

The Implementation of the caption kernel*

Axel Sommerfeldt

`axel.sommerfeldt@f-m.fm`

2011/10/09

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 \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 \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 v1.4, last revised 2011/10/09.

Contents

1 Identification	4
2 Generic helpers	4
3 Information, Warnings, and Errors	8
4 Using the keyval package	8
5 Margin resp. width	14
6 Indentions	16
7 Styles	17
8 Formats	18
9 Label formats	20
10 Label separators	20
11 Text formats	21
12 Fonts	22
13 Justifications	23
13.1 The ragged2e package	24
14 Vertical spaces before and after captions	25
15 Positioning	26
16 Hooks	27
17 Lists	27
18 Debug option	29
19 Document classes & Babel support	29
19.1 The standard L ^A T _E X classes	29
19.2 The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	29
19.3 The beamer class (Part one)	31
19.4 The KOMA-Script classes	32
19.5 The NTG Dutch classes	33
19.6 The thesis class	33

19.7 The frenchb Babel option	34
19.8 The frenchle/pro package	34
19.9 The hungarian and magyar Babel option	35
19.10 Unknown document class (or package)	35
20 Execution of options	36
21 Making an ‘List of’ entry	36
22 Typesetting the caption	37
23 Types & sub-types	41
24 subfig package adaption	52

1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2011/10/09 v1.4 caption3 kernel (AR)]
```

2 Generic helpers

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

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

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

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

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

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

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

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

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

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

would expand to yes.

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

```

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

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

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

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

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

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

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

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

```

```

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

```

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

```

```

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

\caption@gobble \caption@gobble*[\langle arg \rangle][\langle ... \rangle]{\langle arg \rangle}
Same as \@gobble, but gobbles optional arguments as well.
106 \DeclareRobustCommand*\caption@gobble{%
107   \caption@withoptargs\@gobbletwo}

\caption@CheckCommand \caption@CheckCommand{\langle macro \rangle}{\langle definition of macro \rangle}
\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{\langle true \rangle}{\langle false \rangle}
will execute the \langle true \rangle code if one match was finally given, the \langle false \rangle code otherwise.
(It simply checks if \caption@CheckCommand is \@gobbletwo and restores the
starting definition of \caption@CheckCommand.)
108 \newcommand\caption@DoCheckCommand[2]{%
109   \begingroup
110     \let\@tempa#1%
111     #2%
112     \ifx\@tempa#1%
113       \endgroup
114       \let\caption@CheckCommand\@gobbletwo
115     \else
116       \endgroup
117     \fi}
118 \@onlypreamble\caption@DoCheckCommand
119 \let\caption@CheckCommand\caption@DoCheckCommand
120 \@onlypreamble\caption@CheckCommand
121 \newcommand*\caption@IfCheckCommand{%
122   \ifx\caption@CheckCommand\@gobbletwo
123     \let\caption@CheckCommand\caption@DoCheckCommand
124     \expandafter\@firstoftwo
125   \else
126     \expandafter\@secondoftwo
127   \fi}
128 \@onlypreamble\caption@IfCheckCommand

\caption@AtBeginDocument \caption@AtBeginDocument*{\langle code \rangle}
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.
129 \let\caption@begindocumenthook\@empty
130 \let\caption@@begindocumenthook\@empty
131 \def\caption@AtBeginDocument{%
132   \caption@teststar#g@addto@macro
133   \caption@@begindocumenthook\caption@begindocumenthook}
134 %\@onlypreamble\caption@AtBeginDocument
135 \AtBeginDocument{%
136   \caption@InfoNoLine{Begin \noexpand\AtBeginDocument code}%

```

```

137 \def\caption@AtBeginDocument{%
138 \ifstar{\g@addto@macro\caption@@begindocumenthook}\@firstofone}%
139 \caption@begindocumenthook
140 \let\caption@begindocumenthook\relax
141 \def\caption@AtBeginDocument{%
142 \ifstar\@firstofone\@firstofone}%
143 \caption@@begindocumenthook
144 \let\caption@@begindocumenthook\relax
145 \caption@InfoNoLine{End \noexpand\AtBeginDocument code}}

```

3 Information, Warnings, and Errors

```

\caption@Info \caption@Info{<message>}
146 \newcommand*\caption@Info[1]{%
147 \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.

```

148 \newcommand*\caption@InfoNoLine[1]{%
149 \PackageInfo{caption}{#1\@gobble}}

```

```

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

```

```

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

```

```

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

```

```

\caption@KV@err
162 \let\caption@KV@err\caption@Error

```

4 Using the keyval package

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

```

163 \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.

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

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

Analogous to `\@onlypreamble` from `LATEX 2ε`.

```
166 \providecommand*\@preamble@keys{}
167 \providecommand*\@onlypreamble@key[2]{\@cons\@preamble@keys{{#1}{#2}}}
168 \@onlypreamble\@onlypreamble@key
169 \@onlypreamble\@preamble@keys
170 \providecommand*\@notprerr@key[1]{\KV@err{Can be used only in preamble}}
171 \caption@AtBeginDocument{%
172   \def\@elt#1#2{\expandafter\let\csname KV@#1@#2\endcsname\@notprerr@key}%
173   \@preamble@keys
174   \let\@elt\relax}
```

`\DeclareCaptionOption` `\DeclareCaptionOption{<option>}[<default value>]{<code>}`
`\DeclareCaptionOption*{<option>}[<default value>]{<code>}`

We declare our options using these commands (instead of using `\DeclareOption` offered by `LATEX 2ε`), so the keyval package is used. The starred form makes the option available during the lifetime of the current package only, so they can be used with `\usepackage`, but *not* with `\captionsetup` later on.

```
175 \newcommand*\DeclareCaptionOption{%
176   \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
177 \@onlypreamble\DeclareCaptionOption
178 \newcommand*\caption@declareoption[2]{%
179   #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
180 \@onlypreamble\caption@declareoption
```

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

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

```
181 \newcommand*\DeclareCaptionOptionNoValue{%
182   \caption@teststar\caption@declareoption@novalue\AtEndOfPackage\@gobble}
183 \@onlypreamble\DeclareCaptionOptionNoValue
184 \newcommand*\caption@declareoption@novalue[3]{%
185   \caption@declareoption{#1}{#2}[\KV@err]{%
186     \caption@option@novalue{#2}{##1}{#3}}
187 \@onlypreamble\caption@declareoption@novalue
188 \newcommand*\caption@option@novalue[2]{%
189   \ifx\KV@err#2%
190     \expandafter\@firstofone
191   \else
192     \KV@err{No value allowed for #1}%
193     \expandafter\@gobble
194   \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`.)

```
195 \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.

```
196 \newcommand*\captionsetup{%
197   \caption@teststar@\@captionsetup\@gobble\@firstofone}

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

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

202 \def\caption@setup@options[#1]#2{%
203   \@bsphack
204   \ifcaptionsetup@star\captionsetup@starfalse\else\caption@addtooptlist{#1}\fi
205   \expandafter\caption@l@addto@list\csname caption@opt@#1\endcsname{#2}%
206   \@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.

```
207 \newcommand*\clearcaptionsetup{%
208   \caption@teststar@\@clearcaptionsetup\@gobble\@firstofone}

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

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

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

225 \newcommand*\caption@removefromsetup{%
226   \let\caption@tempa\@gobble
227   \caption@@removefromsetup}

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

```

```

233 \@for\@tempa:=\@tempa\do{%
234   \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
235   \ifx\@tempd\@tempc
236     \let\caption@tempa\@gobble
237   \else
238     \expandafter\expandafter\expandafter\caption@l@addto@list
239     \expandafter\csname caption@opt@#2\expandafter\endcsname
240     \expandafter{\@tempa}%
241   \fi}%
242 \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
243 {\caption@removefromoptlist{#2}}}%
244 \caption@tempa{\caption@Warning{%
245   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>*.

```

246 \newcommand*\showcaptionsetup[2][\@firstofone]{%
247   \@bsphack
248   \GenericWarning{ }{%
249     #1 Caption Info: Option list on '#2'\MessageBreak
250     #1 Caption Data: \ifundefined{caption@opt@#2}{%
251       -none-%
252     }{%
253       {\expandafter\expandafter\expandafter\strip@prefix
254         \expandafter\meaning\csname caption@opt@#2\endcsname}%
255       }%
256   \@esphack}

257 \DeclareCaptionOption{options}{\caption@setoptions{#1}}
258 \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.)

```

259 \newcommand*\caption@setoptions{%
260   \caption@teststar\caption@set@options\@gobble\@firstofone}

261 \newcommand*\caption@set@options[2]{%
262   \caption@Debug{options=#2}%
263   \expandafter\let\expandafter\caption@opt\csname caption@opt@#2\endcsname
264   \ifx\caption@opt\relax \else
265     \caption@xsetup\caption@opt
266     #1{\caption@clearsetup{#2}}% #1 = \@firstofone -or- \@gobble
267   \fi}

268 \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”).

```

269 \newcommand*\caption@addtooptlist[1]{%
270   \@ifundefined{caption@opt@#1@lineno}{%
271     \caption@dooptlist\caption@g@addto@list{#1}%
272     \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
273   }{}}

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

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

282 \AtEndDocument{%
283   \caption@for@list\caption@optlist{%
284     \caption@WarningNoLine{%
285       Unused \string\caption@setup[1]
286       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.

```

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

288 \long\def\caption@@setkeys[#1]#2#3{%
289   \@bsphack

290   \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
291   \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
292   \expandafter\let\csname ORI@XKV@err\caption@keydepth\endcsname\XKV@err
293   \edef\caption@keydepth{\caption@keydepth i}%

294   \expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
295   \ifx\KV@err\relax
296     \def\KV@err##1{\PackageError{#1}{##1}{%
297       See the #1 package documentation for explanation.}}%
298   \fi
299   \def\KV@errx{\KV@err}%
300   \def\XKV@err{\let\@tempa\XKV@tkey\KV@err}%

301   \caption@Debug{\protect\setkeys{#2}{#3}}%
302   \setkeys{#2}{#3}%

303   \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
304   \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
305   \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
306   \expandafter\let\expandafter\XKV@err\csname ORI@XKV@err\caption@keydepth\endcsname

307   \ifx\caption@keydepth\@empty \caption@setup@starfalse \fi

308   \@esphack}

309 \let\caption@keydepth\@empty
```

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

We execute our options using the keyval interface, so we use this one instead of `\ExecuteOptions` offered by L^AT_EX 2_ε.

```

310 \newcommand*\caption@ExecuteOptions[2]{%
311   \expandafter\@expandtwoargs\csname caption@setkeys@#1\endcsname{#1}{#2}}%
312 \@onlypreamble\caption@ExecuteOptions

```

`\caption@ProcessOptions`

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

We process our options using the keyval package, so we use this one instead of `\ProcessOptions` offered by L^AT_EX 2_ε. The starred variant do not process the global options. (This code was taken from the hyperref package[2] v6.74 and improved.)

