

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 v1.4b, last revised 2012/01/12.

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	28
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	35
19.9 The hungarian and magyar Babel option	35
19.10 Unknown document class (or package)	36
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	44

1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2012/01/12 v1.4b 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@ifin@list{#2}{#1}}

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

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

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

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

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

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

```

```

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

```

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

```

```

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

\caption@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\[1\]](#), 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 L^AT_EX 2_ε.

```
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 L^AT_EX 2_ε), 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}}
```

`\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.

```

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

265 \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”).

```

266 \newcommand*\caption@addtooptlist[1]{%
267   \@ifundefined{caption@opt@#1@lineno}{%
268     \caption@doptlist\caption@g@addto@list{#1}%
```

```

269 \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
270 }{}

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

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

279 \AtEndDocument{%
280 \caption@for@list\caption@optlist{%
281 \caption@WarningNoLine{%
282 Unused \string\captionsetup[#1]
283 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.

```

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

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

287 \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
288 \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
289 \expandafter\let\csname ORI@XKV@err\caption@keydepth\endcsname\XKV@err
290 \edef\caption@keydepth{\caption@keydepth i}%

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

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

300 \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
301 \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
302 \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
303 \expandafter\let\expandafter\XKV@err\csname ORI@XKV@err\caption@keydepth\endcsname
304 \ifx\caption@keydepth\@empty \captionsetup{starfalse} \fi
305 \@esphack}

306 \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*.

```

307 \newcommand*\caption@ExecuteOptions[2]{%
308 \expandafter\@expandtwoargs\csname caption@setkeys@#1\endcsname{#1}{#2}}%
309 \@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[3] v6.74 and improved.)

```

310 \newcommand*\caption@ProcessOptions{%
311   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
312 \@onlypreamble\caption@ProcessOptions

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

```

334 \newcommand*\caption@packagelist{}
335 \@onlypreamble\caption@packagelist

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

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

```

```

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

```

5 Margin resp. width

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

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

```

359 \newdimen\captionmargin
360 \newdimen\captionmargin@
361 \newdimen\captionwidth

362 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
363 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
364 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
365 \DeclareCaptionOption{width*}{\setcaptionwidth*{#1}}

366 \DeclareCaptionOption{calcmargin}{\caption@setcalcmargin{#1}}
367 \DeclareCaptionOption{calcmargin*}{\caption@setcalcmargin*{#1}}
368 \DeclareCaptionOption{calcwidth}{\caption@setcalcwidth{#1}}
369 \DeclareCaptionOption{calcwidth*}{\caption@setcalcwidth*{#1}}

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

372 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
373 \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.

```

374 \newcommand*\setcaptionmargin{%
375     \caption@resetcalcmargin
376     \caption@setmargin}

377 \newcommand*\caption@setmargin{%
378     \caption@teststar\caption@@setmargin\@gobble\@firstofone}

379 \newcommand*\caption@@setmargin[2]{%
380     #1{\captionwidth\z@}%
381     \caption@@@setmargin#2,#2,\@nil}

382 \def\caption@@@setmargin#1,#2,#3\@nil{%
383     \setlength\captionmargin@{#2}%
384     \setlength\captionmargin{#1}%
385     \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.

```
386 \newcommand*\setcaptionwidth{%
387   \caption@resetcalcmargin
388   \caption@setwidth}

389 \newcommand*\caption@setwidth{%
390   \caption@teststar\caption@@setwidth\@gobble\@firstofone}

391 \newcommand*\caption@@setwidth[2]{%
392   #1{\captionmargin\z@\captionmargin@\z@}%
393   \setlength\captionwidth{#2}}%
```

`\caption@resetcalcmargin`

```
394 \newcommand*\caption@resetcalcmargin{%
395   \let\caption@calcmargin@hook\@empty}
```

`\caption@setcalcmargin`

```
396 \newcommand*\caption@setcalcmargin{%
397   \caption@teststar{\caption@@setcalcmargin\caption@setmargin}%
398   \@secondoftwo\@firstoftwo}

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

`\caption@setcalwidth`

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

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

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

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

`\caption@newlabel` This command is a modified version of `\newlabel` from L^AT_EX2_ε. 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`.)

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

`\caption@thepage` This command is a modified version of `\thepage` from L^AT_EX2_ε. It will be used inside `\caption@ifoddpage` only.

