@for\CurrentOption:=\@classoptionslist\do{%
308 \ifundefined{KV@#2@\CurrentOption}}{%
309 \ifundefined{KV@#2@\CurrentOption @default}{%
310 \PackageInfo{#2}{Global option '\CurrentOption' ignored}}%

```

```

311         }{%
312         \PackageInfo{#2}{Global option '\CurrentOption' processed}%
313         \edef\caption@tempa{\caption@tempa\CurrentOption,}%
314         \@expandtwoargs\@removeelement\CurrentOption
315         \@unusedoptionlist\@unusedoptionlist
316     }%
317 }%
318 }%
319 \let\CurrentOption\@empty
320 }%
321 \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
322 \AtEndOfPackage{\let\@unprocessedoptions\relax}
323 \@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.)

```

324 \newcommand*\caption@packagelist{}
325 \@onlypreamble\caption@packagelist
326 \newcommand\caption@SetupOptions[2]{%
327   \@namedef{caption@setkeys@#1}##1##2{#2}%
328   \expandafter\@onlypreamble\csname caption@setkeys@#1\endcsname
329   \@cons\caption@packagelist{{#1}}
330 \@onlypreamble\caption@SetupOptions
331 \let\caption@onefilewithoptions\@onefilewithoptions
332 \def\@onefilewithoptions#1[#2]{%
333   \begingroup
334   \def\@tempa{%
335     \endgroup
336     \caption@onefilewithoptions{#1}{{#2}}}%
337   \def\@tempb{#1}%
338   \def\@elt##1{%
339     \def\@tempc{##1}%
340     \ifx\@tempb\@tempc
341       \def\@tempa{%
342         \endgroup
343         \caption@ExecuteOptions{#1}{#2}%
344         \caption@onefilewithoptions{#1}[]}%
345     \fi}
346   \caption@packagelist
347   \@tempa}
348 \@onlypreamble\caption@onefilewithoptions

```

## 5 Margin resp. width

`\captionmargin` `\captionmargin` and `\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 *v1.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@`.

```

349 \newdimen\captionmargin
350 \newdimen\captionmargin@
351 \newdimen\captionwidth

352 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
353 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
354 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
355 \DeclareCaptionOption{width*}{\setcaptionwidth*{#1}}

356 \DeclareCaptionOption{calcmargin}{\caption@setcalcmargin{#1}}
357 \DeclareCaptionOption{calcmargin*}{\caption@setcalcmargin*{#1}}
358 \DeclareCaptionOption{calcwidth}{\caption@setcalcwidth{#1}}
359 \DeclareCaptionOption{calcwidth*}{\caption@setcalcwidth*{#1}}

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

362 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
363 \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.**

```

364 \newcommand*\setcaptionmargin{%
365   \caption@resetcalcmargin
366   \caption@setmargin}

367 \newcommand*\caption@setmargin{%
368   \caption@teststar\caption@@setmargin\@gobble\@firstofone}

369 \newcommand*\caption@@setmargin[2]{%
370   #1{\captionwidth\z@}%
371   \caption@@@setmargin#2,#2,\@nil}

372 \def\caption@@@setmargin#1,#2,#3\@nil{%
373   \setlength\captionmargin@{#2}%
374   \setlength\captionmargin{#1}%
375   \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.**

```

376 \newcommand*\setcaptionwidth{%
377   \caption@resetcalcmargin
378   \caption@setwidth}

379 \newcommand*\caption@setwidth{%
380   \caption@teststar\caption@@setwidth\@gobble\@firstofone}

381 \newcommand*\caption@@setwidth[2]{%
382   #1{\captionmargin\z@\captionmargin@\z@}%
383   \setlength\captionwidth{#2}%

```

`\caption@resetcalcmargin`

```

384 \newcommand*\caption@resetcalcmargin{%
385   \let\caption@calcmargin@hook\@empty}

```

`\caption@setcalcmargin`

```
386 \newcommand*\caption@setcalcmargin{%
387   \caption@teststar{\caption@@setcalcmargin\caption@setmargin}%
388   \@secondoftwo\@firstoftwo}

389 \newcommand*\caption@@setcalcmargin[3]{%
390   #2{\caption@resetcalcmargin
391     \l@addto@macro\caption@calcmargin@hook{#1{#3}}}%
392     {\l@addto@macro\caption@calcmargin@hook{#1*{#3}}}}
```

`\caption@setcalwidth`

```
393 \newcommand*\caption@setcalwidth{%
394   \caption@teststar{\caption@@setcalcmargin\caption@setwidth}%
395   \@secondoftwo\@firstoftwo}
```

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

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

397 \newcommand*\caption@stepcounter{%
398   \@tempcnta\caption@thecounter
399   \advance\@tempcnta\@ne
400   \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`.)

```
401 \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.

```
402 \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>X2<sub>ε</sub>. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```
403 \newcommand*\caption@label[1]{%
404   \caption@@label
405   \protected@write\@auxout{\let\caption@thepage\relax}%
406     {\string\caption@newlabel{#1}{\caption@thepage}}}

407 \newcommand*\caption@@label{%
408   \global\let\caption@@label\relax
409   \protected@write\@auxout{}%
410     {\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>X2<sub>ε</sub>. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```
411 \newcommand*\caption@pageref[1]{%
412   \expandafter\ifx\csname caption@r@#1\endcsname\relax
413     \G@refundefinedtrue % => 'There are undefined references.'
414     \@latex@warning{Reference '#1' on page \thepage \space undefined}%
415   \else
416     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
417   \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.

```
418 \newcommand*\caption@ifoddpage{%
419   \caption@iftwoside{%
420     \caption@label\caption@thecounter
421     \caption@pageref\caption@thecounter
422     \ifodd\caption@thepage
423       \let\caption@ifoddpage\@firstoftwo
424     \else
425       \let\caption@ifoddpage\@secondoftwo
426     \fi
427   }{\let\caption@ifoddpage\@firstoftwo}%
428   \caption@ifoddpage}
```

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

```
429 \newcommand*\caption@setoptcmd[2]{%
430   \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` 431 \newdimen\caption@indent  
`\caption@hangindent` 432 \newdimen\caption@parindent  
433 \newdimen\caption@hangindent

```
434 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
435   \setlength\caption@indent{#1}}
436 \DeclareCaptionOption{indentation}[\leftmargini]{%
437   \setlength\caption@indent{#1}}
438 \DeclareCaptionOption{parindent}{%
439   \setlength\caption@parindent{#1}}
440 \DeclareCaptionOption{hangindent}{%
441   \setlength\caption@hangindent{#1}}
442 \DeclareCaptionOption{parskip}{%
443   \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>

```
444 \providecommand*\caption@ifkomaclass{%
445   \ifundefined{scr@caption}\@gobble\@firstofone}
446 \@onlypreamble\caption@ifkomaclass
447 \caption@ifkomaclass{%
```

---

<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!)



```

448 \let\caption@KV@parindent\KV@caption@parindent
449 \DeclareCaptionOption{parindent}[]{%
450   \ifx,#1,%
451     \caption@Debug{Option 'parindent' ignored}%
452   \else
453     \caption@KV@parindent{#1}%
454   \fi}%
455 \let\caption@KV@parskip\KV@caption@parskip
456 \DeclareCaptionOption{parskip}[]{%
457   \ifx,#1,%
458     \caption@Debug{Option 'parskip' ignored}%
459   \else
460     \caption@KV@parskip{#1}%
461   \fi}%
462 }

```

## 7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
463 \newcommand*\DeclareCaptionStyle[1]{%
464   \@testopt{\caption@declarestyle{#1}}{}%
465   \@onlypreamble\DeclareCaptionStyle
466   \def\caption@declarestyle#1[#2]#3{%
467     \global\@namedef{caption@sls@#1}{#2}%
468     \global\@namedef{caption@sty@#1}{#3}%
469     \@onlypreamble\caption@declarestyle
470   }
471   \DeclareCaptionOption{style*}{\caption@setstyle*{#1}}
472   \DeclareCaptionOption{singlelinecheck}[1]{\caption@set@bool\caption@ifslc{#1}}
473   \DeclareCaptionOption{slc}[1]{\KV@caption@singlelinecheck{#1}}

```

```

\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.

```

474 \newcommand*\caption@setstyle{%
475   \caption@teststar\caption@@setstyle\@gobble\@firstofone}
476 \newcommand*\caption@@setstyle[2]{%
477   \@ifundefined{caption@sty@#2}%
478     {#1{\caption@Error{Undefined style '#2'}}}%
479     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
480     \ifx\caption@setstyle@flag\@undefined
481       \let\caption@setstyle@flag\relax
482       \caption@resetstyle
483       \caption@xsetup\caption@sty
484       \let\caption@setstyle@flag\@undefined
485     \else

```

```

486      \caption@xsetup\caption@sty
487      \fi
488      \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
489      \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
490      \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!*

```

491 \newcommand*\caption@resetstyle{%
492   \caption@setup{%
493     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
494     justification=justified,font=,size=,labelfont=,textfont=,%
495     margin=0pt,minmargin=0,maxmargin=0,%
496     indent=0pt,parindent=0pt,hangindent=0pt,%
497     slc,rule,strut}%
498   \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.

```

499 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
500 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
501   format=default,labelsep=default,textformat=default,%
502   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.