```

313 \newcommand*\caption@ProcessOptions{%
314   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
315 \@onlypreamble\caption@ProcessOptions

316 \newcommand\caption@@ProcessOptions[2]{%
317   \let\@tempc\relax
318   \let\caption@tempa\@empty
319   #1{% \@firstofone -or- \@gobble
320     \for\CurrentOption:=\@classoptionslist\do{%
321       \ifundefined{KV@#2\CurrentOption}{}{%
322         \ifundefined{KV@#2\CurrentOption @default}{}%
323         \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
324       }%
325       \PackageInfo{#2}{Global option '\CurrentOption' processed}%
326       \edef\caption@tempa{\caption@tempa\CurrentOption,%
327         \expandtwoargs\@removeelement\CurrentOption
328         \@unusedoptionlist\@unusedoptionlist
329       }%
330     }%
331   }%
332   \let\CurrentOption\@empty
333 }%
334 \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
335 \AtEndOfPackage{\let\@unprocessedoptions\relax}
336 \@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.)

```

337 \newcommand*\caption@packagelist{}
338 \@onlypreamble\caption@packagelist

339 \newcommand\caption@SetupOptions[2]{%
340   \@namedef{caption@setkeys@#1}##1##2{#2}%
341   \expandafter\@onlypreamble\csname caption@setkeys@#1\endcsname
342   \@cons\caption@packagelist{{#1}}
343 \@onlypreamble\caption@SetupOptions

344 \let\caption@onefilewithoptions\@onefilewithoptions
345 \def\@onefilewithoptions#1[#2]{%
346   \begingroup
347   \def\@tempa{%
348     \endgroup
349     \caption@onefilewithoptions{#1}{{#2}}}%
350   \def\@tempb{#1}%
351   \def\@elt##1{%

```

```

352 \def\@tempc{##1}%
353 \ifx\@tempb\@tempc
354 \def\@tempa{%
355 \endgroup
356 \caption@ExecuteOptions{#1}{#2}%
357 \caption@onefilewithoptions{#1}[]}%
358 \fi}
359 \caption@packagelist
360 \@tempa}
361 \@onlypreamble\caption@onefilewithoptions

```

5 Margin resp. width

`\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 *vt.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@`.

```

362 \newdimen\captionmargin
363 \newdimen\captionmargin@
364 \newdimen\captionwidth

365 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
366 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
367 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
368 \DeclareCaptionOption{width*}{\setcaptionwidth*{#1}}

369 \DeclareCaptionOption{calcmargin}{\caption@setcalcmargin{#1}}
370 \DeclareCaptionOption{calcmargin*}{\caption@setcalcmargin*{#1}}
371 \DeclareCaptionOption{calcwidth}{\caption@setcalcwidth{#1}}
372 \DeclareCaptionOption{calcwidth*}{\caption@setcalcwidth*{#1}}

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

375 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
376 \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.

```

377 \newcommand*\setcaptionmargin{%
378 \caption@resetcalcmargin
379 \caption@setmargin}

380 \newcommand*\caption@setmargin{%
381 \caption@teststar\caption@@setmargin\@gobble\@firstofone}

382 \newcommand*\caption@@setmargin[2]{%
383 #1{\captionwidth\z@}%
384 \caption@@@setmargin#2,#2,\@nil}

```

```

385 \def\caption@@@setmargin#1,#2,#3\@nil{%
386   \setlength\captionmargin@{#2}%
387   \setlength\captionmargin{#1}%
388   \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.

```

389 \newcommand*\setcaptionwidth{%
390   \caption@resetcalcmargin
391   \caption@setwidth}
392 \newcommand*\caption@setwidth{%
393   \caption@teststar\caption@@setwidth\@gobble\@firstofone}
394 \newcommand*\caption@@setwidth[2]{%
395   #1\captionmargin\z@\captionmargin@\z@}%
396   \setlength\captionwidth{#2}}%

```

\caption@resetcalcmargin

```

397 \newcommand*\caption@resetcalcmargin{%
398   \let\caption@calcmargin@hook\@empty}

```

\caption@setcalcmargin

```

399 \newcommand*\caption@setcalcmargin{%
400   \caption@teststar{\caption@@setcalcmargin\caption@setmargin}%
401   \@secondoftwo\@firstoftwo}
402 \newcommand*\caption@@setcalcmargin[3]{%
403   #2{\caption@resetcalcmargin
404     \l@addto@macro\caption@calcmargin@hook{#1{#3}}}%
405     {\l@addto@macro\caption@calcmargin@hook{#1*{#3}}}}

```

\caption@setcalwidth

```

406 \newcommand*\caption@setcalwidth{%
407   \caption@teststar{\caption@@setcalcmargin\caption@setwidth}%
408   \@secondoftwo\@firstoftwo}

```

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

```

409 \newcommand*\caption@thecounter{0}
410 \newcommand*\caption@stepcounter{%
411   \@tempcnta\caption@thecounter
412   \advance\@tempcnta\@ne
413   \xdef\caption@thecounter{\the\@tempcnta}}

```

\caption@newlabel This command is a modified version of \newlabel from L^AT_EX2e. 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.)

```

414 \newcommand*\caption@newlabel{\@newl@bel{caption@r}}

```

\caption@thepage This command is a modified version of \thepage from L^AT_EX2e. It will be used inside \caption@ifoddpage only.

```

415 \newcommand*\caption@thepage{\the\c@page}

```

`\caption@label` This command is a modified version of `\label` from L^AT_EX2e. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

416 \newcommand*\caption@label[1]{%
417   \caption@@label
418   \protected@write\@auxout{\let\caption@thepage\relax}%
419     {\string\caption@newlabel{#1}{\caption@thepage}}}
420 \newcommand*\caption@@label{%
421   \global\let\caption@@label\relax
422   \protected@write\@auxout{}%
423     {\string\providecommand*\string\caption@newlabel[2]{}}}
```

`\caption@pageref` This command is a modified version of `\pageref` from L^AT_EX2e. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

424 \newcommand*\caption@pageref[1]{%
425   \expandafter\ifx\csname caption@r@#1\endcsname\relax
426     \G@refundefinedtrue % => 'There are undefined references.'
427     \@latex@warning{Reference '#1' on page \thepage \space undefined}%
428   \else
429     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
430   \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.

```

431 \newcommand*\caption@ifoddpage{%
432   \caption@iftwoside{%
433     \caption@label\caption@thecounter
434     \caption@pageref\caption@thecounter
435     \ifodd\caption@thepage
436       \let\caption@ifoddpage\@firstoftwo
437     \else
438       \let\caption@ifoddpage\@secondoftwo
439     \fi
440   }{\let\caption@ifoddpage\@firstoftwo}%
441   \caption@ifoddpage}
```

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

```

442 \newcommand*\caption@setoptcmd[2]{%
443   \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 444 \newdimen\caption@indent
\caption@hangindent 445 \newdimen\caption@parindent
446 \newdimen\caption@hangindent

447 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
448   \setlength\caption@indent{#1}}
449 \DeclareCaptionOption{indentation}[\leftmargini]{%
450   \setlength\caption@indent{#1}}
```



```

451 \DeclareCaptionOption{parindent}{%
452     \setlength\caption@parindent{#1}}
453 \DeclareCaptionOption{hangindent}{%
454     \setlength\caption@hangindent{#1}}
455 \DeclareCaptionOption{parskip}{%
456     \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).¹

```

457 \providecommand*\caption@ifkomaclass{%
458     \caption@ifundefined\scr@caption\@gobble\@firstofone}
459 \@onlypreamble\caption@ifkomaclass
460 \caption@ifkomaclass{%
461     \let\caption@KV@parindent\KV@caption@parindent
462     \DeclareCaptionOption{parindent}[]{%
463         \ifx,#1,%
464             \caption@Debug{Option 'parindent' ignored}%
465         \else
466             \caption@KV@parindent{#1}%
467         \fi}%
468     \let\caption@KV@parskip\KV@caption@parskip
469     \DeclareCaptionOption{parskip}[]{%
470         \ifx,#1,%
471             \caption@Debug{Option 'parskip' ignored}%
472         \else
473             \caption@KV@parskip{#1}%
474         \fi}%
475 }

```

7 Styles

```

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

```

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

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

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

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

```
487 \newcommand*\caption@setstyle{%
488   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

489 \newcommand*\caption@@setstyle[2]{%
490   \ifundefined{caption@sty@#2}%
491     {#1{\caption@Error{Undefined style `#2'}}}%
492     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
493     \ifx\caption@setstyle@flag\@undefined
494       \let\caption@setstyle@flag\relax
495       \caption@resetstyle
496       \caption@xsetup\caption@sty
497       \let\caption@setstyle@flag\@undefined
498     }else
499       \caption@xsetup\caption@sty
500     \fi
501     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
502     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
503     \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!*

```
504 \newcommand*\caption@resetstyle{%
505   \caption@setup{%
506     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
507     justification=justified,font=,size=,labelfont=,textfont=,%
508     margin=0pt,minmargin=0,maxmargin=0,%
509     indent=0pt,parindent=0pt,hangindent=0pt,%
510     slc,rule,strut}%
511   \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^AT_EX document classes (and was called ‘default’ in the caption kernel *v1.0*), the second one matches the document class actually used.

```
512 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
513 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
514   format=default,labelsep=default,textformat=default,%
515   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.

```

516 \newcommand*\DeclareCaptionFormat{%
517   \caption@teststar\caption@declareformat\@gobble\@firstofone}
518 \@onlypreamble\DeclareCaptionFormat

519 \newcommand*\caption@declareformat[2]{%
520   \@dblarg{\caption@@declareformat#1{#2}}}%
521 \@onlypreamble\caption@declareformat

522 \long\def\caption@@declareformat#1#2[#3]#4{%
523   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
524   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
525   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
526 \@onlypreamble\caption@@declareformat

527 \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`).

```

528 \newcommand*\caption@setformat[1]{%
529   \ifundefined{caption@fmt@#1}%
530     {\caption@Error{Undefined format '#1'}}%
531     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
532       \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
533       \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

`\DeclareCaptionDefaultFormat`

```

534 \newcommand*\DeclareCaptionDefaultFormat[1]{%
535   \expandafter\def\expandafter\caption@fmt@default\expandafter
536     {\csname caption@fmt@#1\endcsname}%
537   \expandafter\def\expandafter\caption@slfmt@default\expandafter
538     {\csname caption@slfmt@#1\endcsname}%
539   \expandafter\def\expandafter\caption@ifh@default\expandafter
540     {\csname caption@ifh@#1\endcsname}}
541 \@onlypreamble\DeclareCaptionDefaultFormat

```

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

```

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

543 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
544   \caption@ifin@list\caption@lsepclist\caption@lsepname
545   {\caption@Error{%
546     The option ‘\labelsep=\caption@lsepname’ does not work\MessageBreak
547     with ‘format=hang’}}%
548   {\@hangfrom{#1#2}%
549     \advance\caption@parindent\hangindent
550     \advance\caption@hangindent\hangindent
551     \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

552 \DeclareCaptionDefaultFormat{plain}

```

9 Label formats

```

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

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

```

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

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

```

557 \newcommand*{\caption@setlabelformat}[1]{%
558   \ifundefined{caption@lfmt@#1}%
559     {\caption@Error{Undefined label format `#1'}}%
560     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}

```

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

```

561 \DeclareCaptionLabelFormat{empty}{}
562 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
563 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace}{#2}}
564 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace}#2)}

```

‘default’ usually maps to ‘simple’.