```
412 \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`.

```

413 \newcommand*\caption@label[1]{%
414   \caption@@label
415   \protected@write\@auxout{\let\caption@thepage\relax}%
416     {\string\caption@newlabel{#1}{\caption@thepage}}}
417 \newcommand*\caption@@label{%
418   \global\let\caption@@label\relax
419   \protected@write\@auxout{}%
420     {\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`.

```

421 \newcommand*\caption@pageref[1]{%
422   \expandafter\ifx\csname caption@r@#1\endcsname\relax
423     \G@refundefinedtrue % => 'There are undefined references.'
424     \@latex@warning{Reference '#1' on page \thepage \space undefined}%
425   \else
426     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
427   \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[4] and `\ifodd\getpagerefnumber{...}`.
Note: This macro re-defines itself so the `.aux` file will only be used once per group.

```

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

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

```

439 \newcommand*\caption@setoptcmd[2]{%
440   \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 441 \newdimen\caption@indent
\caption@hangindent 442 \newdimen\caption@parindent
443 \newdimen\caption@hangindent

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



```

448 \DeclareCaptionOption{parindent}{%
449     \setlength\caption@parindent{#1}}
450 \DeclareCaptionOption{hangindent}{%
451     \setlength\caption@hangindent{#1}}
452 \DeclareCaptionOption{parskip}{%
453     \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

```

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

```

7 Styles

```

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

```
484 \newcommand*\caption@setstyle{%
485   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

486 \newcommand*\caption@@setstyle[2]{%
487   \ifundefined{caption@sty@#2}%
488     {#1{\caption@Error{Undefined style `#2'}}}%
489     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
490     \ifx\caption@setstyle@flag\@undefined
491       \let\caption@setstyle@flag\relax
492       \caption@resetstyle
493       \caption@xsetup\caption@sty
494       \let\caption@setstyle@flag\@undefined
495     }else
496       \caption@xsetup\caption@sty
497     \fi
498     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
499     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
500     \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!*

```
501 \newcommand*\caption@resetstyle{%
502   \caption@setup{%
503     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
504     justification=justified,font=,size=,labelfont=,textfont=,%
505     margin=0pt,minmargin=0,maxmargin=0,%
506     indent=0pt,parindent=0pt,hangindent=0pt,%
507     slc,rule,strut}%
508   \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.

```
509 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
510 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
511   format=default,labelsep=default,textformat=default,%
512   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.

```

513 \newcommand*\DeclareCaptionFormat{%
514   \caption@teststar\caption@declareformat\@gobble\@firstofone}
515 \@onlypreamble\DeclareCaptionFormat

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

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

524 \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`).

```

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

```

`\DeclareCaptionDefaultFormat`

```

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

```

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

```

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

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

```

‘default’ usually maps to ‘plain’.

```

549 \DeclareCaptionDefaultFormat{plain}

```

9 Label formats

```

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

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

```

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

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

```

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

```

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

```

558 \DeclareCaptionLabelFormat{empty}{}
559 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
560 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace}{#2}}
561 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace}#2)}

```

‘default’ usually maps to ‘simple’.

```
562 \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`.

```

563 \newcommand\DeclareCaptionLabelSeparator{%
564   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
565 \@onlypreamble\DeclareCaptionLabelSeparator

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

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

576 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
577 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}

```

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

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

```

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

```

```

581     {\edef\caption@lsepname{#1}%
582     \expandafter\let\expandafter\caption@iflfl\csname caption@iflfl@#1\endcsname
583     \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’.

```

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

```

`\caption@setdefaultlabelsep`

```

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

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

```

‘default’ usually maps to ‘colon’.

```

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

```

11 Text formats

```

\DeclareCaptionTextFormat \DeclareCaptionTextFormat{<name>}{<code with #1>}
602 \newcommand*\DeclareCaptionTextFormat[2]{%
603   \global\long\@namedef{caption@tfmt@#1}##1{#2}}
604 \onlypreamble\DeclareCaptionTextFormat

605 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
606 \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`).