```

503 \newcommand*\DeclareCaptionFormat{%
504   \caption@teststar\caption@declareformat\@gobble\@firstofone}
505 \onlypreamble\DeclareCaptionFormat

506 \newcommand*\caption@declareformat[2]{%
507   \@dblarg{\caption@@declareformat#1{#2}}
508 \onlypreamble\caption@declareformat

509 \long\def\caption@@declareformat#1#2[#3]#4{%
510   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
511   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
512   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
513 \onlypreamble\caption@@declareformat

514 \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`).

```

515 \newcommand*\caption@setformat[1]{%
516   \ifundefined{caption@fmt@#1}%

```

```

517     {\caption@Error{Undefined format ‘#1’}}%
518     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
519      \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
520      \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

clareCaptionDefaultFormat

```

521 \newcommand*\DeclareCaptionDefaultFormat[1]{%
522   \expandafter\def\expandafter\caption@fmt@default\expandafter
523     {\csname caption@fmt@#1\endcsname}%
524   \expandafter\def\expandafter\caption@slfmt@default\expandafter
525     {\csname caption@slfmt@#1\endcsname}%
526   \expandafter\def\expandafter\caption@ifh@default\expandafter
527     {\csname caption@ifh@#1\endcsname}}
528 \@onlypreamble\DeclareCaptionDefaultFormat

```

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

```

529 \DeclareCaptionFormat{plain}{#1#2#3\par}
530 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
531   \caption@ifin@list\caption@lsepclist\caption@lsepname
532   {\caption@Error{%
533     The option ‘labelsep=\caption@lsepname’ does not work\MessageBreak
534     with ‘format=hang’}}%
535   {\@hangfrom{#1#2}%
536     \advance\caption@parindent\hangindent
537     \advance\caption@hangindent\hangindent
538     \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

539 \DeclareCaptionDefaultFormat{plain}

```

## 9 Label formats

```

DeclareCaptionLabelFormat \DeclareCaptionLabelFormat{<name>}{<code with #1 and #2>}
540 \newcommand*\DeclareCaptionLabelFormat[2]{%
541   \global\@namedef{caption@lfmt@#1}##1##2{#2}}
542 \@onlypreamble\DeclareCaptionLabelFormat

543 \DeclareCaptionOption{labelformat}{\caption@setlabelformat{#1}}

```

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

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

```

544 \newcommand*\caption@setlabelformat[1]{%
545   \@ifundefined{caption@lfmt@#1}%
546   {\caption@Error{Undefined label format ‘#1’}}%
547   {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}

```

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

```

548 \DeclareCaptionLabelFormat{empty}{}
549 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}{#2}}
550 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace}{#2}}
551 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace}{#2}}

```

‘default’ usually maps to ‘simple’.

```

552 \def\caption@lfmt@default{\caption@lfmt@simple}

```

## 10 Label separators

```

\DeclareCaptionLabelSeparator{\DeclareCaptionLabelSeparator{<name>}{<code>}
\DeclareCaptionLabelSeparator*{<name>}{<code>}
The starred form causes the label separator to be typeset without using \captionlabelfont.

553 \newcommand\DeclareCaptionLabelSeparator{%
554   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
555 \@onlypreamble\DeclareCaptionLabelSeparator

556 \newcommand\caption@declarelabelseparator[3]{%
557   \global\@namedef{caption@iflfl@#2}{#1}%
558   \global\long\@namedef{caption@lsep@#2}{#3}%
559   \caption@@declarelabelseparator{#2}{#3}\@nil}
560 \@onlypreamble\caption@declarelabelseparator

561 \long\def\caption@@declarelabelseparator#1#2\@#3\@nil{%
562   \def\@tempa{#3}\ifx\@tempa\@empty \else
563     \caption@g@addto@list\caption@lsep@list{#1}%
564   \fi}
565 \@onlypreamble\caption@@declarelabelseparator

566 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
567 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}

caption@setlabelseparator \caption@setlabelseparator{<name>}
Selecting a caption label separator simply means saving the code (in \caption@lsep).

568 \newcommand*\caption@setlabelseparator[1]{%
569   \@ifundefined{caption@lsep@#1}%
570     {\caption@Error{Undefined label separator `#1'}}%
571     {\edef\caption@lsepname{#1}%
572       \expandafter\let\expandafter\caption@iflfl\csname caption@iflfl@#1\endcsname
573       \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’.

574 \DeclareCaptionLabelSeparator{none}{}
575 \DeclareCaptionLabelSeparator{colon}{: }
576 \DeclareCaptionLabelSeparator{period}{. }
577 \DeclareCaptionLabelSeparator{space}{ }
578 \DeclareCaptionLabelSeparator*{quad}{\quad}
579 \DeclareCaptionLabelSeparator*{newline}{\}
580 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

caption@setdefaultlabelsep

581 \newcommand*\caption@setdefaultlabelsep[1]{%
582   \ifx\caption@lsep\caption@lsep@default
583     \caption@set@default@labelsep{#1}%
584     \caption@setlabelseparator{default}%
585   \else
586     \caption@set@default@labelsep{#1}%
587   \fi}

588 \newcommand*\caption@set@default@labelsep[1]{%
589   \def\caption@lsep@default{\@nameuse{caption@lsep@#1}}%
590   \def\caption@iflfl@default{\@nameuse{caption@iflfl@#1}}%

```

‘default’ usually maps to ‘colon’.

```
591 \caption@set@default@labelsep{colon}
```

## 11 Text formats

```
\DeclareCaptionTextFormat \DeclareCaptionTextFormat{<name>}{<code with #1>}
592 \newcommand*\DeclareCaptionTextFormat[2]{%
593   \global\long\@namedef{caption@tfmt@#1}##1{#2}}
594 \@onlypreamble\DeclareCaptionTextFormat

595 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
596 \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).
597 \newcommand*\caption@settextformat[1]{%
598   \ifundefined{caption@tfmt@#1}%
599     {\caption@Error{Undefined text format `#1'}}%
600     {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}
```

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

```
601 \DeclareCaptionTextFormat{simple}{#1}
602 \DeclareCaptionTextFormat{period}{#1.}
```

‘default’ usually maps to ‘simple’.

```
603 \def\caption@tfmt@default{\caption@tfmt@simple}
```

## 12 Fonts

```
\DeclareCaptionFont \DeclareCaptionFont{<name>}{<code>}
604 \newcommand*\DeclareCaptionFont[2]{%
605   \define@key{caption@fnt}{#1}[]{\l@addto@macro\caption@fnt{#2}}
606 \@onlypreamble\DeclareCaptionFont

\DeclareCaptionDefaultFont \DeclareCaptionDefaultFont{<name>}{<code>}
607 \newcommand*\DeclareCaptionDefaultFont[2]{%
608   \global\@namedef{caption#1@default}{#2}}
609 \@onlypreamble\DeclareCaptionDefaultFont

610 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
611 \DeclareCaptionOption{font+}{\caption@addtofont{font}{#1}}
612 \DeclareCaptionDefaultFont{font}{}

613 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
614 \DeclareCaptionOption{labelfont+}{\caption@addtofont{labelfont}{#1}}
615 \DeclareCaptionDefaultFont{labelfont}{}

616 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
617 \DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
618 \DeclareCaptionDefaultFont{textfont}{}%
```

`\caption@setfont` `\caption@setfont{<name>}{<keyval-list of names>}`  
**Selecting a caption font means saving all the code snippets in `\caption<name>`.**

```
619 \newcommand*\caption@setfont[1]{%
620   \expandafter\let\csname caption#1\endcsname\@empty
621   \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.**

```
622 \newcommand*\caption@addtofont[2]{%
623   \begingroup
624     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
625     \define@key{caption@fnt}{default}[]{%
626       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
627     \caption@setkeys[caption]{caption@fnt}{#2}%
628     \global\let\caption@tempa\caption@fnt
629   \endgroup
630   \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`.**

```
631 \newcommand*\caption@font{%
632   \caption@teststar\caption@@font\@firstofone
633   {\caption@setkeys[caption]{caption@fnt}}%
634 \newcommand*\caption@@font[2]{%
635   \begingroup
636   \def\caption@fnt{\endgroup}%
637   #1{#2}%
638   \caption@fnt}
```

**These are the pre-defined font code snippets.**

```
639 \DeclareCaptionFont{normalcolor}{\normalcolor}
640 \DeclareCaptionFont{color}{\color{#1}}

641 \DeclareCaptionFont{normalfont}{\normalfont}
642 \DeclareCaptionFont{up}{\upshape}
643 \DeclareCaptionFont{it}{\itshape}
644 \DeclareCaptionFont{sl}{\slshape}
645 \DeclareCaptionFont{sc}{\scshape}
646 \DeclareCaptionFont{md}{\mdseries}
647 \DeclareCaptionFont{bf}{\bfseries}
648 \DeclareCaptionFont{rm}{\rmfamily}
649 \DeclareCaptionFont{sf}{\sffamily}
650 \DeclareCaptionFont{tt}{\ttfamily}

651 \DeclareCaptionFont{scriptsize}{\scriptsize}
652 \DeclareCaptionFont{footnotesize}{\footnotesize}
653 \DeclareCaptionFont{small}{\small}
654 \DeclareCaptionFont{normalsize}{\normalsize}
655 \DeclareCaptionFont{large}{\large}
656 \DeclareCaptionFont{Large}{\Large}

657 \DeclareCaptionFont{sansmath}{\sansmath}
```

```

658 \DeclareCaptionFont{singlespacing}{%
659   \@ifundefined{setspace@singlespace}{}{%
660     \setstretch\setspace@singlespace}% normally 1
661 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
662 \DeclareCaptionFont{doublespacing}{\doublespacing}
663 \DeclareCaptionFont{stretch}{\setstretch{#1}}