```
565 \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`.

```

566 \newcommand\DeclareCaptionLabelSeparator{%
567   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
568 \@onlypreamble\DeclareCaptionLabelSeparator

569 \newcommand\caption@declarelabelseparator[3]{%
570   \global\@namedef{caption@iflf@#2}{#1}%
571   \global\long\@namedef{caption@lsep@#2}{#3}%
572   \caption@@declarelabelseparator{#2}#3\\@nil}
573 \@onlypreamble\caption@declarelabelseparator

574 \long\def\caption@@declarelabelseparator#1#2\\#3\\@nil{%
575   \def\@tempa{#3}\ifx\@tempa\@empty \else
576     \caption@g@addto@list\caption@lsep@#1}%
577   \fi}
578 \@onlypreamble\caption@@declarelabelseparator

579 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
580 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}

```

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

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

```

581 \newcommand*{\caption@setlabelseparator}[1]{%
582   \ifundefined{caption@lsep@#1}%
583     {\caption@Error{Undefined label separator `#1'}}%

```

```

584     {\edef\caption@lsepname{#1}%
585     \expandafter\let\expandafter\caption@iflfl\csname caption@iflfl@#1\endcsname
586     \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’.

```

587 \DeclareCaptionLabelSeparator{none}{}
588 \DeclareCaptionLabelSeparator{colon}{: }
589 \DeclareCaptionLabelSeparator{period}{. }
590 \DeclareCaptionLabelSeparator{space}{ }
591 \DeclareCaptionLabelSeparator*{quad}{\quad}
592 \DeclareCaptionLabelSeparator*{newline}{\\}
593 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

```

`\caption@setdefaultlabelsep`

```

594 \newcommand*\caption@setdefaultlabelsep[1]{%
595   \ifx\caption@lsep\caption@lsep@default
596     \caption@set@default@labelsep{#1}%
597     \caption@setlabelseparator{default}%
598   \else
599     \caption@set@default@labelsep{#1}%
600   \fi}

601 \newcommand*\caption@set@default@labelsep[1]{%
602   \def\caption@lsep@default{\@nameuse{caption@lsep@#1}}%
603   \def\caption@iflfl@default{\@nameuse{caption@iflfl@#1}}}

```

‘default’ usually maps to ‘colon’.

```

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

```

11 Text formats

```

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

605 \newcommand*\DeclareCaptionTextFormat[2]{%
606   \global\long\@namedef{caption@tfmt@#1}##1{#2}}
607 \onlypreamble\DeclareCaptionTextFormat

608 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
609 \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`).

```

610 \newcommand*\caption@settextformat[1]{%
611   \@ifundefined{caption@tfmt@#1}%
612     {\caption@Error{Undefined text format `#1'}}%
613     {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

614 \DeclareCaptionTextFormat{empty}{}
615 \DeclareCaptionTextFormat{simple}{#1}
616 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

12 Fonts

```

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

DeclareCaptionDefaultFont \DeclareCaptionDefaultFont{<name>}{<code>}
621 \newcommand*\DeclareCaptionDefaultFont[2]{%
622   \global\@namedef{caption#1@default}{#2}}
623 \@onlypreamble\DeclareCaptionDefaultFont

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

627 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
628 \DeclareCaptionOption{labelfont+}{\caption@addtofont{labelfont}{#1}}
629 \DeclareCaptionDefaultFont{labelfont}{}

630 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
631 \DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
632 \DeclareCaptionDefaultFont{textfont}{}

\caption@setfont \caption@setfont{<name>}{<keyval-list of names>}
Selecting a caption font means saving all the code snippets in \caption{<name>}.
633 \newcommand*\caption@setfont[1]{%
634   \expandafter\let\csname caption#1\endcsname\@empty
635   \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.
636 \newcommand*\caption@addtofont[2]{%
637   \begingroup
638     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
639     \define@key{caption@fnt}{default}[]{%
640       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
641     \caption@setkeys[caption]{caption@fnt}{#2}%
642     \global\let\caption@tempa\caption@fnt
643   \endgroup
644   \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.
645 \newcommand*\caption@font{%
646   \caption@teststar\caption@@font\@firstofone
647   {\caption@setkeys[caption]{caption@fnt}}
648 \newcommand*\caption@@font[2]{%
649   \begingroup
650   \def\caption@fnt{\endgroup}%
651   #1{#2}%
652   \caption@fnt}

```

These are the pre-defined font code snippets.

```

653 \DeclareCaptionFont{normalcolor}{\normalcolor}
654 \DeclareCaptionFont{color}{\color{#1}}

655 \DeclareCaptionFont{normalfont}{\normalfont}
656 \DeclareCaptionFont{up}{\upshape}
657 \DeclareCaptionFont{it}{\itshape}
658 \DeclareCaptionFont{sl}{\slshape}
659 \DeclareCaptionFont{sc}{\scshape}
660 \DeclareCaptionFont{md}{\mdseries}
661 \DeclareCaptionFont{bf}{\bfseries}
662 \DeclareCaptionFont{rm}{\rmfamily}
663 \DeclareCaptionFont{sf}{\sffamily}
664 \DeclareCaptionFont{tt}{\ttfamily}

665 \DeclareCaptionFont{scriptsize}{\scriptsize}
666 \DeclareCaptionFont{footnotesize}{\footnotesize}
667 \DeclareCaptionFont{small}{\small}
668 \DeclareCaptionFont{normalsize}{\normalsize}
669 \DeclareCaptionFont{large}{\large}
670 \DeclareCaptionFont{Large}{\Large}

671 \DeclareCaptionFont{sansmath}{\sansmath}

672 \DeclareCaptionFont{singlespacing}{%
673   \caption@ifundefined{setstretch}{\setstretch{1}}{
674     \setstretch\setstretch{#1}}}% normally 1
675 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
676 \DeclareCaptionFont{doublespacing}{\doublespacing}
677 \DeclareCaptionFont{stretch}{\setstretch{#1}}

678 %\DeclareCaptionFont{normal}{%
679 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
680 \DeclareCaptionFont{normal}{%
681   \caption@font*{%
682     \KV@caption@fnt@normalcolor\@unused
683     \KV@caption@fnt@normalfont\@unused
684     \KV@caption@fnt@normalsize\@unused
685     \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.

```

686 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
687 \DeclareCaptionDefaultFont{size}{}

```

13 Justifications

```

\DeclareCaptionJustification{\name}{\code}

688 \newcommand*\DeclareCaptionJustification[2]{%
689   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v1.0
690   \DeclareCaptionFont{#1}{#2}}
691 \@onlypreamble\DeclareCaptionJustification

\DeclareCaptionDefaultJustification{\code}

692 \newcommand*\DeclareCaptionDefaultJustification[1]{%

```

```

693 \global\@namedef{caption@hj@default}{#1}% for compatibility to v1.0
694 \DeclareCaptionDefaultFont{@hj}{#1}
695 \@onlypreamble\DeclareCaptionDefaultJustification

696 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
697 \DeclareCaptionDefaultJustification{}

```

```

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

```

These are the pre-defined justification code snippets.

```

699 \DeclareCaptionJustification{justified}{}
700 \DeclareCaptionJustification{centering}{\centering}
701 \DeclareCaptionJustification{centerfirst}{\centerfirst}
702 \DeclareCaptionJustification{centerlast}{\centerlast}
703 \DeclareCaptionJustification{raggedleft}{\raggedleft}
704 \DeclareCaptionJustification{raggedright}{\raggedright}

```

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

```

705 \providecommand\centerfirst{%
706   \let\\\@centercr
707   \edef\caption@normaladjust{%
708     \leftskip\the\leftskip
709     \rightskip\the\rightskip
710     \parfillskip\the\parfillskip\relax}%
711   \leftskip\z@\@plus -1fil%
712   \rightskip\z@\@plus 1fil%
713   \parfillskip\z@skip
714   \noindent\hskip\z@\@plus 2fil%
715   \@setpar{\@@par\@restorepar\caption@normaladjust}}

```

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

```

716 \providecommand\centerlast{%
717   \let\\\@centercr
718   \leftskip\z@\@plus 1fil%
719   \rightskip\z@\@plus -1fil%
720   \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.

```

721 \DeclareCaptionJustification{Centering}{%
722   \caption@ragged\Centering\centering}
723 \DeclareCaptionJustification{RaggedLeft}{%
724   \caption@ragged\RaggedLeft\raggedleft}
725 \DeclareCaptionJustification{RaggedRight}{%
726   \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.

```

727 \newcommand*\caption@ragged{%
728   \caption@Debug{We need ragged2e}%
729   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
730   \global\let\caption@ragged\caption@@ragged
731   \caption@ragged}

732 \caption@AtBeginDocument{%
733   \@ifundefined{caption@r@ragged2e}{%
734     \newcommand*\caption@@ragged{%
735       \caption@Warning{%
736         'ragged2e' support has been changed.\MessageBreak
737         Rerun to get captions right}%
738       \global\let\caption@ragged\@secondoftwo % suppress further warnings
739       \caption@ragged}%
740   }{%
741     \caption@Debug{We load ragged2e}%
742     \IfFileExists{ragged2e.sty}{%
743       \RequirePackage{ragged2e}%
744       \let\caption@@ragged\@firstoftwo
745     }{%
746       \newcommand*\caption@@ragged[2]{%
747         \@ifundefined{caption\string#1}{%
748           \caption@Warning{%
749             'ragged2e' package not loaded, therefore\MessageBreak
750             substituting \string#2 for \string#1\MessageBreak}%
751           \global\@namedef{caption\string#1}{}%
752           #2}%
753         }%
754       }%

```

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
755 \caption@ifundefined\abovecaptionskip{%
756   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p{}}%
757 \caption@ifundefined\belowcaptionskip{%
758   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p{}}%

759 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
760 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
761 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

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

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

```

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

```

```

763 \newcommand*\caption@hrule{\hrule\@height\z@}
764 \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.