```

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

```

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

```

611 \DeclareCaptionTextFormat{empty}{}
612 \DeclareCaptionTextFormat{simple}{#1}
613 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

12 Fonts

```

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

DeclareCaptionDefaultFont \DeclareCaptionDefaultFont{<name>}{<code>}
618 \newcommand*\DeclareCaptionDefaultFont[2]{%
619   \global\@namedef{caption#1@default}{#2}}
620 \@onlypreamble\DeclareCaptionDefaultFont

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

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

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

\caption@setfont \caption@setfont{<name>}{<keyval-list of names>}
Selecting a caption font means saving all the code snippets in \caption{<name>}.
630 \newcommand*\caption@setfont[1]{%
631   \expandafter\let\csname caption#1\endcsname\@empty
632   \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.
633 \newcommand*\caption@addtofont[2]{%
634   \begingroup
635     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
636     \define@key{caption@fnt}{default}[]{%
637       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
638     \caption@setkeys[caption]{caption@fnt}{#2}%
639     \global\let\caption@tempa\caption@fnt
640   \endgroup
641   \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.
642 \newcommand*\caption@font{%
643   \caption@teststar\caption@@font\@firstofone
644   {\caption@setkeys[caption]{caption@fnt}}
645 \newcommand*\caption@@font[2]{%
646   \begingroup
647   \def\caption@fnt{\endgroup}%
648   #1{#2}%
649   \caption@fnt}

```

These are the pre-defined font code snippets.

```

650 \DeclareCaptionFont{normalcolor}{\normalcolor}
651 \DeclareCaptionFont{color}{\color{#1}}

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

662 \DeclareCaptionFont{scriptsize}{\scriptsize}
663 \DeclareCaptionFont{footnotesize}{\footnotesize}
664 \DeclareCaptionFont{small}{\small}
665 \DeclareCaptionFont{normalsize}{\normalsize}
666 \DeclareCaptionFont{large}{\large}
667 \DeclareCaptionFont{Large}{\Large}

668 \DeclareCaptionFont{sansmath}{\sansmath}

669 \DeclareCaptionFont{singlespacing}{%
670   \caption@ifundefined{setspace@singlespace}{%
671     \setstretch\setspace@singlespace}}% normally 1
672 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
673 \DeclareCaptionFont{doublespacing}{\doublespacing}
674 \DeclareCaptionFont{stretch}{\setstretch{#1}}

675 %\DeclareCaptionFont{normal}{%
676 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
677 \DeclareCaptionFont{normal}{%
678   \caption@font*{%
679     \KV@caption@fnt@normalcolor\@unused
680     \KV@caption@fnt@normalfont\@unused
681     \KV@caption@fnt@normalsize\@unused
682     \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.

```

683 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
684 \DeclareCaptionDefaultFont{size}{}

```

13 Justifications

```

\DeclareCaptionJustification{\name}{\code}

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

\DeclareCaptionDefaultJustification{\code}

689 \newcommand*\DeclareCaptionDefaultJustification[1]{%

```

```

690 \global\@namedef{caption@hj@default}{#1}% for compatibility to v1.0
691 \DeclareCaptionDefaultFont{@hj}{#1}
692 \@onlypreamble\DeclareCaptionDefaultJustification

693 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
694 \DeclareCaptionDefaultJustification{}

```

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

Selecting a caption justification simply means saving the code (in `\caption@hj`).

```

695 \newcommand*\caption@setjustification{\caption@setfont{@hj}}

```

These are the pre-defined justification code snippets.

```

696 \DeclareCaptionJustification{justified}{}
697 \DeclareCaptionJustification{centering}{\centering}
698 \DeclareCaptionJustification{centerfirst}{\centerfirst}
699 \DeclareCaptionJustification{centerlast}{\centerlast}
700 \DeclareCaptionJustification{raggedleft}{\raggedleft}
701 \DeclareCaptionJustification{raggedright}{\raggedright}

```

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

```

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

```

`\centerlast` This is based on code from Anne Brüggemann-Klein[2]

```

713 \providecommand\centerlast{%
714   \let\\\@centercr
715   \leftskip\z@\@plus 1fil%
716   \rightskip\z@\@plus -1fil%
717   \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.

```

718 \DeclareCaptionJustification{Centering}{%
719   \caption@ragged\Centering\centering}
720 \DeclareCaptionJustification{RaggedLeft}{%
721   \caption@ragged\RaggedLeft\raggedleft}
722 \DeclareCaptionJustification{RaggedRight}{%
723   \caption@ragged\RaggedRight\raggedright}

```

`\caption@ragged` `\caption@ragged{<yes-code>}{<no-code>}` executes the `<yes-code>` if the `ragged2e` package is loaded and `<no-code>` if not. Additionally it tries to load the `ragged2e` package.

```

724 \newcommand*\caption@ragged{\caption@ifpackageloaded{ragged2e}}