664 %\DeclareCaptionFont{normal}{%
665 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
666 \DeclareCaptionFont{normal}{%
667   \caption@font*{%
668     \KV@caption@fnt@normalcolor\@unused
669     \KV@caption@fnt@normalfont\@unused
670     \KV@caption@fnt@normalsize\@unused
671     \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.

```

672 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
673 \DeclareCaptionDefaultFont{size}{}

```

## 13 Justifications

```

\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
674 \newcommand*\DeclareCaptionJustification[2]{%
675   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v1.0
676   \DeclareCaptionFont{#1}{#2}}
677 \@onlypreamble\DeclareCaptionJustification

\captionDefaultJustification \DeclareCaptionDefaultJustification{<code>}
678 \newcommand*\DeclareCaptionDefaultJustification[1]{%
679   \global\@namedef{caption@hj@default}{#1}% for compatibility to v1.0
680   \DeclareCaptionDefaultFont{@hj}{#1}}
681 \@onlypreamble\DeclareCaptionDefaultJustification

682 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
683 \DeclareCaptionDefaultJustification{}

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

```

These are the pre-defined justification code snippets.

```

685 \DeclareCaptionJustification{justified}{}
686 \DeclareCaptionJustification{centering}{\centering}
687 \DeclareCaptionJustification{centerfirst}{\centerfirst}
688 \DeclareCaptionJustification{centerlast}{\centerlast}
689 \DeclareCaptionJustification{raggedleft}{\raggedleft}
690 \DeclareCaptionJustification{raggedright}{\raggedright}

```

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

```

691 \providecommand\centerfirst{%
692   \let\\\@centercr

```

```

693 \edef\caption@normaladjust{%
694   \leftskip\the\leftskip
695   \rightskip\the\rightskip
696   \parfillskip\the\parfillskip\relax}%
697 \leftskip\z@\@plus -1fil%
698 \rightskip\z@\@plus 1fil%
699 \parfillskip\z@skip
700 \noindent\hskip\z@\@plus 2fil%
701 \@setpar{\@@par\@restorepar\caption@normaladjust}}

```

\centerlast This is based on code from Anne Brüggemann-Klein[1]

```

702 \providecommand\centerlast{%
703   \let\\\@centercr
704   \leftskip\z@\@plus 1fil%
705   \rightskip\z@\@plus -1fil%
706   \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.

```

707 \DeclareCaptionJustification{Centering}{%
708   \caption@ragged\Centering\centering}
709 \DeclareCaptionJustification{RaggedLeft}{%
710   \caption@ragged\RaggedLeft\raggedleft}
711 \DeclareCaptionJustification{RaggedRight}{%
712   \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.

```

713 \newcommand*\caption@ragged{%
714   \caption@Debug{We need ragged2e}%
715   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
716   \global\let\caption@ragged\caption@@ragged
717   \caption@ragged}

718 \caption@AtBeginDocument{%
719   \@ifundefined{caption@r@ragged2e}{%
720     \newcommand*\caption@@ragged{%
721       \caption@Warning{%
722         'ragged2e' support has been changed.\MessageBreak
723         Rerun to get captions right}%
724     \global\let\caption@ragged\@secondoftwo % suppress further warnings
725     \caption@ragged}%
726   }{%
727     \caption@Debug{We load ragged2e}%
728     \IfFileExists{ragged2e.sty}{%

```



```

729     \RequirePackage{ragged2e}%
730     \let\caption@@ragged\@firstoftwo
731   }{%
732     \newcommand*\caption@@ragged[2]{%
733       \@ifundefined{caption\string#1}{%
734         \caption@Warning{%
735           'ragged2e' package not loaded, therefore\MessageBreak
736             substituting \string#2 for \string#1\MessageBreak}%
737       \global\@namedef{caption\string#1}}}%
738     #2}%
739   }%
740 }

```

## 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`

```

741 \@ifundefined{abovecaptionskip}{%
742   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
743 \@ifundefined{belowcaptionskip}{%
744   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

745 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
746 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
747 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

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

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

```

748 \newcommand*\caption@rule{\caption@ifrule\caption@hrule\relax}
749 \newcommand*\caption@hrule{\hrule\@height\z@}

750 \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.

```

751 \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.*

```

752 \newcommand*\caption@setposition[1]{%
753   \caption@ifinlist{#1}{d,default}}{%
754     \let\caption@position\caption@defaultpos
755   }{\caption@ifinlist{#1}{t,top,above}}{%

```

```

756 \let\caption@position\@firstoftwo
757 }{\caption@ifinlist{#1}{b,bottom,below}{%
758 \let\caption@position\@secondoftwo
759 }{\caption@ifinlist{#1}{a,auto}{%
760 \let\caption@position\@undefined
761 }{%
762 \caption@Error{Undefined position '#1'}%
763 }}}}}

```

`\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’ correspondents to the `\@makecaption` implementation in the standard L<sup>A</sup>T<sub>E</sub>X document classes, but ‘auto’ should give better results in most cases.

```

764 %\caption@setdefaultpos{a}% default = auto
765 \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.)

```

766 \newcommand*\caption@iftop{%
767 \ifx\caption@position\@undefined
768 \let\caption@position\@secondoftwo
769 % = \caption@setposition b%
770 \fi
771 \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.

```

772 \newcommand*\caption@fixposition{%
773 \ifx\caption@position\@undefined
774 \caption@autoposition
775 \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.

```

776 \newcommand*\caption@autoposition{%
777 \ifvmode
778 \edef\caption@tempa{\the\prevdepth}%
779 \caption@Debug{\protect\prevdepth=\caption@tempa}%
780 \ifdim\prevdepth>-\p@
781 \let\caption@position\@secondoftwo
782 \else
783 \let\caption@position\@firstoftwo
784 \fi
785 % = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%

```

```

786 \else
787   \caption@Debug{no \protect\prevdepth}%
788   \let\caption@position\@secondoftwo
789 %   = \caption@setposition b%
790 \fi}

```

```

\caption@setautoposition \caption@setautoposition{<position>}
replaces the above algorithm by a different one (or a fixed position setting).
791 \newcommand*\caption@setautoposition[1]{%
792   \def\caption@autoposition{\caption@setposition{#1}}

```

## 16 Hooks

```

\AtBeginCaption \AtBeginCaption {<code>}
\AtEndCaption \AtEndCaption {<code>}
These hooks can be used analogous to \AtBeginDocument and \AtEndDocument.
793 \newcommand*\caption@beginhook{}
794 \newcommand*\caption@endhook{}
795 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
796 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

## 17 Lists

```

797 \DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
798 \DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}
\caption@setlist \caption@setlist{<boolean>}
799 \newcommand*\caption@setlist{\caption@set@bool\caption@iflist}
\DeclareCaptionListFormat \DeclareCaptionListFormat{<name>}{<code with #1 and #2>}
800 \newcommand*\DeclareCaptionListFormat[2]{%
801   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
802 \@onlypreamble\DeclareCaptionListFormat
803 \DeclareCaptionOption{listformat}{\caption@setlistformat{#1}}
\caption@setlistformat \caption@setlistformat{<name>}
Selecting a caption list format simply means saving the code (in \caption@lstfmt).
804 \newcommand*\caption@setlistformat[1]{%
805   \ifundefined{caption@lstfmt@#1}%
806     {\caption@Error{Undefined list format `#1'}}%
807     {\expandafter\let\expandafter\caption@lstfmt
808       \csname caption@lstfmt@#1\endcsname}}

```

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

```

809 \DeclareCaptionListFormat{empty}{}
810 \DeclareCaptionListFormat{simple}{#1#2}
811 \DeclareCaptionListFormat{parens}{#1 (#2)}
812 \DeclareCaptionListFormat{subsimple}{#2}
813 \DeclareCaptionListFormat{subparens}{(#2)}

```

tion@setdefaultlistformat

```
814 \newcommand*\caption@setdefaultlistformat[1]{%
815   \ifx\caption@lstfmt\caption@lstfmt@default
816     \caption@set@default@listformat{#1}%
817     \caption@setlistformat{default}%
818   \else
819     \caption@set@default@listformat{#1}%
820   \fi}

821 \newcommand*\caption@set@default@listformat[1]{%
822   \def\caption@lstfmt@default{\@nameuse{caption@lstfmt@#1}}}
```

‘default’ usually maps to ‘subsimple’.

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

## 18 Debug option

```
824 \DeclareCaptionOption{debug}[1]{%
825   \caption@set@bool\caption@ifdebug{#1}%
826   \caption@ifdebug
827     {\let\caption@Debug\caption@Info}%
828     {\let\caption@Debug\@gobble}}

829 \DeclareOption{debug}{\setkeys{caption}{debug}}

830 \setkeys{caption}{debug=0}
```

## 19 Document classes & Babel support

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

```
831 \caption@CheckCommand\@makecaption{%
832   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
833   \long\def\@makecaption#1#2{%
834     \vskip\abovecaptionskip
835     \sbox\@tempboxa{#1: #2}%
836     \ifdim \wd\@tempboxa >\hsize
837       #1: #2\par
838     \else
839       \global \@minipagefalse
840       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
841     \fi
842     \vskip\belowcaptionskip}}
```

### 19.2 The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes

\caption@ifamsclass

```
843 \providecommand*\caption@ifamsclass{%
844   \@ifundefined{captionheadfont}\@gobble\@firstofone}
845 \@onlypreamble\caption@ifamsclass