```

765 \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.*

```

766 \newcommand*\caption@setposition[1]{%
767   \caption@ifinlist{#1}{d,default}{%
768     \let\caption@position\caption@defaultpos
769   }{\caption@ifinlist{#1}{t,top,above}{%
770     \let\caption@position\@firstoftwo
771   }{\caption@ifinlist{#1}{b,bottom,below}{%
772     \let\caption@position\@secondoftwo
773   }{\caption@ifinlist{#1}{a,auto}{%
774     \let\caption@position\@undefined
775   }{%
776     \caption@Error{Undefined position `#1'}%
777   }}}}%

```

`\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 \LaTeX document classes, but ‘auto’ should give better results in most cases.

```

778 %\caption@setdefaultpos{a}% default = auto
779 \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.)

```

780 \newcommand*\caption@iftop{%
781   \ifx\caption@position\@undefined
782     \let\caption@position\@secondoftwo
783 %   = \caption@setposition b%
784   \fi
785   \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.

```

786 \newcommand*\caption@fixposition{%

```

```

787 \ifx\caption@position\@undefined
788 \caption@autoposition
789 \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^[4] 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.

```

790 \newcommand*\caption@autoposition{%
791   \ifvmode
792     \edef\caption@tempa{\the\prevdepth}%
793     \caption@Debug{\protect\prevdepth=\caption@tempa}%
794     \ifdim\prevdepth>-\p@
795       \let\caption@position\@secondoftwo
796     \else
797       \let\caption@position\@firstoftwo
798     \fi
799 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
800 \else
801   \caption@Debug{no \protect\prevdepth}%
802   \let\caption@position\@secondoftwo
803 %   = \caption@setposition b%
804 \fi}

```

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

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

```

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

```

16 Hooks

```

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

```

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

```

807 \newcommand*\caption@beginhook{}
808 \newcommand*\caption@endhook{}
809 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
810 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

17 Lists

```

811 \DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
812 \DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}

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

```

```

814 \DeclareCaptionOption{listtype}{\caption@setlisttype{#1}}
815 \DeclareCaptionOption{listtype+}{\caption@setlisttype@ext{#1}}

\caption@setlisttype \caption@setlisttype{<type>}
816 \newcommand*\caption@setlisttype{%
817   \caption@setlisttype@ext{}%
818   \caption@@setlisttype\caption@listtype}

819 \newcommand*\caption@@setlisttype[2]{%
820   \edef#1{#2}%
821   \ifx#1\@empty \let#1\@undefined \fi}

\caption@setlisttype@ext \caption@setlisttype@ext{<type extension>}
822 \newcommand*\caption@setlisttype@ext{%
823   \caption@@setlisttype\caption@listtype@ext}

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

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

\caption@setlistformat \caption@setlistformat{<name>}
Selecting a caption list format simply means saving the code (in \caption@lstfmt).

828 \newcommand*\caption@setlistformat[1]{%
829   \ifundefined{caption@lstfmt@#1}%
830     {\caption@Error{Undefined list format `#1'}}%
831     {\expandafter\let\expandafter\caption@lstfmt
832       \csname caption@lstfmt@#1\endcsname}}

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

833 \DeclareCaptionListFormat{empty}{}
834 \DeclareCaptionListFormat{simple}{#1#2}
835 \DeclareCaptionListFormat{parens}{#1(#2)}
836 \DeclareCaptionListFormat{subsimple}{#2}
837 \DeclareCaptionListFormat{subparens}{(#2)}

\caption@setdefaultlistformat

838 \newcommand*\caption@setdefaultlistformat[1]{%
839   \ifx\caption@lstfmt\caption@lstfmt@default
840     \caption@set@default@listformat{#1}%
841     \caption@setlistformat{default}%
842   \else
843     \caption@set@default@listformat{#1}%
844   \fi}

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

‘default’ usually maps to ‘subsimple’.

847 \caption@set@default@listformat{subsimple}

```

18 Debug option

```
848 \DeclareCaptionOption{debug}[1]{%
849   \caption@set@bool\caption@ifdebug{#1}%
850   \caption@ifdebug
851     {\let\caption@Debug\caption@Info}%
852     {\let\caption@Debug@gobble}}
853 \DeclareOption{debug}{\setkeys{caption}{debug}}
854 \setkeys{caption}{debug=0}
```

19 Document classes & Babel support

19.1 The standard L^AT_EX classes

```
855 \caption@CheckCommand\@makecaption{%
856   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
857   \long\def\@makecaption#1#2{%
858     \vskip\abovecaptionskip
859     \sbox\@tempboxa{#1: #2}%
860     \ifdim \wd\@tempboxa >\hsize
861       #1: #2\par
862     \else
863       \global \@minipagefalse
864       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
865     \fi
866     \vskip\belowcaptionskip}}
```

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

\caption@ifamsclass

```
867 \providecommand*\caption@ifamsclass{%
868   \caption@ifundefined\@captionheadfont\@gobble\@firstofone}
869 \@onlypreamble\caption@ifamsclass

870 \caption@ifamsclass{%
871   \caption@CheckCommand\@makecaption{%
872     % amsart|amsproc|amsbook [2004/08/06 v2.20]
873     \long\def\@makecaption#1#2{%
874       \setbox\@tempboxa\vbox{\color@setgroup
875         \advance\hsize-2\captionindent\noindent
876         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
877           {\@cdr#2\@nil}\@captionfont\upshape\enspace#2}%
878       \unskip\kern-2\captionindent\par
879       \global\setbox\@ne\lastbox\color@endgroup}%
880     \ifhbox\@ne % the normal case
881       \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
882     \fi
883     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
884       \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
885     \else % tempboxa contained more than one line
886       \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
887         \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
888     \fi
889     \ifnum\@tempcnta<64 % if the float IS a figure...
```

```

890         \addvspace\abovecaptionskip
891         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
892     \else % if the float IS NOT a figure...
893         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
894         \nobreak
895         \vskip\belowcaptionskip
896     \fi
897 \relax
898 }}

899 \caption@CheckCommand\@makecaption{%
900 % smfart|smfbook [1999/11/15 vl.2f Classe LaTeX pour les articles publies par
901 \long\def\@makecaption#1#2{%
902     \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
903     \setbox\@tempboxa\vbox{\color@setgroup
904         \advance\hsize-2\captionindent\noindent
905         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
906             {\@cdr#2\@nil}\{\@addpunct{.}\@captionfont\upshape\enspace#2}%
907         \unskip\kern-2\captionindent\par
908         \global\setbox\@ne\lastbox\color@endgroup}%
909     \ifhbox\@ne % the normal case
910         \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
911     \fi
912     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
913         \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
914         \@tempdima\wd\@ne\advance\@tempdima-\captionindent
915         \wd\@ne\@tempdima
916     \else % tempboxa contained more than one line
917         \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
918             \unvbox\@tempboxa\parskip\z@skip
919             \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
920     \fi
921     \ifnum\@tempcnta<64 % if the float IS a figure...
922         \addvspace\abovecaptionskip
923         \noindent\kern\captionindent\box\@ne
924     \else % if the float IS NOT a figure...
925         \noindent\kern\captionindent\box\@ne
926         \nobreak
927         \vskip\belowcaptionskip
928     \fi
929 \relax
930 }}

931 \let\captionmargin\captionindent % set to 3pc by AMS class
932 \begingroup\edef\@tempa{\endgroup
933     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
934     {margin=\the\captionmargin
935     \caption@ifundefined\smf@makecaption{{},maxmargin=.1\linewidth}}}}
936 \@tempa
937 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
938 \DeclareCaptionLabelSeparator{default}{.\enspace}
939 \DeclareCaptionDefaultFont{font}{\@captionfont}
940 \DeclareCaptionDefaultFont{labelfont}{\@captionheadfont}
941 \DeclareCaptionDefaultFont{textfont}{\@captionfont\upshape}
942 \captionsetup[figure]{position=b}

```

```

943 \captionsetup[table]{position=t}
944 }

```

19.3 The beamer class (Part one)

\caption@ifbeamerclass

```

945 \providecommand*\caption@ifbeamerclass{%
946 \ifclassloaded{beamer}\@firstofone\@gobble}
947 \onlypreamble\caption@ifbeamerclass

948 \caption@ifbeamerclass{%
949 \caption@CheckCommand\beamer@makecaption{%
950 % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
951 \long\def\beamer@makecaption#1#2{%
952 \def\insertcaptionname{\csname#1name\endcsname}%
953 \def\insertcaptionnumber{\csname the#1\endcsname}%
954 \def\insertcaption{#2}%
955 \nobreak\vskip\abovecaptionskip\nobreak
956 \sbox\@tempboxa{\usebeamertemplate**{caption}}%
957 \ifdim \wd\@tempboxa >\hsize
958 \usebeamertemplate**{caption}\par
959 \else
960 \global \@minipagefalse
961 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
962 \fi
963 \nobreak\vskip\belowcaptionskip\nobreak}}

```

\caption@ifbeamertemplate

```

964 \newcommand*\caption@ifbeamertemplate[1]{%
965 \begingroup
966 \let\beamer@@tmpl\caption@ORI\beamer@@tmpl\caption
967 \@nameuse{beamer@@tmpop\caption@#1}%
968 \ifx\beamer@@tmpl\caption@ORI\beamer@@tmpl\caption
969 \endgroup\expandafter\@firstoftwo
970 \else
971 \endgroup\expandafter\@secondoftwo
972 \fi}

973 \DeclareCaptionLabelFormat{default}{%
974 #1\caption@ifbeamertemplate{numbered}{~#2}}
975 \caption@declarelabeleseparator
976 {\caption@ifbeamertemplate{caption name own line}\@gobble\@firstofone}
977 {default}
978 {\caption@ifbeamertemplate{caption name own line}{\:\ }}
979 \DeclareCaptionDefaultFont{font}{%
980 \usebeamerfont*{caption}%
981 \usebeamerfont*{caption}}
982 \DeclareCaptionDefaultFont{labelfont}{%
983 \usebeamerfont*{caption name}%
984 \usebeamerfont*{caption name}}
985 \DeclareCaptionDefaultJustification{\raggedright}
986 \DeclareOption{beamerclass}{%
987 \renewcommand\caption@ifslc{%

```

```

988     \caption@ifbeamertemplate{caption name own line}\@secondoftwo\@firstoftwo}
989     % Since the beamer class do not offer a 'list of figures' we switch this supp
990     \captionsetup{list=0}}
991     \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.)