```



```

\caption@ifpackageloaded \caption@ifpackageloaded{<package>}{<yes-code>}{<no-code>} executes the
<yes-code> if the given package is loaded and <no-code> if not. Additionally it tries to
load the package.

725 \newcommand*\caption@ifpackageloaded[1]{%
726   \ifundefined{caption@ifpkg@#1}%
727     {\caption@RequirePackage{#1}%
728       \caption@pkg@true{#1}}%
729     {}%
730   \caption@ifpkg{#1}}

731 \AtBeginDocument{\renewcommand*\caption@ifpackageloaded[1]{%
732   \ifundefined{caption@ifpkg@#1}%
733     {\caption@addto@pkg@list{#1}%
734       \caption@pkg@false{#1}%
735       \caption@Warning{%
736         `#1' support has been changed.\MessageBreak
737         Rerun to get captions right}}%
738     {}%
739   \caption@ifpkg{#1}}}

740 \newcommand*\caption@ifpkg[1]{%
741   \csname caption@ifpkg@#1\endcsname}
742 \newcommand*\caption@pkg@true[1]{%
743   \global\expandafter\let\csname caption@ifpkg@#1\endcsname\@firstoftwo}
744 \newcommand*\caption@pkg@false[1]{%
745   \global\expandafter\let\csname caption@ifpkg@#1\endcsname\@secondoftwo}

746 \newcommand*\caption@pkg@list{}
747 \newcommand*\caption@addto@pkg@list[1]{%
748   \protected@write\@auxout{}{%
749     \string\@cons\string\caption@pkg@list{{#1}}}}

750 \AtBeginDocument{%
751   \def\caption@tempa{\endgroup}%
752   \begingroup
753   \def\@elt#1{%
754     \g@addto@macro\caption@tempa{%
755       \caption@RequirePackage{#1}%
756       \@namedef{caption@ifpkg@#1}{%
757         \caption@addto@pkg@list{#1}%
758         \caption@pkg@true{#1}%
759         \caption@ifpkg{#1}}}%
760     \caption@pkg@list
761     \caption@tempa}

762 \newcommand*\caption@RequirePackage[1]{%
763   \caption@Info{We need package `#1'}%
764   \RequirePackage{#1}}
765 \@onlypreamble\caption@RequirePackage

```

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`

```

766 \caption@ifundefined\abovecaptionskip{%

```

```

767 \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}{}
768 \caption@ifundefined\belowcaptionskip{%
769 \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}{}

770 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
771 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
772 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

\caption@rule \caption@rule
Draws an invisible rule to adjust the “skip” setting.
773 \newcommand*\caption@rule{\caption@ifrule\caption@hrule\relax}
774 \newcommand*\caption@hrule{\hrule\@height\z@}

775 \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.

```

776 \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 pack-
age or document, but use the wrapper macro \caption@iftop instead.
777 \newcommand*\caption@setposition[1]{%
778 \caption@ifinlist{#1}{d,default}{%
779 \let\caption@position\caption@defaultpos
780 }{\caption@ifinlist{#1}{t,top,above}{%
781 \let\caption@position\@firstoftwo
782 }{\caption@ifinlist{#1}{b,bottom,below}{%
783 \let\caption@position\@secondoftwo
784 }{\caption@ifinlist{#1}{a,auto}{%
785 \let\caption@position\@undefined
786 }{%
787 \caption@Error{Undefined position `#1'}%
788 }}}}}

```

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

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

```

789 %\caption@setdefaultpos{a}% default = auto
790 \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.)

```
791 \newcommand*\caption@iftop{%
792   \ifx\caption@position\undefined
793     \let\caption@position\@secondoftwo
794 %   = \caption@setposition b%
795   \fi
796   \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.

```
797 \newcommand*\caption@fixposition{%
798   \ifx\caption@position\undefined
799     \caption@autoposition
800   \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^[5] 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.

```
801 \newcommand*\caption@autoposition{%
802   \ifvmode
803     \edef\caption@tempa{\the\prevdepth}%
804     \caption@Debug{\protect\prevdepth=\caption@tempa}%
805     \ifdim\prevdepth>-\p@
806       \let\caption@position\@secondoftwo
807     \else
808       \let\caption@position\@firstoftwo
809     \fi
810 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
811   \else
812     \caption@Debug{no \protect\prevdepth}%
813     \let\caption@position\@secondoftwo
814 %   = \caption@setposition b%
815   \fi}
```

`\caption@setautoposition` `\caption@setautoposition{<position>}`
 replaces the above algorithm by a different one (or a fixed position setting).

```
816 \newcommand*\caption@setautoposition[1]{%
817   \def\caption@autoposition{\caption@setposition{#1}}}
```