846 \caption@ifamsclass{%
847   \caption@CheckCommand\@makecaption{%
848     % amsart|amsproc|amsbook [2004/08/06 v2.20]
849     \long\def\@makecaption#1#2{%
```

```

850 \setbox\@tempboxa\vbox{\color@setgroup
851 \advance\hsize-2\captionindent\noindent
852 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
853 {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
854 \unskip\kern-2\captionindent\par
855 \global\setbox\@ne\lastbox\color@endgroup}%
856 \ifhbox\@ne % the normal case
857 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
858 \fi
859 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
860 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
861 \else % tempboxa contained more than one line
862 \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
863 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
864 \fi
865 \ifnum\@tempcnta<64 % if the float IS a figure...
866 \addvspace\abovecaptionskip
867 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
868 \else % if the float IS NOT a figure...
869 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
870 \nobreak
871 \vskip\belowcaptionskip
872 \fi
873 \relax
874 }}

875 \caption@CheckCommand\makecaption{%
876 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
877 \long\def\makecaption#1#2{%
878 \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
879 \setbox\@tempboxa\vbox{\color@setgroup
880 \advance\hsize-2\captionindent\noindent
881 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
882 {\@cdr#2\@nil}{.\@addpunct{.}\@captionfont\upshape\enspace#2}%
883 \unskip\kern-2\captionindent\par
884 \global\setbox\@ne\lastbox\color@endgroup}%
885 \ifhbox\@ne % the normal case
886 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
887 \fi
888 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
889 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
890 \@tempdima\wd\@ne\advance\@tempdima-\captionindent
891 \wd\@ne\@tempdima
892 \else % tempboxa contained more than one line
893 \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
894 \unvbox\@tempboxa\parskip\z@skip
895 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
896 \fi
897 \ifnum\@tempcnta<64 % if the float IS a figure...
898 \addvspace\abovecaptionskip
899 \noindent\kern\captionindent\box\@ne
900 \else % if the float IS NOT a figure...
901 \noindent\kern\captionindent\box\@ne
902 \nobreak
903 \vskip\belowcaptionskip

```

```

904     \fi
905     \relax
906   }}

907   \let\captionmargin\captionindent % set to 3pc by AMS class
908   \begingroup\edef\@tempa{\endgroup
909     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
910       {margin=\the\captionmargin
911         \@ifundefined{smf@makecaption}{{},{,maxmargin=.1\linewidth}}}}
912   \@tempa
913   \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
914   \DeclareCaptionLabelSeparator{default}{{.\enspace}
915   \DeclareCaptionDefaultFont{font}{\@captionfont}
916   \DeclareCaptionDefaultFont{labelfont}{\@captionheadfont}
917   \DeclareCaptionDefaultFont{textfont}{\@captionfont\upshape}
918   \captionsetup[figure]{position=b}
919   \captionsetup[table]{position=t}

920 }

```

### 19.3 The beamer class (Part one)

\caption@ifbeamerclass

```

921 \providecommand*\caption@ifbeamerclass{%
922   \@ifclassloaded{beamer}\@firstofone\@gobble}
923 \@onlypreamble\caption@ifbeamerclass

924 \caption@ifbeamerclass{%
925   \caption@CheckCommand\beamer@makecaption{%
926     % beamerbaselocalstructure.sty, v 1.53 2007/01/28 20:48:21 tantau
927     \long\def\beamer@makecaption#1#2{%
928       \def\insertcaptionname{\csname#1name\endcsname}%
929       \def\insertcaptionnumber{\csname the#1\endcsname}%
930       \def\insertcaption{#2}%
931       \nobreak\vskip\abovecaptionskip\nobreak
932       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
933       \ifdim \wd\@tempboxa >\hsize
934         \usebeamertemplate**{caption}\par
935       \else
936         \global \@minipagefalse
937         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
938       \fi
939       \nobreak\vskip\belowcaptionskip\nobreak}}

```

\caption@ifbeamertemplate

```

940 \newcommand*\caption@ifbeamertemplate[1]{%
941   \begingroup
942     \let\beamer@@tmpl@caption@ORI\beamer@@tmpl@caption
943     \@nameuse{beamer@@tmpop@caption@#1}%
944     \ifx\beamer@@tmpl@caption@ORI\beamer@@tmpl@caption
945       \endgroup\expandafter\@firstoftwo
946     \else
947       \endgroup\expandafter\@secondoftwo
948     \fi}

```

```

949 \DeclareCaptionLabelFormat{default}{%
950   #1\caption@ifbeamertemplate{numbered}{~#2}{}}
951 \caption@declarelabeleseparator
952 {\caption@ifbeamertemplate{caption name own line}\@gobble\@firstofone}
953 {default}
954 {\caption@ifbeamertemplate{caption name own line}{\}\{:\ }}
955 \DeclareCaptionDefaultFont{font}{%
956   \usebeamerfont*{caption}%
957   \usebeamercolor[fg]{caption}}
958 \DeclareCaptionDefaultFont{labelfont}{%
959   \usebeamercolor[fg]{caption name}%
960   \usebeamerfont*{caption name}}
961 \DeclareCaptionDefaultJustification{\raggedright}
962 \DeclareOption{beamerclass}{%
963   \renewcommand\caption@ifslc{%
964     \caption@ifbeamertemplate{caption name own line}\@secondoftwo\@firstoftwo}
965   % Since the beamer class do not offer a 'list of figures' we switch this supp
966   \captionsetup{list=0}}
967 \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.)

```

968 \defbeamertemplate{caption}{caption3}{%
969   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
970 \DeclareOption{beamer}{%
971   % \usebeamertemplate*{caption} will set font
972   \DeclareCaptionDefaultFont{font}{}%
973   \setbeamertemplate{caption}[caption3]}
974 %
975 % \begin{macrocode]
976 }

```

## 19.4 The KOMA-Script classes

`\caption@ifkomaclass`

```

977 \providecommand*\caption@ifkomaclass{%
978   \@ifundefined{scr@caption}\@gobble\@firstofone}
979 \@onlypreamble\caption@ifkomaclass

980 \caption@ifkomaclass{%
981   \caption@CheckCommand\@makecaption{%
982     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
983     \long\def\@makecaption#1#2{%
984       \if@captionabove
985         \vskip\belowcaptionskip
986       \else
987         \vskip\abovecaptionskip
988       \fi
989       \@@makecaption\@firstofone{#1}{#2}%
990       \if@captionabove
991         \vskip\abovecaptionskip
992       \else

```

```

993         \vskip\belowcaptionskip
994     \fi}}

995 \DeclareCaptionFormat{default}[#1#2#3\par]{%
996     \ifdofullc@p
997         \caption@ifin@list\caption@lsepclist\caption@lsepname
998         {\caption@Error{%
999             The option 'labelsep=\caption@lsepname' does not work\MessageBreak
1000             with \noexpand\setcaphanging (which is set by default)}}%
1001         {\caption@fmt@hang{#1}{#2}{#3}}%
1002     \else
1003         #1#2%
1004         \ifdim\cap@indent<\z@
1005             \par
1006             \noindent\hspace*{ -\cap@indent}%
1007         \else\if@capbreak
1008             \par
1009         \fi\fi
1010         #3\par
1011     \fi}
1012 \DeclareCaptionLabelSeparator{default}{\captionformat}
1013 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
1014 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
1015 }

```

## 19.5 The NTG Dutch classes

\caption@ifntgclass

```

1016 \providecommand*\caption@ifntgclass{%
1017     \@ifundefined{CaptionFonts}\@gobble\@firstofone}
1018 \onlypreamble\caption@ifntgclass

1019 \caption@ifntgclass{%
1020     \caption@CheckCommand\@makecaption{%
1021         % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
1022         \long\def\@makecaption#1#2{%
1023             \vskip\abovecaptionskip
1024             \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
1025             \ifdim \wd\@tempboxa >\hsize
1026                 {\CaptionLabelFont#1:} \CaptionTextFont#2\par
1027             \else
1028                 \global \@minipagefalse
1029                 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1030             \fi
1031             \vskip\belowcaptionskip}}
1032 \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
1033 \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
1034 }

```

## 19.6 The thesis class

\caption@ifthesisclass



```

1035 \providecommand*\caption@ifthesisclass{%
1036   \@ifundefined{cph@font}{\@gobble}{\@ifundefined{cpb@font}{\@gobble\@firstofone}}
1037 \caption@ifthesisclass{%
1038   \caption@CheckCommand\@makecaption{%
1039     % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
1040     \long\def\@makecaption#1#2{%
1041       \vskip\abovecaptionskip
1042       \setbox\@tempboxa\hbox{{\cph@font #1:} {\cpb@font #2}}%
1043       \ifdim \wd\@tempboxa >\hsize
1044         \hangfrom{\cph@font #1:} {\cpb@font #2\par}%
1045       \else
1046         \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
1047       \fi
1048       \vskip\belowcaptionskip}}
1049 \DeclareCaptionDefaultFormat{hang}
1050 \DeclareCaptionDefaultFont{labelfont}{\cph@font}
1051 \DeclareCaptionDefaultFont{textfont}{\cpb@font}
1052 }