```

992     \defbeamertemplate{caption}{caption3}{%
993         \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
994     \DeclareOption{beamer}{%
995         % \usebeamertemplate**{caption} will set font
996         \DeclareCaptionDefaultFont{font}{}%
997         \setbeamertemplate{caption}[caption3]}
998 %
999 %     \begin{macrocode]
1000 }

```

19.4 The KOMA-Script classes

`\caption@ifkomaaclass`

```

1001 \providecommand*\caption@ifkomaaclass{%
1002     \caption@ifundefined\scr@caption\@gobble\@firstofone}
1003 \@onlypreamble\caption@ifkomaaclass

1004 \caption@ifkomaaclass{%
1005     \caption@CheckCommand\@makecaption{%
1006         % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
1007         \long\def\@makecaption#1#2{%
1008             \if@captionabove
1009                 \vskip\belowcaptionskip
1010             \else
1011                 \vskip\abovecaptionskip
1012             \fi
1013             \@@makecaption\@firstofone{#1}{#2}%
1014             \if@captionabove
1015                 \vskip\abovecaptionskip
1016             \else
1017                 \vskip\belowcaptionskip
1018             \fi}}
1019     \DeclareCaptionFormat{default}[#1#2#3\par]{%
1020         \ifdofullc@p
1021             \caption@ifin@list\caption@lsep@crlist\caption@lsepname
1022             {\caption@Error{%
1023                 The option 'labelsep=\caption@lsepname' does not work\MessageBreak
1024                 with \noexpand\setcaphanging (which is set by default)}}%
1025             {\caption@fmt@hang{#1}{#2}{#3}}%
1026         \else
1027             #1#2%
1028             \ifdim\cap@indent<\z@
1029                 \par
1030                 \noindent\hspace*{-\cap@indent}%

```



```

1031     \else\if@capbreak
1032     \par
1033     \fi\fi
1034     #3\par
1035     \fi}
1036 \DeclareCaptionLabelSeparator{default}{\captionformat}
1037 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
1038 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
1039 }

```

19.5 The NTG Dutch classes

\caption@ifntgclass

```

1040 \providecommand*\caption@ifntgclass{%
1041   \caption@ifundefined\CaptionFonts\@gobble\@firstofone}
1042 \@onlypreamble\caption@ifntgclass

1043 \caption@ifntgclass{%
1044   \caption@CheckCommand\@makecaption{%
1045     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
1046     \long\def\@makecaption#1#2{%
1047       \vskip\abovecaptionskip
1048       \sbox\@tempboxa{\@CaptionLabelFont#1:} \CaptionTextFont#2}%
1049       \ifdim \wd\@tempboxa >\hsize
1050         {\@CaptionLabelFont#1:} \CaptionTextFont#2\par
1051       \else
1052         \global \@minipagefalse
1053         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1054       \fi
1055       \vskip\belowcaptionskip}}
1056 \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
1057 \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
1058 }

```

19.6 The thesis class

\caption@ifthesisclass

```

1059 \providecommand*\caption@ifthesisclass{%
1060   \caption@ifundefined\cph@font
1061   {\@gobble}%
1062   {\caption@ifundefined\cpb@font\@gobble\@firstofone}}

1063 \caption@ifthesisclass{%
1064   \caption@CheckCommand\@makecaption{%
1065     % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
1066     \long\def\@makecaption#1#2{%
1067       \vskip\abovecaptionskip
1068       \setbox\@tempboxa\hbox{\@cph@font #1:} {\@cpb@font #2}}%
1069       \ifdim \wd\@tempboxa >\hsize
1070         \@hangfrom{\@cph@font #1:} {\@cpb@font #2\par}%
1071       \else

```

```

1072         \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
1073     \fi
1074     \vskip\belowcaptionskip}}

1075 \DeclareCaptionDefaultFormat{hang}
1076 \DeclareCaptionDefaultFont{labelfont}{\cph@font}
1077 \DeclareCaptionDefaultFont{textfont}{\cpb@font}
1078 }

```

19.7 The frenchb Babel option

```

1079 \caption@ifundefined\FB@makecaption{}{%
1080     \caption@CheckCommand\@makecaption{%
1081         % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
1082         % frenchb.ldf [2007/10/05 v2.0e French support from the babel system]
1083         \long\def\@makecaption#1#2{%
1084             \vskip\abovecaptionskip
1085             \sbox\@tempboxa{#1\CaptionSeparator #2}%
1086             \ifdim \wd\@tempboxa >\hsize
1087                 #1\CaptionSeparator #2\par
1088             \else
1089                 \global \@minipagefalse
1090                 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1091             \fi
1092             \vskip\belowcaptionskip}}

1093 \ifx\@makecaption\STD@makecaption
1094     \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
1095     \def\caption@frenchb{% supress frenchb warning
1096         \let\STD@makecaption\@makecaption
1097         \let\FB@makecaption\@makecaption}
1098 \else
1099     \ifx\@makecaption\@undefined\else
1100         \caption@InfoNoLine{%
1101             The definition of \protect\@makecaption\space
1102             has been changed,\MessageBreak
1103             frenchb will NOT customize it}%
1104     \fi
1105 \fi

1106 }

```

19.8 The frenchle/pro package

```

1107 \caption@ifundefined\frenchTeXmods{}{%
1108     \caption@CheckCommand\@makecaption{%
1109         % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
1110         % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
1111         \def\@makecaption#1#2{%
1112             \ifFTY%
1113                 \def\@secondofmany##1##2\void{##2}%
1114                 \def\@tempa{\@secondofmany#2\void}%
1115                 \ifx\@tempa\empty%
1116                     \let\captionseparator\empty%
1117                 \fi%

```

```

1118     \@mcORI{#1}{\relax\captionfont{#2}}%
1119     \else
1120     \@mcORI{#1}{#2}%
1121     \fi}}%

1122 \caption@CheckCommand\@makecaption{%
1123   % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
1124   \def\@makecaption#1#2{%
1125     \ifFTY%
1126     \def\@secondofmany##1##2\void{##2}%
1127     \protected@edef\@tempa{\@secondofmany#2\void}%
1128     \ifx\@tempa\empty%
1129     \let\captionseparator\empty%
1130     \fi%
1131     \@mcORI{#1}{\relax\captionfont{#2}}%
1132     \else
1133     \@mcORI{#1}{#2}%
1134     \fi}}%

1135 \DeclareCaptionDefaultFont{textfont}{\itshape}%
1136 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%

1137 }

```

19.9 The hungarian and magyar Babel option

```

1138 \DeclareCaptionListFormat{subperiod}{#2.}

1139 \caption@ifundefined\hunnewlabel{}{%
1140   \caption@CheckCommand\@makecaption{%
1141     % magyar.1df [2005/03/30 v1.4j Magyar support from the babel system]
1142     \def\@makecaption#1#2{%
1143       \vskip\abovecaptionskip
1144       \sbox\@tempboxa{#1. #2}%
1145       \ifdim \wd\@tempboxa >\hsize
1146       {#1. #2\csize par\endcsize}
1147       \else
1148       \global \@minipagefalse
1149       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1150       \fi
1151       \vskip\belowcaptionskip}}}%

1152 \def\caption@tempa#1{\@ifundefined{extras#1}}{%
1153   \expandafter\addto\csize extras#1\endcsize{%
1154     % change default labelsep and listformat
1155     \caption@setdefaultlabelsep{period}%
1156     \caption@setdefaultlistformat{subperiod}}}%
1157 \expandafter\addto\csize noextras#1\endcsize{%
1158   % change default labelsep and listformat
1159   \caption@setdefaultlabelsep{colon}%
1160   \caption@setdefaultlistformat{subsimple}}}%

1161 }}

1162 \caption@tempa{hungarian}
1163 \caption@tempa{magyar}

```

19.10 Unknown document class (or package)

```

1164 \caption@ifcheckcommand{%
1165   \caption@setbool{documentclass}{1}%
1166 }{%
1167   \caption@setbool{documentclass}{0}%
1168   \caption@info{no line}%
1169   Unknown document class (or package), \MessageBreak
1170   standard defaults will be used}%
1171 \caption@debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1172 }

```

20 Execution of options

```

1173 \captionsetup{style=default,position=default,%
1174               list,listformat=default,twoside=\if@twoside 1\else 0\fi}
1175 \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`.

```

1176 \newcommand\caption@addcontentsline[2]{%
1177   \caption@ifcontentsline{#2}{%
1178     \begingroup
1179     \let\@tempa\@gobble
1180     \caption@ifundefined\caption@listtype
1181     {\edef\caption@listtype{#1}}%
1182     {\let\@tempa\@firstofone}%
1183     \caption@ifundefined\caption@listtype@ext
1184     {}%
1185     {\edef\caption@listtype{\caption@listtype\caption@listtype@ext}%
1186      \let\@tempa\@firstofone}%
1187     \@tempa
1188     {\caption@debug{addcontentsline: #1 => \caption@listtype}%
1189     \caption@setoptions*\caption@listtype
1190     \@namedef{the\caption@listtype}{\@nameuse{the#1}}}%
1191     \expandafter\caption@@addcontentsline\expandafter{\caption@listtype}{#2}%
1192     \endgroup}}
1193 \newcommand\caption@@addcontentsline[2]{%
1194   {\let\space
1195    \@ifundefined{ext@#1}%
1196    {\caption@error{No float type ‘#1’ defined}}%
1197    {\caption@@@addcontentsline
1198     {\csname ext@#1\endcsname}%
1199     {#1}%
1200     {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1201     {\ignorespaces #2}}}}
1202 \newcommand*\caption@@@addcontentsline[4]{%
1203   \addcontentsline{#1}{#2}{\protect\numberline{#3}{#4}}}
1204 \newcommand\caption@ifcontentsline[1]{%
1205   \caption@iflist
1206   {\def\@tempa{#1}}%

```

```

1207     {\let\@tempa\@empty}%
1208     \ifx\@tempa\@empty
1209         \expandafter\@gobble
1210     \else
1211         \expandafter\@firstofone
1212     \fi}

```

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.)
1213 \newif\ifcaption@star

\caption@fnum    \caption@fnum{<float type>}
                  Typesets the caption label; as replacement for \fnum{<float type>}.
1214 \newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}

\caption@make    \caption@make{<float name>}{<ref. number>}{<text>}
                  Typesets the caption.
1215 \newcommand\caption@make[2]{\caption@@make{\caption@lfmt{#1}{#2}}}

\caption@@make   \caption@@make{<caption label>}{<caption text>}
1216 \newcommand\caption@@make[2]{%
1217     \begingroup
1218     \caption@stepcounter
1219     \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
1220 % TODO: Move this to \caption@calcmargin!?
1221     \ifx\caption@maxmargin\undefined \else
1222         \ifdim\captionmargin>\caption@maxmargin\relax
1223             \captionmargin\caption@maxmargin\relax
1224         \fi
1225     \fi
1226     \ifx\caption@minmargin\undefined \else
1227         \ifdim\captionmargin<\caption@minmargin\relax
1228             \captionmargin\caption@minmargin\relax
1229         \fi
1230     \fi

Special single-line treatment (option singlelinecheck=)
1231     \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%

Typeset the left margin (option margin=)
1232     \caption@calcmargin
1233     \@tempdima\captionmargin
1234     \ifdim\captionmargin@=\z@ \else
1235         \caption@ifoddpage{}{\advance\@tempdima\captionmargin@}%
1236     \fi
1237     \caption@ifh{\advance\@tempdima\caption@indent}%
1238     \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*`.

```
1239 \@tempdima\captionwidth
1240 \caption@ifh{\advance\@tempdima-\caption@indent}%
1241 \caption@parbox\@tempdima{%
```

Typeset the indentation (option `indention=`)

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

```
1242 \caption@ifh{%
1243 \ifdim\caption@indent=\z@
1244 \leavevmode
1245 \else
1246 \hskip-\caption@indent
1247 \fi}%
```

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

```
1248 \caption@@@make{#1}{#2}}%
```

Typeset the right margin (option `margin=`)

```
1249 \@tempdima\captionmargin
1250 \ifdim\captionmargin=\z@ \else
1251 \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
1252 \fi
1253 \hspace\@tempdima
1254 \caption@endhook
1255 \endgroup
1256 \global\caption@starfalse}
```

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

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

```
1257 \newcommand*\caption@calcmargin{%
1258 \caption@calcmargin@hook
1259 \ifdim\captionwidth=\z@
1260 \captionwidth\linewidth
1261 \advance\captionwidth by -2\captionmargin
1262 \advance\captionwidth by -\captionmargin@
1263 \else
1264 \captionmargin\linewidth
1265 \advance\captionmargin by -\captionwidth
1266 \divide\captionmargin by 2
1267 \captionmargin@\z@
1268 \fi

1269 \caption@Debug{%
1270 \string\hsize=\the\hsize,
1271 \string\linewidth=\the\linewidth,\MessageBreak
1272 \string\leftmargin=\the\leftmargin,
1273 \string\rightmargin=\the\rightmargin,\MessageBreak
1274 \string\margin=\the\captionmargin,
1275 \string\margin@=\the\captionmargin@,
1276 \string\width=\the\captionwidth}%
1277 }
```

```

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.
1278 \newcommand\caption@slc[4]{%
1279 \caption@@slc{#1}{#2}{#3}{\caption@singleline#4}{}}
1280 \newcommand\caption@@slc[5]{%
1281 \caption@Debug{Begin SLC}%
1282 \begingroup
1283 \caption@singleline
1284 \let\caption@hj\@empty
1285 \caption@calcmargin % calculate #3 if necessary
1286 \caption@prepareslc
1287 \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1288 \ifdim\wd\@tempboxa>#3%
1289 \endgroup
1290 #5%
1291 \else
1292 \endgroup
1293 #4%
1294 \fi
1295 \caption@Debug{End SLC}}
1296 \newcommand*\caption@singleline{%
1297 \caption@xsetup\caption@opt@singleline
1298 \let\caption@fmt\caption@slfmt}

```

```

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1299 \newcommand*\caption@prepareslc{%
1300 \let\label\caption@gobble
1301 \let\caption@footnote@ORI\footnote
1302 \def\footnote{\caption@withoptargs\caption@footnote}%
1303 \let\caption@footnotemark@ORI\footnotemark
1304 \def\footnotemark{\caption@withoptargs\caption@footnotemark}%
1305 \let\@footnotetext\@gobble
1306 \let\@endnotetext\@gobble}
1307 \newcommand\caption@footnote[2]{%
1308 \begingroup
1309 \let\stepcounter\caption@l@stepcounter
1310 \caption@footnote@ORI#1{#2}%
1311 \endgroup}
1312 \newcommand\caption@footnotemark[1]{%
1313 \begingroup
1314 \let\stepcounter\caption@l@stepcounter
1315 \caption@footnotemark@ORI#1%
1316 \endgroup}
1317 \newcommand*\caption@l@stepcounter[1]{%
1318 \advance\c@#1\endcsname\@ne\relax}

```

```

\caption@parbox \caption@parbox{<width>}{<contents>}
This macro defines the box which surrounds the caption paragraph.
1319 \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=.
1320 \newcommand*\caption@applyfont{%
1321   \caption@hj\captionfont\captionsize}

\caption@@@make \caption@@@make{\langle caption label \rangle}{\langle caption text \rangle}
This one finally typesets the caption paragraph, without margin and indentation.
1322 \newcommand\caption@@@make[2]{%
If the label is empty, we use no caption label separator.
1323   \sbox\@tempboxa{\#1}%
1324   \ifdim\wd\@tempboxa=\z@
1325     \let\caption@lsep\relax
1326 %   \@capbreakfalse
1327   \fi
If the text is empty, we use no caption label separator, too. (And no text format either.)
1328   \caption@ifempty{\#2}{%
1329     \let\caption@lsep\@empty
1330     \let\caption@tfmt\@firstofone
1331 %   \@capbreakfalse
1332 %   \let\caption@ifstrut\@secondoftwo
1333   }%
Take care that \caption@parindent and \caption@hangindent will be used
to typeset the paragraph.
1334   \@setpar{\@@par\caption@@par}\caption@@par
Finally typeset the caption.
1335   \caption@applyfont
1336   \caption@fmt
1337   {\ifcaption@star\else{\captionlabelfont\#1}\fi}%
1338   {\ifcaption@star\else{\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1339   {\caption@textfont
1340     \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1341     \nobreak\hskip\z@skip % enable hyphenation
1342     \caption@tfmt{\#2}%
1343 %   \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1344   \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1345   \par}}

\caption@ifempty \caption@ifempty{\langle text \rangle}{\langle true \rangle} (no \langle false \rangle)
This one tests if the \langle text \rangle 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.
1346 \newcommand\caption@ifempty[1]{%
1347   \caption@ifempty{\#1}%
1348   \caption@ifempty\@unused}

1349 \newcommand\caption@ifempty[1]{%
1350   \def\caption@tempa{\#1}%
1351   \ifx\caption@tempa\@empty
1352     \let\caption@ifempty\@secondoftwo

```



```

1353 \else
1354 \expandafter\def\expandafter\caption@tempa\expandafter{%
1355 \caption@car#1\caption@if@empty\caption@nil}%
1356 \def\caption@tempb{\caption@if@empty}%
1357 \ifx\caption@tempa\caption@tempb
1358 \let\caption@if@empty\@secondoftwo
1359 \else
1360 \def\caption@tempb{\ignorespaces}%
1361 \ifx\caption@tempa\caption@tempb
1362 \expandafter\caption@if@empty\expandafter{\@gobble#1}%
1363 \else
1364 \def\caption@tempb{\label}%
1365 \ifx\caption@tempa\caption@tempb
1366 \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1367 \else
1368 \def\caption@tempb{\index}%
1369 \ifx\caption@tempa\caption@tempb
1370 \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1371 \else
1372 \def\caption@tempb{\glossary}%
1373 \ifx\caption@tempa\caption@tempb
1374 \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1375 \else
1376 \let\caption@if@empty\@gobbletwo
1377 \fi
1378 \fi
1379 \fi
1380 \fi
1381 \fi
1382 \fi}

```

```

1383 \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.

```

1384 \newcommand*\caption@@par{%
1385 \parindent\caption@parindent\hangindent\caption@hangindent}%

```

23 Types & sub-types

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

```

1386 \newcommand*\DeclareCaptionType{%
1387 \@testopt\@DeclareCaptionType{}%
1388 \@onlypreamble\DeclareCaptionType
1389 \def\@DeclareCaptionType[#1]#2{%
1390 \def\caption@type{#2}%
1391 \caption@Debug{New type `#2'}%
1392 \newcounter{#2}\@namedef{theH#2}{}%
1393 \KV@caption@DCT@within\caption@within@default
1394 \caption@DeclareWithinOption{#2}%
1395 \KV@caption@DCT@placement{tbp}%
1396 \caption@ifundefined\c@float@type
1397 {\newcounter{float@type}%

```

```

1398 \setcounter{float@type}{\caption@ifundefined\c@figure14}}%
1399 {}%
1400 \caption@Debug{float type `#2'=\the\value{float@type}}%
1401 \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1402 \addtocounter{float@type}{\value{float@type}}%
1403 \KV@caption@DCT@fileext{lo#2}%
1404 \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1405 \newenvironment{#2}{\@float{#2}}{\end@float}%
1406 \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1407 \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1408 \expandafter\newcommand\csname listof#2es\endcsname{\caption@listof{#2}}%
1409 \caption@ifundefined\l@figure
1410 {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1411 {\expandafter\let\csname l@#2\endcsname\l@figure}%

1412 \expandafter\newcommand\csname #2name\endcsname{}%
1413 \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
1414 \uppercase\expandafter{\@tempa}%
1415 \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
1416 \@tempb
1417 \expandafter\let\csname #2name\endcsname\@tempa
1418 \expandafter\newcommand\csname list#2name\endcsname{}%
1419 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%

1420 \@cons\caption@typelist{#2}%
1421 \caption@setkeys[caption]{caption@DCT}{#1}%

1422 \expandafter\caption@DeclareTypeExt\expandafter{\@nameuse{ext@#2}}{#2}%
1423 \@ifnextchar[{\@@DeclareCaptionType\relax}
1424 \@onlypreamble\@@DeclareCaptionType

1425 \def\@@DeclareCaptionType[#1]{%
1426 \KV@caption@DCT@name{#1}%
1427 \@ifnextchar[{\@@@DeclareCaptionType\relax}
1428 \@onlypreamble\@@@DeclareCaptionType

1429 \def\@@@DeclareCaptionType[#1]{%
1430 \KV@caption@DCT@listname{#1}}
1431 \@onlypreamble\@@@DeclareCaptionType

1432 \let\DeclareFloatingEnvironment\DeclareCaptionType % alternative command name
1433 \@onlypreamble\DeclareFloatingEnvironment

```

\caption@DeclareTypeExt

```

1434 \newcommand*\caption@DeclareTypeExt[2]{%
1435 \caption@ifundefined\@chapter\relax\caption@PatchChapter
1436 \let\caption@PatchChapter\relax

1437 \@ifundefined{caption@ext@#1}{%
1438 \@namedef{caption@ext@#1}{#2}%

```

Support of float package

TODO: Needs revision

```

1439 \caption@ifundefined\float@exts{\newtoks\float@exts}{}%
1440 \let\float@do=\relax
1441 \edef\@tempa{\noexpand\float@exts{\the\float@exts\float@do{#1}}}%
1442 \@tempa
1443 \caption@ifundefined\caption@addtolists{%

```

```

1444 \newcommand\caption@addtolists[1]{%
1445 \def\float@do####1{\addtocontents{####1}{##1}}\the\float@exts}}}%

```

Support of tocbasic package

```

1446 \caption@ifundefined\addtotoclist{}{%
1447 \addtotoclist[float]{#1}%
1448 \setuptoc{#1}{chapteratlist}%
1449 \@namedef{listof#1name}{\list#2name}}%
1450 }{}%

```

Support of titletoc package

```

1451 \caption@ifundefined\contentsuse{}{%
1452 \contentsuse{#2}{#1}}%
1453 }
1454 \@onlypreamble\caption@DeclareTypeExt

```

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

```

1455 \newcommand*\caption@within@default{%
1456 \caption@ifundefined\c@chapter{none}{chapter}}
1457 \@onlypreamble\caption@within@default

```

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