```

## 19.7 The frenchb Babel option

```

1053 \@ifundefined{FB@makecaption}{}{%
1054   \caption@CheckCommand\@makecaption{%
1055     % frenchb.1df [2005/02/06 v1.6g French support from the babel system]
1056     % frenchb.1df [2007/10/05 v2.0e French support from the babel system]
1057     \long\def\@makecaption#1#2{%
1058       \vskip\abovecaptionskip
1059       \sbox\@tempboxa{#1\CaptionSeparator #2}%
1060       \ifdim \wd\@tempboxa >\hsize
1061         #1\CaptionSeparator #2\par
1062       \else
1063         \global \@minipagefalse
1064         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1065       \fi
1066       \vskip\belowcaptionskip}}
1067 \ifx\@makecaption\STD@makecaption
1068   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
1069   \def\caption@frenchb{% supress frenchb warning
1070     \let\STD@makecaption\@makecaption
1071     \let\FB@makecaption\@makecaption}
1072 \else
1073   \ifx\@makecaption\@undefined\else
1074     \caption@InfoNoLine{%
1075       The definition of \protect\@makecaption\space
1076       has been changed,\MessageBreak
1077       frenchb will NOT customize it}%
1078   \fi
1079 \fi
1080 }

```

## 19.8 The frenchle/pro package

```

1081 \ifundefined{frenchTeXmods}{}{%
1082   \caption@CheckCommand\@makecaption{%
1083     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
1084     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
1085     \def\@makecaption#1#2{%
1086       \ifFTY%
1087         \def\@secondofmany##1##2\void{##2}%
1088         \def\@tempa{\@secondofmany#2\void}%
1089         \ifx\@tempa\empty%
1090           \let\captionseparator\empty%
1091         \fi%
1092         \@mcORI{#1}{\relax\captionfont{#2}}%
1093       \else
1094         \@mcORI{#1}{#2}%
1095       \fi}}
1096   \caption@CheckCommand\@makecaption{%
1097     % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
1098     \def\@makecaption#1#2{%
1099       \ifFTY%
1100         \def\@secondofmany##1##2\void{##2}%
1101         \protected@edef\@tempa{\@secondofmany#2\void}%
1102         \ifx\@tempa\empty%
1103           \let\captionseparator\empty%
1104         \fi%
1105         \@mcORI{#1}{\relax\captionfont{#2}}%
1106       \else
1107         \@mcORI{#1}{#2}%
1108       \fi}}
1109   \DeclareCaptionDefaultFont{textfont}{\itshape}%
1110   \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
1111 }

```

## 19.9 The hungarian and magyar Babel option

```

1112 \DeclareCaptionListFormat{subperiod}{#2.}
1113 \def\caption@tempa#1{\@ifundefined{extras#1}{}{%
1114   \expandafter\addto\csname extras#1\endcsname{%
1115     % change default labelsep and listformat
1116     \caption@setdefaultlabelsep{period}%
1117     \caption@setdefaultlistformat{subperiod}}}%
1118   \expandafter\addto\csname noextras#1\endcsname{%
1119     % change default labelsep and listformat
1120     \caption@setdefaultlabelsep{colon}%
1121     \caption@setdefaultlistformat{subsimple}}}%
1122 }}
1123 \caption@tempa{hungarian}
1124 \caption@tempa{magyar}

```

## 19.10 Unknown document class (or package)

```

1125 \caption@IfCheckCommand{%
1126   \caption@setbool{documentclass}{1}%
1127 }{%

```

```

1128 \caption@setbool{documentclass}{0}%
1129 \caption@InfoNoLine{%
1130     Unknown document class (or package),\MessageBreak
1131     standard defaults will be used}%
1132 \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1133 }

```

## 20 Execution of options

```

1134 \captionsetup{style=default,position=default,%
1135     list,listformat=default,twoside=\if@twoside 1\else 0\fi}
1136 \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`.

```

1137 \newcommand*\caption@addcontentsline[2]{%
1138     \caption@iflist
1139     {\def\@tempa{#2}}%
1140     {\let\@tempa\@empty}%
1141     \ifx\@tempa\@empty \else
1142     {\let\\space
1143     \ifundefined{ext@#1}%
1144     {\caption@Error{No float type ‘#1’ defined}}%
1145     {\caption@@addcontentsline
1146         {\csname ext@#1\endcsname}%
1147         {#1}%
1148         {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1149         {\ignorespaces #2}}}%
1150     \fi}

1151 \newcommand*\caption@@addcontentsline[4]{%
1152     \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`.)

```

1153 \newif\ifcaption@star

```

`\caption@fnum` `\caption@fnum{<float type>}`  
 Typesets the caption label; as replacement for `\fnum{<float type>}`.

```

1154 \newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}

```

`\caption@make` `\caption@make{<float name>}{<ref. number>}{<text>}`  
 Typesets the caption.

```

1155 \newcommand\caption@make[2]{\caption@@make{\caption@lfmt{#1}{#2}}}

```

```

\caption@@make \caption@@make{\caption label}{\caption text}
1156 \newcommand\caption@@make[2]{%
1157 \begingroup
1158 \caption@stepcounter
1159 \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
1160 % TODO: Move this to \caption@calcmargin!?
1161 \ifx\caption@maxmargin\undefined \else
1162 \ifdim\captionmargin>\caption@maxmargin\relax
1163 \captionmargin\caption@maxmargin\relax
1164 \fi
1165 \fi
1166 \ifx\caption@minmargin\undefined \else
1167 \ifdim\captionmargin<\caption@minmargin\relax
1168 \captionmargin\caption@minmargin\relax
1169 \fi
1170 \fi

Special single-line treatment (option singlelinecheck=)
1171 \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%

Typeset the left margin (option margin=)
1172 \caption@calcmargin
1173 \@tempdima\captionmargin
1174 \ifdim\captionmargin@=\z@ \else
1175 \caption@ifoddpage{}{\advance\@tempdima\captionmargin@}%
1176 \fi
1177 \caption@ifh{\advance\@tempdima\caption@indent}%
1178 \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*.
1179 \@tempdima\captionwidth
1180 \caption@ifh{\advance\@tempdima-\caption@indent}%
1181 \caption@parbox\@tempdima{%

Typeset the indentation (option indentation=)
Bugfix 04-05-05: \hskip-\caption@indent replaced by \ifdim\caption@indent=\z@...
1182 \caption@ifh{%
1183 \ifdim\caption@indent=\z@
1184 \leavevmode
1185 \else
1186 \hskip-\caption@indent
1187 \fi}%

Typeset the caption itself and close the \caption@parbox
1188 \caption@@@make{#1}{#2}%

Typeset the right margin (option margin=)
1189 \@tempdima\captionmargin
1190 \ifdim\captionmargin@=\z@ \else
1191 \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
1192 \fi
1193 \hspace\@tempdima

```

```

1194 \caption@endhook
1195 \endgroup
1196 \global\caption@starfalse}

\caption@calcmargin \caption@calcmargin
Calculate \captionmargin & \captionwidth, so both contain valid values.
1197 \newcommand*\caption@calcmargin{%
1198 \caption@calcmargin@hook
1199 \ifdim\captionwidth=\z@
1200 \captionwidth\linewidth
1201 \advance\captionwidth by -2\captionmargin
1202 \advance\captionwidth by -\captionmargin@
1203 \else
1204 \captionmargin\linewidth
1205 \advance\captionmargin by -\captionwidth
1206 \divide\captionmargin by 2
1207 \captionmargin@\z@
1208 \fi

1209 \caption@Debug{%
1210 \string\hsize=\the\hsize,
1211 \string\linewidth=\the\linewidth,\MessageBreak
1212 \string\leftmargin=\the\leftmargin,
1213 \string\rightmargin=\the\rightmargin,\MessageBreak
1214 \string\margin=\the\captionmargin,
1215 \string\margin@=\the\captionmargin@,
1216 \string\width=\the\captionwidth}%
1217 }

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.
1218 \newcommand\caption@slc[4]{%
1219 \caption@@slc{#1}{#2}{#3}{\caption@singleline#4}{}}

1220 \newcommand\caption@@slc[5]{%
1221 \caption@Debug{Begin SLC}%
1222 \beginngroup
1223 \caption@singleline
1224 \let\caption@hj\@empty
1225 \caption@calcmargin % calculate #3 if necessary
1226 \caption@prepareslc
1227 \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1228 \ifdim\wd\@tempboxa>#3%
1229 \endgroup
1230 #5%
1231 \else
1232 \endgroup
1233 #4%
1234 \fi
1235 \caption@Debug{End SLC}}

1236 \newcommand*\caption@singleline{%
1237 \caption@xsetup\caption@opt@singleline
1238 \let\caption@fmt\caption@slfmt}

```

`\caption@prepareslc` `\caption@prepareslc`  
**Re-define anything which would disturb the single-line-check.**

```

1239 \newcommand*\caption@prepareslc{%
1240   \def\label{\caption@withoptargs\@gobbletwo}%

1241   \let\caption@footnote@ORI\footnote
1242   \def\footnote{\caption@withoptargs\caption@footnote}%
1243   \let\caption@footnotemark@ORI\footnotemark
1244   \def\footnotemark{\caption@withoptargs\caption@footnotemark}%
1245   \let\@footnotetext\@gobble

1246   \let\@endnotetext\@gobble}

1247 \newcommand\caption@footnote[2]{%
1248   \begingroup
1249     \let\stepcounter\caption@l@stepcounter
1250     \caption@footnote@ORI#1{#2}%
1251   \endgroup}
1252 \newcommand\caption@footnotemark[1]{%
1253   \begingroup
1254     \let\stepcounter\caption@l@stepcounter
1255     \caption@footnotemark@ORI#1%
1256   \endgroup}

1257 \newcommand*\caption@l@stepcounter[1]{%
1258   \advance\csize c@#1\endcsname\@ne\relax}

```

`\caption@parbox` `\caption@parbox{<width>}{<contents>}`  
**This macro defines the box which surrounds the caption paragraph.**

```

1259 \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=.**

```

1260 \newcommand*\caption@applyfont{%
1261   \caption@hj\captionfont\captionsize}

```

`\caption@@@make` `\caption@@@make{<caption label>}{<caption text>}`  
**This one finally typesets the caption paragraph, without margin and indentation.**

```

1262 \newcommand\caption@@@make[2]{%

If the label is empty, we use no caption label separator.