```

1458 \caption@ifundefined\listoftoc{%
1459 \newcommand*\caption@listof[1]{%
1460 \begingroup
1461 \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1462 \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1463 \let\caption@ORI@starttoc\@starttoc
1464 \renewcommand*\@starttoc[1]{%
1465 \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1466 \listoffigures
1467 \endgroup}%
1468 }{%
1469 \newcommand*\caption@listof[1]{%
1470 \expandafter\listoftoc\expandafter{\@nameuse{ext@#1}}}%
1471 }

```

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

```

1472 \newcommand*\caption@typelist{}

```

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

```

1473 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
1474 \@onlypreamble@key{caption@DCT}{fileext}
1475 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1476 \@onlypreamble@key{caption@DCT}{listname}
1477 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1478 \@onlypreamble@key{caption@DCT}{name}
1479 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
1480 \@onlypreamble@key{caption@DCT}{placement}
1481 \define@key{caption@DCT}{within}{%
1482 \caption@ifundefined\c@chapter{}{\@removefromreset\caption@type{chapter}}%
1483 \@removefromreset\caption@type{section}%

```

```

1484 \begingroup
1485 \caption@setkeys[caption]{caption@within}{#1}%
1486 \endgroup
1487 %\@onlypreamble@key{caption@DCT}{within}
1488 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1489 %\@onlypreamble@key{caption@DCT}{without}

1490 \define@key{caption@within}{none}[]{%
1491 \caption@within{}}
1492 %\@onlypreamble@key{caption@within}{none}
1493 \define@key{caption@within}{section}[]{%
1494 \@addtoreset\caption@type{section}%
1495 \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
1496 %\@onlypreamble@key{caption@within}{section}
1497 \caption@ifundefined\c@chapter{}{%
1498 \define@key{caption@within}{chapter}[]{%
1499 \@addtoreset\caption@type{chapter}%
1500 \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
1501 }% \@onlypreamble@key{caption@within}{chapter}}

```

\caption@within \caption@within{*the code*}{*the Hcode*}

```

1502 \newcommand*\caption@within{%
1503 \expandafter\caption@within@\expandafter{\caption@type}}
1504 %\@onlypreamble\caption@within
1505 \newcommand*\caption@within@[3]{%
1506 \global\@namedef{the#1}{#2\arabic{#1}}%
1507 \@ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
1508 {\global\@namedef{theH#1}{#3\arabic{#1}}}%
1509 %\@onlypreamble\caption@within@

```

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

```

1510 \providecommand*\@removefromreset[2]{%
1511 \expandafter\let\csname c@#1\endcsname\@removefromreset
1512 \def\@elt##1{%
1513 \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1514 \else
1515 \noexpand\@elt{##1}%
1516 \fi}%
1517 \expandafter\xdef\csname cl@#2\endcsname{%
1518 \csname cl@#2\endcsname}}

```

\caption@PatchChapter We try to patch \chapter so \caption@addtolists will be supported.

```

1519 \newcommand*\caption@PatchChapter{%
1520 \providecommand*\chapterlistsgap{10\p}%

1521 % report.cls [2005/09/16 v1.4f Standard LaTeX document class]
1522 \caption@patch@chapter{report}{%
1523 \ifnum \c@secnumdepth >\m@ne
1524 \refstepcounter{chapter}%
1525 \typeout{\@chapapp\space\thechapter.}%
1526 \addcontentsline{toc}{chapter}%
1527 {\protect\numberline{\thechapter}##1}%
1528 \else
1529 \addcontentsline{toc}{chapter}{##1}%

```

```

1530 \fi
1531 \chaptermark{##1}%
1532 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1533 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1534 \if@twocolumn
1535 \topnewpage[\@makechapterhead{##2}]%
1536 \else
1537 \makechapterhead{##2}%
1538 \afterheading
1539 \fi
1540 }{%
1541 \ifnum \c@secnumdepth >\m@ne
1542 \refstepcounter{chapter}%
1543 \typeout{\@chapapp\space\thechapter.}%
1544 \addcontentsline{toc}{chapter}%
1545 {\protect\numberline{\thechapter}##1}%
1546 \else
1547 \addcontentsline{toc}{chapter}{##1}%
1548 \fi
1549 \chaptermark{##1}%
1550 \ifdim \@chapterlistsgap>\z@
1551 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1552 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1553 \caption@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1554 \fi
1555 \if@twocolumn
1556 \topnewpage[\@makechapterhead{##2}]%
1557 \else
1558 \makechapterhead{##2}%
1559 \afterheading
1560 \fi}%

1561 % book.cls [2005/09/16 v1.4f Standard LaTeX document class]
1562 \caption@patch@chapter{book}{%
1563 \ifnum \c@secnumdepth >\m@ne
1564 \if@mainmatter
1565 \refstepcounter{chapter}%
1566 \typeout{\@chapapp\space\thechapter.}%
1567 \addcontentsline{toc}{chapter}%
1568 {\protect\numberline{\thechapter}##1}%
1569 \else
1570 \addcontentsline{toc}{chapter}{##1}%
1571 \fi
1572 \else
1573 \addcontentsline{toc}{chapter}{##1}%
1574 \fi
1575 \chaptermark{##1}%
1576 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1577 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1578 \if@twocolumn
1579 \topnewpage[\@makechapterhead{##2}]%
1580 \else
1581 \makechapterhead{##2}%
1582 \afterheading
1583 \fi

```

```

1584 }{%
1585   \ifnum \c@secnumdepth >\m@ne
1586     \if@mainmatter
1587       \refstepcounter{chapter}%
1588       \typeout{\@chapapp\space\thechapter.}%
1589       \addcontentsline{toc}{chapter}%
1590         {\protect\numberline{\thechapter}##1}%
1591     \else
1592       \addcontentsline{toc}{chapter}{##1}%
1593     \fi
1594 \else
1595   \addcontentsline{toc}{chapter}{##1}%
1596 \fi
1597 \chaptermark{##1}%
1598 \ifdim \@chapterlistsgap>\z@
1599   \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1600   \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1601   \caption@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1602 \fi
1603 \if@twocolumn
1604   \@topnewpage[\@makechapterhead{##2}]%
1605 \else
1606   \@makechapterhead{##2}%
1607   \@afterheading
1608 \fi}%
1609 % amsbook.cls [2004/08/06 v2.20]
1610 % smfbook.cls [1999/11/15 v1.2f Classe LaTeX pour les monographies editees par
1611 \caption@patch@chapter{ams/smfbook}{%
1612   \refstepcounter{chapter}%
1613   \ifnum \c@secnumdepth<\z@ \let\@secnumber\@empty
1614   \else \let\@secnumber\thechapter \fi
1615   \typeout{\chaptername\space\@secnumber}%
1616   \def\@toclevel{0}%
1617   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1618   \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1619   \chaptermark{##1}%
1620   \addtocontents{lof}{\protect\addvspace{10\p@}}%
1621   \addtocontents{lot}{\protect\addvspace{10\p@}}%
1622   \@makechapterhead{##2}\@afterheading
1623 }{%
1624   \refstepcounter{chapter}%
1625   \ifnum \c@secnumdepth<\z@ \let\@secnumber\@empty
1626   \else \let\@secnumber\thechapter \fi
1627   \typeout{\chaptername\space\@secnumber}%
1628   \def\@toclevel{0}%
1629   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1630   \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1631   \chaptermark{##1}%
1632   \ifdim \@chapterlistsgap>\z@
1633     \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1634     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1635     \caption@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1636   \fi
1637   \@makechapterhead{##2}\@afterheading}%

```

```

1638 % scrreprt/scrbook.cls
1639 \caption@ifundefined{KOMAClassName}{}{%
1640   \caption@Debug{document class '\KOMAClassName' detected}%
1641   \let\caption@patch@chapter\@gobblethree}%
1642 % rapport1/3.cls [2004/06/07 v2.1a NTG LaTeX document class]
1643 \caption@patch@chapter{rapport}{}{%
1644   \ifnum \c@secnumdepth >\m@ne
1645     \refstepcounter{chapter}%
1646     \typeout{\@chapapp\space\thechapter.}%
1647     \addcontentsline{toc}{chapter}%
1648       {\protect\numberline{\thechapter}\toc@font0 ##1}%
1649   \else
1650     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1651   \fi
1652   \chaptermark{##1}%
1653   \addtocontents{lof}{\protect\addvspace{10\p@}}%
1654   \addtocontents{lot}{\protect\addvspace{10\p@}}%
1655   \if@twocolumn
1656     \@topnewpage[\@makechapterhead{##2}]%
1657   \else
1658     \@makechapterhead{##2}%
1659     \@afterheading
1660   \fi
1661 }{%
1662   \ifnum \c@secnumdepth >\m@ne
1663     \refstepcounter{chapter}%
1664     \typeout{\@chapapp\space\thechapter.}%
1665     \addcontentsline{toc}{chapter}%
1666       {\protect\numberline{\thechapter}\toc@font0 ##1}%
1667   \else
1668     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1669   \fi
1670   \chaptermark{##1}%
1671   \ifdim \@chapterlistsgap>\z@
1672     \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1673     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1674     \caption@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1675   \fi
1676   \if@twocolumn
1677     \@topnewpage[\@makechapterhead{##2}]%
1678   \else
1679     \@makechapterhead{##2}%
1680     \@afterheading
1681   \fi}%
1682 % boek(3).cls [2004/06/07 v2.1a NTG LaTeX document class]
1683 \caption@patch@chapter{boek}{}{%
1684   \ifnum \c@secnumdepth >\m@ne
1685     \if@mainmatter
1686       \refstepcounter{chapter}%
1687       \typeout{\@chapapp\space\thechapter.}%
1688       \addcontentsline{toc}{chapter}%
1689         {\protect\numberline{\thechapter}\toc@font0 ##1}%
1690     \else

```

```

1691         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1692     \fi
1693 \else
1694     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1695 \fi
1696 \chaptermark{##1}%
1697 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1698 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1699 \if@twocolumn
1700     \topnewpage[\@makechapterhead{##2}]%
1701 \else
1702     \@makechapterhead{##2}%
1703     \@afterheading
1704 \fi
1705 }{%
1706     \ifnum \c@secnumdepth >\m@ne
1707         \if@mainmatter
1708             \refstepcounter{chapter}%
1709             \typeout{\@chapapp\space\thechapter.}%
1710             \addcontentsline{toc}{chapter}%
1711                 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1712         \else
1713             \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1714         \fi
1715     \else
1716         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1717     \fi
1718     \chaptermark{##1}%
1719     \ifdim \@chapterlistsgap>\z@
1720         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1721         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1722         \caption@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1723     \fi
1724     \if@twocolumn
1725         \topnewpage[\@makechapterhead{##2}]%
1726     \else
1727         \@makechapterhead{##2}%
1728         \@afterheading
1729     \fi}%
1730 % thesis.cls [1996/25/01 1.0g LaTeX document class (wm).]
1731 \caption@patch@chapter{thesis}{%
1732     \ifnum \c@secnumdepth >\m@ne
1733         \if@mainmatter
1734             \refstepcounter{chapter}%
1735             \typeout{\chaptername\space\thechapter.}
1736             \if@thema
1737                 \ifx\@shortauthor\@empty
1738                     \addcontentsline{toc}{chapter}{%
1739                         \protect\numberline{\thechapter.}##1}%
1740                 \else
1741                     \addcontentsline{toc}{chapter}{%
1742                         \protect\numberline{\thechapter.}%
1743                         \@shortauthor\hfill\mbox{}}\vskip\normallineskip ##1}%
1744                 \fi
1745             \fi
1746         \fi
1747     \fi

```