1263   \sbox\@tempboxa{#1}%
1264   \ifdim\wd\@tempboxa=\z@
1265     \let\caption@lsep\relax
1266   %   \@capbreakfalse
1267   \fi

If the text is empty, we use no caption label separator, too. (And no text format either.)

1268   \caption@ifempty{#2}{%
1269     \let\caption@lsep\@empty
1270     \let\caption@tfmt\@firstofone
1271   %   \@capbreakfalse
1272   %   \let\caption@ifstrut\@secondoftwo
1273   }%

```

Take care that `\caption@parindent` and `\caption@hangindent` will be used to typeset the paragraph.

```
1274 \setpar{\@@par\caption@@par}\caption@@par
```

Finally typeset the caption.

```
1275 \caption@applyfont
1276 \caption@fmt
1277 {\ifcaption@star\else{\captionlabelfont#1}\fi}%
1278 {\ifcaption@star\else{\caption@iflfl\captionlabelfont\caption@lsep}\fi}%
1279 {\caption@textfont
1280 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1281 \nobreak\hskip\z@skip % enable hyphenation
1282 \caption@tfmt{#2}%
1283 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1284 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1285 \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.

```
1286 \newcommand\caption@ifempty[1]{%
1287 \caption@ifempty{#1}%
1288 \caption@ifempty\@unused}

1289 \newcommand\caption@ifempty[1]{%
1290 \def\caption@tempa{#1}%
1291 \ifx\caption@tempa\@empty
1292 \let\caption@ifempty\@secondoftwo
1293 \else
1294 \expandafter\def\expandafter\caption@tempa\expandafter{%
1295 \caption@car#1\caption@ifempty\caption@nil}%
1296 \def\caption@tempb{\caption@ifempty}%
1297 \ifx\caption@tempa\caption@tempb
1298 \let\caption@ifempty\@secondoftwo
1299 \else
1300 \def\caption@tempb{\ignorespaces}%
1301 \ifx\caption@tempa\caption@tempb
1302 \expandafter\caption@ifempty\expandafter{\@gobble#1}%
1303 \else
1304 \def\caption@tempb{\label}%
1305 \ifx\caption@tempa\caption@tempb
1306 \expandafter\caption@ifempty\expandafter{\@gobbletwo#1}%
1307 \else
1308 \def\caption@tempb{\index}%
1309 \ifx\caption@tempa\caption@tempb
1310 \expandafter\caption@ifempty\expandafter{\@gobbletwo#1}%
1311 \else
1312 \def\caption@tempb{\glossary}%
1313 \ifx\caption@tempa\caption@tempb
1314 \expandafter\caption@ifempty\expandafter{\@gobbletwo#1}%
1315 \else
1316 \let\caption@ifempty\@gobbletwo
```

```

1317             \fi
1318         \fi
1319     \fi
1320     \fi
1321     \fi
1322 \fi}

1323 \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.
1324 \newcommand*\caption@@par{%
1325     \parindent\caption@parindent\hangindent\caption@hangindent}%

```

## 23 Types & sub-types

```

\DeclareCaptionType \DeclareCaptionType[<options>]{<environment>}[<name>][<list name>]

1326 \newcommand*\DeclareCaptionType{%
1327     \@testopt\@DeclareCaptionType{}%
1328 \@onlypreamble\DeclareCaptionType

1329 \def\@DeclareCaptionType[#1]#2{%
1330     \def\caption@type{#2}%
1331     \caption@Debug{New type `#2'}%
1332     \newcounter{#2}\@namedef{theH#2}{}%
1333     \KV@caption@DCT@within\caption@within@default
1334     \caption@DeclareWithinOption{#2}%
1335     \KV@caption@DCT@placement{tbp}%
1336     \ifundefined{c@float@type}%
1337     {\newcounter{float@type}%
1338      \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1339     {}%
1340     \caption@Debug{float type `#2'=\the\value{float@type}}%
1341     \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1342     \addtocounter{float@type}{\value{float@type}}%
1343     \KV@caption@DCT@fileext{lo#2}%
1344     \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1345     \newenvironment{#2}{\@float{#2}}{\end@float}%
1346     \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1347     \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1348     \expandafter\newcommand\csname listof#2es\endcsname{\caption@listof{#2}}%
1349     \ifundefined{l@figure}%
1350     {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1351     {\expandafter\let\csname l@#2\endcsname\l@figure}%

1352     \expandafter\newcommand\csname #2name\endcsname{}%
1353     \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
1354     \uppercase\expandafter{\@tempa}%
1355     \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
1356     \@tempb
1357     \expandafter\let\csname #2name\endcsname\@tempa
1358     \expandafter\newcommand\csname list#2name\endcsname{}%
1359     \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%

```



```

1360 \@cons\caption@typelist{{#2}}%
1361 \caption@setkeys[caption]{caption@DCT}{#1}%

1362 \@ifundefined{float@exts}{\newtoks\float@exts}{}%
1363 \let\float@do=\relax
1364 \edef\@tempa{\noexpand\float@exts{\the\float@exts\float@do{\@nameuse{ext@#2}}}}%
1365 \@tempa
1366 \@ifundefined{float@addtolists}{%
1367   \newcommand\float@addtolists[1]{%
1368     \def\float@do###1{\addtocontents{###1}{##1}}\the\float@exts}%
1369   \@ifundefined{@chapter}{}{\caption@PatchChapter}}{}%

1370 \@ifnextchar[\@@DeclareCaptionType\relax}
1371 \@onlypreamble\@DeclareCaptionType

1372 \def\@@DeclareCaptionType[#1]{%
1373   \KV@caption@DCT@name{#1}%
1374   \@ifnextchar[\@@@DeclareCaptionType\relax}
1375 \@onlypreamble\@@DeclareCaptionType

1376 \def\@@@DeclareCaptionType[#1]{%
1377   \KV@caption@DCT@listname{#1}%
1378 \@onlypreamble\@@@DeclareCaptionType

1379 \let\DeclareFloatingEnvironment\DeclareCaptionType % old command name
1380 \@onlypreamble\DeclareFloatingEnvironment

```

`\caption@within@default` The default ‘within’ value.

```

1381 \newcommand*\caption@within@default{\@ifundefined{c@chapter}{none}{chapter}}
1382 \@onlypreamble\caption@within@default

```

`\caption@listof` `\caption@listof{<float type>}`

```

1383 \newcommand*\caption@listof[1]{%
1384   \begingroup
1385     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1386     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1387     \let\caption@ORI@starttoc\@starttoc
1388     \renewcommand*\@starttoc[1]{%
1389       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1390     \listoffigures
1391   \endgroup}

```

`\caption@typelist` An `\elt`-list containing the caption types defined with `\DeclareCaptionType`.

```

1392 \newcommand*\caption@typelist{}

```

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

```

1393 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
1394 \@onlypreamble@key{caption@DCT}{fileext}
1395 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1396 \@onlypreamble@key{caption@DCT}{listname}
1397 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1398 \@onlypreamble@key{caption@DCT}{name}
1399 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
1400 \@onlypreamble@key{caption@DCT}{placement}
1401 \define@key{caption@DCT}{within}{%

```

```

1402 \@ifundefined{c@chapter}{}{\@removefromreset\caption@type{chapter}}%
1403 \@removefromreset\caption@type{section}%
1404 \begingroup
1405   \caption@setkeys[caption]{caption@within}{#1}%
1406 \endgroup
1407 %\@onlypreamble@key{caption@DCT}{within}
1408 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1409 %\@onlypreamble@key{caption@DCT}{without}

1410 \define@key{caption@within}{none}[]{%
1411   \caption@within{}}{}
1412 %\@onlypreamble@key{caption@within}{none}
1413 \define@key{caption@within}{section}[]{%
1414   \@addtoreset\caption@type{section}%
1415   \caption@within{\ifnum\c@section>z@ \thesection.\fi}{\theHsection.}}
1416 %\@onlypreamble@key{caption@within}{section}
1417 \@ifundefined{c@chapter}{}{%
1418   \define@key{caption@within}{chapter}[]{%
1419     \@addtoreset\caption@type{chapter}%
1420     \caption@within{\ifnum\c@chapter>z@ \thechapter.\fi}{\theHchapter.}}
1421 }% \@onlypreamble@key{caption@within}{chapter}}

```

\caption@within \caption@within{*thecode*}{*theHcode*}

```

1422 \newcommand*\caption@within{%
1423   \expandafter\caption@within@\expandafter{\caption@type}}
1424 %\@onlypreamble\caption@within
1425 \newcommand*\caption@within@[3]{%
1426   \global\@namedef{the#1}{#2\arabic{#1}}%
1427   \@ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
1428     {\global\@namedef{theH#1}{#3\arabic{#1}}}}
1429 %\@onlypreamble\caption@within@

```

\@removefromreset This code was taken from the `remreset` package which is part of the ‘`carlisle`’ package bundle. (Copyright 1997 David Carlisle)

```

1430 \providecommand*\@removefromreset[2]{%
1431   \expandafter\let\csname c@#1\endcsname\@removefromreset
1432   \def\@elt##1{%
1433     \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1434     \else
1435       \noexpand\@elt{##1}%
1436     \fi}%
1437   \expandafter\xdef\csname cl@#2\endcsname{%
1438     \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.)

```

1439 \newcommand*\caption@PatchChapter{%
1440   \providecommand*\@chapterlistsgap{10\p}%

1441   % report.cls [2005/09/16 v1.4f Standard LaTeX document class]
1442   \caption@patchchapter{report}{%
1443     \ifnum \c@secnumdepth >\m@ne
1444       \refstepcounter{chapter}%
1445       \typeout{\@chapapp\space\thechapter.}%
1446       \addcontentsline{toc}{chapter}%

```

```

1447         {\protect\numberline{\thechapter}##1}%
1448     \else
1449         \addcontentsline{toc}{chapter}{##1}%
1450     \fi
1451     \chaptermark{##1}%
1452     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1453     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1454     \if@twocolumn
1455         \topnewpage[\@makechapterhead{##2}]%
1456     \else
1457         \@makechapterhead{##2}%
1458         \@afterheading
1459     \fi
1460 }{%
1461     \ifnum \c@secnumdepth >\m@ne
1462         \refstepcounter{chapter}%
1463         \typeout{\@chapapp\space\thechapter.}%
1464         \addcontentsline{toc}{chapter}%
1465             {\protect\numberline{\thechapter}##1}%
1466     \else
1467         \addcontentsline{toc}{chapter}{##1}%
1468     \fi
1469     \chaptermark{##1}%
1470     \ifdim \@chapterlistsgap>\z@
1471         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1472         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1473         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1474     \fi
1475     \if@twocolumn
1476         \topnewpage[\@makechapterhead{##2}]%
1477     \else
1478         \@makechapterhead{##2}%
1479         \@afterheading
1480     \fi}%
1481 % book.cls [2005/09/16 v1.4f Standard LaTeX document class]
1482 \caption@patch@chapter{book}{%
1483     \ifnum \c@secnumdepth >\m@ne
1484         \if@mainmatter
1485             \refstepcounter{chapter}%
1486             \typeout{\@chapapp\space\thechapter.}%
1487             \addcontentsline{toc}{chapter}%
1488                 {\protect\numberline{\thechapter}##1}%
1489         \else
1490             \addcontentsline{toc}{chapter}{##1}%
1491         \fi
1492     \else
1493         \addcontentsline{toc}{chapter}{##1}%
1494     \fi
1495     \chaptermark{##1}%
1496     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1497     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1498     \if@twocolumn
1499         \topnewpage[\@makechapterhead{##2}]%
1500     \else

```

```

1501     \@makechapterhead{##2}%
1502     \@afterheading
1503   \fi
1504 }{%
1505   \ifnum \c@secnumdepth >\m@ne
1506     \if@mainmatter
1507       \refstepcounter{chapter}%
1508       \typeout{\@chapapp\space\thechapter.}%
1509       \addcontentsline{toc}{chapter}%
1510         {\protect\numberline{\thechapter}##1}%
1511     \else
1512       \addcontentsline{toc}{chapter}{##1}%
1513     \fi
1514   \else
1515     \addcontentsline{toc}{chapter}{##1}%
1516   \fi
1517   \chaptermark{##1}%
1518   \ifdim \@chapterlistsgap>\z@
1519     \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1520     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1521     \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1522   \fi
1523   \if@twocolumn
1524     \topnewpage[\@makechapterhead{##2}]%
1525   \else
1526     \@makechapterhead{##2}%
1527     \@afterheading
1528   \fi}%
1529 % amsbook.cls [2004/08/06 v2.20]
1530 % smfbook.cls [1999/11/15 v1.2f Classe LaTeX pour les monographies editees par
1531 \caption@patch@chapter{ams/smfbook}{%
1532   \refstepcounter{chapter}%
1533   \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1534   \else \let\@secnumber\thechapter \fi
1535   \typeout{\chaptername\space\@secnumber}%
1536   \def\@toclevel{0}%
1537   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1538   \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1539   \chaptermark{##1}%
1540   \addtocontents{lof}{\protect\addvspace{10\p@}}%
1541   \addtocontents{lot}{\protect\addvspace{10\p@}}%
1542   \@makechapterhead{##2}\@afterheading
1543 }{%
1544   \refstepcounter{chapter}%
1545   \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1546   \else \let\@secnumber\thechapter \fi
1547   \typeout{\chaptername\space\@secnumber}%
1548   \def\@toclevel{0}%
1549   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1550   \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1551   \chaptermark{##1}%
1552   \ifdim \@chapterlistsgap>\z@
1553     \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1554     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%

```

```

1555     \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1556     \fi
1557     \@makechapterhead{##2}\@afterheading}%

1558 % scrreprt/scrbook.cls
1559 \@ifundefined{KOMAClassName}{}{%
1560     \caption@Debug{document class '\KOMAClassName' detected}%
1561     \let\caption@patch@chapter\@gobblethree}%

1562 % rapport1/3.cls [2004/06/07 v2.1a NTG LaTeX document class]
1563 \caption@patch@chapter{rapport}{%
1564     \ifnum \c@secnumdepth >\m@ne
1565         \refstepcounter{chapter}%
1566         \typeout{\@chapapp\space\thechapter.}%
1567         \addcontentsline{toc}{chapter}%
1568             {\protect\numberline{\thechapter}\toc@font0 ##1}%
1569     \else
1570         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1571     \fi
1572     \chaptermark{##1}%
1573     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1574     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1575     \if@twocolumn
1576         \@topnewpage[\@makechapterhead{##2}]%
1577     \else
1578         \@makechapterhead{##2}%
1579         \@afterheading
1580     \fi
1581 }{%
1582     \ifnum \c@secnumdepth >\m@ne
1583         \refstepcounter{chapter}%
1584         \typeout{\@chapapp\space\thechapter.}%
1585         \addcontentsline{toc}{chapter}%
1586             {\protect\numberline{\thechapter}\toc@font0 ##1}%
1587     \else
1588         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1589     \fi
1590     \chaptermark{##1}%
1591     \ifdim \@chapterlistsgap>\z@
1592         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1593         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1594         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1595     \fi
1596     \if@twocolumn
1597         \@topnewpage[\@makechapterhead{##2}]%
1598     \else
1599         \@makechapterhead{##2}%
1600         \@afterheading
1601     \fi}%

1602 % boek(3).cls [2004/06/07 v2.1a NTG LaTeX document class]
1603 \caption@patch@chapter{boek}{%
1604     \ifnum \c@secnumdepth >\m@ne
1605         \if@mainmatter
1606             \refstepcounter{chapter}%
1607             \typeout{\@chapapp\space\thechapter.}%

```

```

1608         \addcontentsline{toc}{chapter}%
1609         {\protect\numberline{\thechapter}\toc@font0 ##1}%
1610     \else
1611         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1612     \fi
1613 \else
1614     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1615 \fi
1616 \chaptermark{##1}%
1617 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1618 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1619 \if@twocolumn
1620     \topnewpage[\@makechapterhead{##2}]%
1621 \else
1622     \@makechapterhead{##2}%
1623     \@afterheading
1624 \fi
1625 }{%
1626     \ifnum \c@secnumdepth >\m@ne
1627         \if@mainmatter
1628             \refstepcounter{chapter}%
1629             \typeout{\@chapapp\space\thechapter.}%
1630             \addcontentsline{toc}{chapter}%
1631             {\protect\numberline{\thechapter}\toc@font0 ##1}%
1632         \else
1633             \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1634         \fi
1635     \else
1636         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1637     \fi
1638     \chaptermark{##1}%
1639     \ifdim \@chapterlistsgap>\z@
1640         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1641         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1642         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1643     \fi
1644     \if@twocolumn
1645         \topnewpage[\@makechapterhead{##2}]%
1646     \else
1647         \@makechapterhead{##2}%
1648         \@afterheading
1649     \fi}%
1650 % thesis.cls [1996/25/01 1.0g LaTeX document class (wm).]
1651 \caption@patch@chapter{thesis}{%
1652     \ifnum \c@secnumdepth >\m@ne
1653         \if@mainmatter
1654             \refstepcounter{chapter}%
1655             \typeout{\@chaptername\space\thechapter.}
1656             \if@thema
1657                 \ifx\@shortauthor\@empty
1658                     \addcontentsline{toc}{chapter}{%
1659                         \protect\numberline{\thechapter.}##1}%
1660                 \else
1661                     \addcontentsline{toc}{chapter}{%

```