```

1745         \else
1746             \addcontentsline{toc}{chapter}{%
1747                 \protect\numberline{\thechapter.}##1}%
1748         \fi
1749     \else
1750         \addcontentsline{toc}{chapter}{##1}
1751     \fi
1752 \else
1753     \addcontentsline{toc}{chapter}{##1}
1754 \fi
1755 \chaptermark{##1}
1756 \addtocontents{lof}{\protect\addvspace{10pt}}
1757 \addtocontents{lot}{\protect\addvspace{10pt}}
1758 \if@twocolumn
1759     \@topnewpage[\@makechapterhead{##2}]
1760 \else
1761     \@makechapterhead{##2}
1762     \@afterheading
1763 \fi
1764 }{%
1765     \ifnum \c@secnumdepth >\m@ne
1766         \if@mainmatter
1767             \refstepcounter{chapter}%
1768             \typeout{\chaptername\space\thechapter.}%
1769             \if@thema
1770                 \ifx\@shortauthor\@empty
1771                     \addcontentsline{toc}{chapter}{%
1772                         \protect\numberline{\thechapter.}##1}%
1773                 \else
1774                     \addcontentsline{toc}{chapter}{%
1775                         \protect\numberline{\thechapter.}%
1776                         \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
1777                 \fi
1778             \else
1779                 \addcontentsline{toc}{chapter}{%
1780                     \protect\numberline{\thechapter.}##1}%
1781             \fi
1782         \else
1783             \addcontentsline{toc}{chapter}{##1}%
1784         \fi
1785     \else
1786         \addcontentsline{toc}{chapter}{##1}%
1787     \fi
1788     \chaptermark{##1}%
1789     \ifdim \@chapterlistsgap>\z@
1790         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1791         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1792         \caption@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1793     \fi
1794     \if@twocolumn
1795         \@topnewpage[\@makechapterhead{##2}]%
1796     \else
1797         \@makechapterhead{##2}%
1798         \@afterheading

```

```

1799     \fi}%
1800 \ifx\caption@patch@chapter\@gobblethree \else
1801     \caption@Debug{%
1802         Unsupported document class detected,\MessageBreak
1803         or \noexpand\@chapter was redefined by another package}%
1804     \fi
1805     \let\caption@PatchChapter\@undefined}
1806 %\@onlypreamble\caption@PatchChapter

1807 \newcommand\caption@patch@chapter[3]{%
1808     \begingroup
1809 %     \let\if@twocolumn\iffalse
1810     \let\if@mainmatter\iffalse
1811     \let\if@thema\iffalse
1812     \def\@tempa[##1]##2{#2}%
1813     \ifx\@tempa\@chapter
1814         \caption@Debug{document class ‘#1’ detected}%
1815         \gdef\@chapter[##1]##2{#3}%
1816         \global\let\caption@patch@chapter\@gobblethree
1817     \fi
1818     \endgroup}
1819 %\@onlypreamble\caption@patch@chapter

1820 \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.)

```

1821 \newcommand*\caption@patch@stpelt{%
1822     \let\caption@stpelt\@stpelt
1823     \def\@stpelt##1{%
1824         \caption@stpelt{##1}%
1825         \begingroup
1826             \let\@elt\caption@stpelt
1827             \csname caption@cl@##1\endcsname
1828         \endgroup}%
1829     \let\caption@patch@stpelt\relax}
1830 \@onlypreamble\caption@patch@stpelt

```

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

```

1831 \newcommand*\caption@addtoreset[2]{%
1832     \caption@patch@stpelt
1833     \@ifundefined{caption@cl@#2}{\@namedef{caption@cl@#2}}{}%
1834     \expandafter\@cons\csname caption@cl@#2\endcsname{#1}}
1835 \@onlypreamble\caption@addtoreset

```

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

```

1836 \newcommand*\caption@removefromreset[2]{%
1837     \begingroup
1838         \expandafter\let\csname c@#1\endcsname\caption@removefromreset
1839         \def\@elt##1{%
1840             \expandafter\ifx\csname c@##1\endcsname\caption@removefromreset
1841             \else
1842                 \noexpand\@elt{##1}%
1843             \fi}%

```

```

1844 \expandafter\xdef\csname caption@cl@#2\endcsname{%
1845 \csname caption@cl@#2\endcsname}%
1846 \endgroup}
1847 \@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>*.

```

1848 \newcommand*\DeclareCaptionSubType{%
1849 \caption@teststar\caption@declaresubtype\@firstoftwo\@secondoftwo}
1850 \@onlypreamble\DeclareCaptionSubType
1851 \newcommand*\caption@declaresubtype[1]{%
1852 \@testopt{\caption@@declaresubtype{#1}}{alph}}
1853 \@onlypreamble\caption@declaresubtype
1854 \def\caption@@declaresubtype#1[#2]#3{%
1855 \@ifundefined{c@#3}%
1856 {\caption@Error{No float type ‘#3’ defined}}%
1857 {\@ifundefined{c@sub#3}%
1858 {\caption@Debug{New subtype ‘sub#3’}%
1859 \newcounter{sub#3}%
1860 \caption@addtoreset{sub#3}{#3}%
1861 \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1862 \caption@declaresublistentry{#3}%
1863 \@cons\caption@subtypelist{{#3}}}%
1864 {\caption@Debug{Modify caption ‘sub#3’}}}%

```

Support of titletoc package

```

1865 \caption@ifundefined\contentsuse{}{%
1866 \contentsuse{sub#3}{\csname ext@sub#3\endcsname}}%
1867 \@namedef{sub#3name}{}%
1868 \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
1869 #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
1870 {\@namedef{p@sub#3}}{%
1871 \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1872 {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1873 \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1874 \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}%
1875 }}
1876 \@onlypreamble\caption@@declaresubtype
1877 \newcommand*\caption@declaresublistentry{%
1878 \caption@ifundefined\l@chapter
1879 {\caption@@declaresublistentry\l@subsubsection}%
1880 {\caption@@declaresublistentry\l@section}}
1881 \@onlypreamble\caption@declaresublistentry
1882 \newcommand*\caption@@declaresublistentry[2]{%
1883 \ifx#1\@undefined
1884 \caption@@@declaresublistentry\relax\@dottedtocline\caption@nil{#2}%
1885 \else
1886 \expandafter\caption@@@declaresublistentry#1{}{}\@dottedtocline\caption@nil{#2}%
1887 \fi}
1888 \@onlypreamble\caption@@declaresublistentry

```

```

1889 \long\def\caption@@@declaresublistentry#1\@dottedtocline#2\caption@nil#3{%
1890   \def\@tempa{#1}%
1891 % Does \l@ (sub)section start with \@dottedtocline?
1892   \ifx\@tempa\@empty
1893 % Yes
1894     \caption@@@declaresublistentry{#3}#2\caption@nil
1895   \else
1896 % No
1897     \caption@@@declaresublistentry{#3}@{3.8em}{3.2em}\caption@nil
1898   \fi}
1899 \@onlypreamble\caption@@@declaresublistentry
1900 \def\caption@@@declaresublistentry#1#2#3#4#5\caption@nil{%
1901   \expandafter\caption@@@declaresublistentry\expandafter
1902     {\csname @dotted\csname ext@#1\endcsname line\endcsname}{#1}{#3}{#4}}
1903 \@onlypreamble\caption@@@declaresublistentry
1904 \newcommand*\caption@@@declaresublistentry[4]{%
1905   \@namedef{l@sub#2}{#1{2}{#3}{#4}}%
1906   \caption@@@declaresublistentry#1{c@\csname ext@#2\endcsname depth}}
1907 \@onlypreamble\caption@@@declaresublistentry
1908 \newcommand*\caption@@@declaresublistentry[2]{
1909   \ifx#1\relax
1910     \def#1##1{%
1911       \def\next{\@dottedtocline{##1}}%
1912       \@ifundefined{#2}{}{%
1913         \ifnum ##1>\@nameuse{#2}\relax
1914         \let\next\@gobblefour
1915       }%
1916     }%
1917   \fi}
1918 \@onlypreamble\caption@@@declaresublistentry

```

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

```

1919 \newcommand*\caption@subtypelist{}

```

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

```

1920 \newcommand*\caption@For{\caption@withoptargs\caption@For}
1921 %\@onlypreamble\caption@For
1922 \newcommand\caption@@For[3]{%
1923   \caption@AtBeginDocument#1{%
1924     \def\@elt##1{#3}%
1925     \@nameuse{caption@#2}%
1926     \let\@elt\relax}}%
1927 %\@onlypreamble\caption@@For

```

24 subfig package adoptions

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

```

1928 \caption@AtBeginDocument{%
1929   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1930   \ifx\subref\@tempa
1931     \caption@InfoNoLine{subfig package 1.2 or 1.3 is loaded}%

```

```

1932 \let\caption@setfloattype\@gobble
1933 \let\@dottedxxxline\sf@NEW@dottedxxxline
1934 \let\sf@subfloat\sf@NEW@subfloat

1935 \fi
1936 \let\sf@NEW@dottedxxxline\@undefined
1937 \let\sf@NEW@subfloat\@undefined}

1938 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1939 \begingroup
1940 \caption@setfloattype{#1}%
1941 \caption@setoptions{subfloat}%
1942 \caption@setoptions{sub#1}%
1943 \ifnum #3>\@nameuse{c@#2depth}\else
1944 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1945 \fi
1946 \endgroup}

1947 \def\sf@NEW@subfloat{%
1948 \begingroup
1949 \caption@setfloattype\@capttype
1950 \sf@ifpositiontop{%
1951 \maincaptiontoptrue
1952 }{%
1953 \maincaptiontopfalse
1954 }%
1955 \caption@setoptions{subfloat}%
1956 \caption@setoptions{sub\@capttype}%
1957 \let\sf@oldlabel=\label
1958 \let\label=\subfloat@label
1959 \ifmaincaptiontop\else
1960 \advance\@nameuse{c@\@capttype}\@ne
1961 \fi
1962 \refstepcounter{sub\@capttype}%
1963 \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1964 \ifnextchar [% %] match left bracket
1965 {\sf@@subfloat}%
1966 {\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^AT_EX,
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