```

1662         \protect\numberline{\thechapter.}%
1663         \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
1664     \fi
1665     \else
1666         \addcontentsline{toc}{chapter}{%
1667         \protect\numberline{\thechapter.}##1}%
1668     \fi
1669     \else
1670         \addcontentsline{toc}{chapter}{##1}
1671     \fi
1672 \else
1673     \addcontentsline{toc}{chapter}{##1}
1674 \fi
1675 \chaptermark{##1}
1676 \addtocontents{lof}{\protect\addvspace{10pt}}
1677 \addtocontents{lot}{\protect\addvspace{10pt}}
1678 \if@twocolumn
1679     \topnewpage[\@makechapterhead{##2}]
1680 \else
1681     \@makechapterhead{##2}
1682     \@afterheading
1683 \fi
1684 }{%
1685     \ifnum \c@secnumdepth >\m@ne
1686         \if@mainmatter
1687             \refstepcounter{chapter}%
1688             \typeout{\chaptername\space\thechapter.}%
1689             \if@thema
1690                 \ifx\@shortauthor\@empty
1691                     \addcontentsline{toc}{chapter}{%
1692                     \protect\numberline{\thechapter.}##1}%
1693                 \else
1694                     \addcontentsline{toc}{chapter}{%
1695                     \protect\numberline{\thechapter.}%
1696                     \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
1697                 \fi
1698             \else
1699                 \addcontentsline{toc}{chapter}{%
1700                 \protect\numberline{\thechapter.}##1}%
1701             \fi
1702         \else
1703             \addcontentsline{toc}{chapter}{##1}%
1704         \fi
1705     \else
1706         \addcontentsline{toc}{chapter}{##1}%
1707     \fi
1708     \chaptermark{##1}%
1709     \ifdim \@chapterlistsgap>\z@
1710         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1711         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1712         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1713     \fi
1714     \if@twocolumn
1715         \topnewpage[\@makechapterhead{##2}]%

```

```

1716     \else
1717         \@makechapterhead{##2}%
1718         \@afterheading
1719     \fi}%

1720 \ifx\caption@patch@chapter\@gobblethree \else
1721     \caption@Debug{%
1722         Unsupported document class detected,\MessageBreak
1723         or \noexpand\@chapter was redefined by another package}%
1724 \fi
1725 \let\caption@PatchChapter\@undefined}
1726 %\@onlypreamble\caption@PatchChapter

1727 \newcommand\caption@patch@chapter[3]{%
1728     \begingroup
1729 %     \let\if@twocolumn\iffalse
1730     \let\if@mainmatter\iffalse
1731     \let\if@thema\iffalse
1732     \def\@tempa[##1]##2{#2}%
1733     \ifx\@tempa\@chapter
1734         \caption@Debug{document class '#1' detected}%
1735         \gdef\@chapter[##1]##2{#3}%
1736         \global\let\caption@patch@chapter\@gobblethree
1737     \fi
1738     \endgroup}
1739 %\@onlypreamble\caption@patch@chapter

1740 \long\def \@gobblethree #1#2#3{}

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

1741 \newcommand*\caption@patch@stpelt{%
1742     \let\caption@stpelt\@stpelt
1743     \def\@stpelt##1{%
1744         \caption@stpelt{##1}%
1745         \begingroup
1746             \let\@elt\caption@stpelt
1747             \csname caption@cl@##1\endcsname
1748         \endgroup}%
1749     \let\caption@patch@stpelt\relax}
1750 \@onlypreamble\caption@patch@stpelt

\caption@addtoreset Like \@addtoreset from ltcounts.dtx

1751 \newcommand*\caption@addtoreset[2]{%
1752     \caption@patch@stpelt
1753     \@ifundefined{caption@cl@#2}{\@namedef{caption@cl@#2}{}{}}{%
1754         \expandafter\@cons\csname caption@cl@#2\endcsname{{#1}}}%
1755 \@onlypreamble\caption@addtoreset

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

1756 \newcommand*\caption@removefromreset[2]{%
1757     \begingroup
1758         \expandafter\let\csname c@#1\endcsname\caption@removefromreset
1759     \def\@elt##1{%
1760         \expandafter\ifx\csname c@##1\endcsname\caption@removefromreset

```



```

1761         \else
1762             \noexpand\@elt{##1}%
1763         \fi}%
1764     \expandafter\xdef\csname caption@cl@#2\endcsname{%
1765         \csname caption@cl@#2\endcsname}%
1766 \endgroup}
1767 \@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>*.

```

1768 \newcommand*\DeclareCaptionSubType{%
1769     \caption@teststar\caption@declaresubtype\@firstoftwo\@secondoftwo}
1770 \@onlypreamble\DeclareCaptionSubType

1771 \newcommand*\caption@declaresubtype[1]{%
1772     \@testopt{\caption@@declaresubtype{#1}}{\alph}}
1773 \@onlypreamble\caption@declaresubtype

1774 \def\caption@@declaresubtype#1[#2]#3{%
1775     \ifundefined{c@#3}%
1776         {\caption@Error{No float type ‘#3’ defined}}%
1777     {\ifundefined{c@sub#3}%
1778         {\caption@Debug{New subtype ‘sub#3’}%
1779         \newcounter{sub#3}%
1780         \caption@addtoreset{sub#3}{#3}%
1781         \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1782         \caption@declaresublistentry{#3}%
1783         \@cons\caption@subtypelist{{#3}}}%
1784         {\caption@Debug{Modify caption ‘sub#3’}}}%

1785     \@namedef{sub#3name}{}%
1786     \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
1787     #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
1788     {\@namedef{p@sub#3}{}%
1789     \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1790     {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1791     \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1792     \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}%
1793     }}

1794 \@onlypreamble\caption@@declaresubtype

1795 \newcommand*\caption@declaresublistentry{%
1796     \ifundefined{l@chapter}%
1797         {\caption@@declaresublistentry\l@subsubsection}%
1798         {\caption@@declaresublistentry\l@section}}
1799 \@onlypreamble\caption@declaresublistentry

1800 \newcommand*\caption@@declaresublistentry[2]{%
1801     \ifx#1\@undefined
1802         \caption@@@declaresublistentry\relax\@dottedtocline\caption@nil{#2}%
1803     \else
1804         \expandafter\caption@@@declaresublistentry#1{}{}\@dottedtocline\caption@nil{#
1805     \fi}
1806 \@onlypreamble\caption@@declaresublistentry

```

```

1807 \long\def\caption@@@declaresublistentry#1\@dottedtocline#2\caption@nil#3{%
1808   \def\@tempa{#1}%
1809 % Does \l@ (sub)section start with \@dottedtocline?
1810   \ifx\@tempa\@empty
1811 % Yes
1812     \caption@@@declaresublistentry{#3}#2\caption@nil
1813   \else
1814 % No
1815     \caption@@@declaresublistentry{#3}@{3.8em}{3.2em}\caption@nil
1816   \fi}
1817 \@onlypreamble\caption@@@declaresublistentry
1818 \def\caption@@@declaresublistentry#1#2#3#4#5\caption@nil{%
1819   \expandafter\caption@@@declaresublistentry\expandafter
1820     {\csname @dotted\csname ext@#1\endcsname line\endcsname}{#1}{#3}{#4}}
1821 \@onlypreamble\caption@@@declaresublistentry
1822 \newcommand*\caption@@@declaresublistentry[4]{%
1823   \@namedef{l@sub#2}{#1{2}{#3}{#4}}%
1824   \caption@@@declaresublistentry#1{c@\csname ext@#2\endcsname depth}}
1825 \@onlypreamble\caption@@@declaresublistentry
1826 \newcommand*\caption@@@@declaresublistentry[2]{
1827   \ifx#1\relax
1828     \def#1##1{%
1829       \def\next{\@dottedtocline{##1}}%
1830       \@ifundefined{#2}{}{%
1831         \ifnum ##1>\@nameuse{#2}\relax
1832         \let\next\@gobblefour
1833       \fi}%
1834     \next}%
1835   \fi}
1836 \@onlypreamble\caption@@@@declaresublistentry

```

`\caption@subtypelist` An `\@elt-list` containing the subtypes defined with `\DeclareCaptionSubType`.

```

1837 \newcommand*\caption@subtypelist{}

```

`\caption@For` `\caption@For{<elt-list>}{<code with #1>}`  
`\caption@For*{<elt-list>}{<code with #1>}`

```

1838 \newcommand*\caption@For{\caption@withoptargs\caption@For}
1839 %\@onlypreamble\caption@For
1840 \newcommand\caption@@For[3]{%
1841   \caption@AtBeginDocument#1{%
1842     \def\@elt##1{#3}%
1843     \@nameuse{caption@#2}%
1844     \let\@elt\relax}}%
1845 %\@onlypreamble\caption@@For

```

## 24 subfig package adoptions

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

```

1846 \caption@AtBeginDocument{%
1847   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1848   \ifx\subref\@tempa
1849     \caption@InfoNoLine{subfig package 1.2 or 1.3 is loaded}%

```

```

1850 \let\caption@setfloattype\@gobble
1851 \let\@dottedxxxline\sf@NEW@dottedxxxline
1852 \let\sf@subfloat\sf@NEW@subfloat

1853 \fi
1854 \let\sf@NEW@dottedxxxline\@undefined
1855 \let\sf@NEW@subfloat\@undefined}

1856 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1857 \begingroup
1858 \caption@setfloattype{#1}%
1859 \caption@setoptions{subfloat}%
1860 \caption@setoptions{sub#1}%
1861 \ifnum #3>\@nameuse{c@#2depth}\else
1862 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1863 \fi
1864 \endgroup}

1865 \def\sf@NEW@subfloat{%
1866 \begingroup
1867 \caption@setfloattype\@capttype
1868 \sf@ifpositiontop{%
1869 \maincaptiontoptrue
1870 }{%
1871 \maincaptiontopfalse
1872 }%
1873 \caption@setoptions{subfloat}%
1874 \caption@setoptions{sub\@capttype}%
1875 \let\sf@oldlabel=\label
1876 \let\label=\subfloat@label
1877 \ifmaincaptiontop\else
1878 \advance\@nameuse{c@\@capttype}\@ne
1879 \fi
1880 \refstepcounter{sub\@capttype}%
1881 \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1882 \ifnextchar [% %] match left bracket
1883 {\sf@@subfloat}%
1884 {\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