16 Hooks

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

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

```

818 \newcommand*\caption@beginhook{}
819 \newcommand*\caption@endhook{}
820 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
821 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

17 Lists

```

822 \DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
823 \DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}

\caption@setlist \caption@setlist{\langle boolean \rangle}

824 \newcommand*\caption@setlist{\caption@set@bool\caption@iflist}

825 \DeclareCaptionOption{listtype}{\caption@setlisttype{#1}}
826 \DeclareCaptionOption{listtype+}{\caption@setlisttype@ext{#1}}

\caption@setlisttype \caption@setlisttype{\langle type \rangle}

827 \newcommand*\caption@setlisttype{%
828   \caption@setlisttype@ext{}%
829   \caption@@setlisttype\caption@listtype}

830 \newcommand*\caption@@setlisttype[2]{%
831   \edef#1{#2}%
832   \ifx#1\@empty \let#1\@undefined \fi}

\caption@setlisttype@ext \caption@setlisttype@ext{\langle type extension \rangle}

833 \newcommand*\caption@setlisttype@ext{%
834   \caption@@setlisttype\caption@listtype@ext}

\DeclareCaptionListFormat \DeclareCaptionListFormat{\langle name \rangle}{\langle code with #1 and #2 \rangle}

835 \newcommand*\DeclareCaptionListFormat[2]{%
836   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
837 \@onlypreamble\DeclareCaptionListFormat

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

\caption@setlistformat \caption@setlistformat{\langle name \rangle}

```

Selecting a caption list format simply means saving the code (in \caption@lstfmt).

```

839 \newcommand*\caption@setlistformat[1]{%
840   \@ifundefined{caption@lstfmt@#1}%
841     {\caption@Error{Undefined list format `#1'}}%
842     {\expandafter\let\expandafter\caption@lstfmt
843       \csname caption@lstfmt@#1\endcsname}}

```

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

```

844 \DeclareCaptionListFormat{empty}{}
845 \DeclareCaptionListFormat{simple}{#1#2}
846 \DeclareCaptionListFormat{parens}{#1 (#2)}
847 \DeclareCaptionListFormat{subsimple}{#2}
848 \DeclareCaptionListFormat{subparens}{(#2)}

```

tion@setdefaultlistformat

```

849 \newcommand*\caption@setdefaultlistformat[1]{%
850   \ifx\caption@lstfmt\caption@lstfmt@default
851     \caption@set@default@listformat{#1}%
852     \caption@setlistformat{default}%
853   \else
854     \caption@set@default@listformat{#1}%
855   \fi}

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

‘default’ usually maps to ‘subsimple’.

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

18 Debug option

```

859 \DeclareCaptionOption{debug}[1]{%
860   \caption@set@bool\caption@ifdebug{#1}%
861   \caption@ifdebug
862     {\let\caption@Debug\caption@Info}%
863     {\let\caption@Debug\@gobble}}

864 \DeclareOption{debug}{\setkeys{caption}{debug}}

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

19 Document classes & Babel support

19.1 The standard L^AT_EX classes

```

866 \caption@CheckCommand\@makecaption{%
867   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
868   \long\def\@makecaption#1#2{%
869     \vskip\abovecaptionskip
870     \sbox\@tempboxa{#1: #2}%
871     \ifdim \wd\@tempboxa >\hsize
872       #1: #2\par
873     \else
874       \global \@minipagefalse
875       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
876     \fi
877     \vskip\belowcaptionskip}}
```

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

\caption@ifamsclass

```

878 \providecommand*\caption@ifamsclass{%
879   \caption@ifundefined\@captionheadfont\@gobble\@firstofone}
880 \@onlypreamble\caption@ifamsclass

881 \caption@ifamsclass{%
882   \caption@CheckCommand\@makecaption{%
883     % amsart|amsproc|amsbook [2004/08/06 v2.20]
884     \long\def\@makecaption#1#2{%
```

```

885 \setbox\@tempboxa\vbox{\color@setgroup
886 \advance\hsize-2\captionindent\noindent
887 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
888 {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
889 \unskip\kern-2\captionindent\par
890 \global\setbox\@ne\lastbox\color@endgroup}%
891 \ifhbox\@ne % the normal case
892 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
893 \fi
894 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
895 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
896 \else % tempboxa contained more than one line
897 \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
898 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
899 \fi
900 \ifnum\@tempcnta<64 % if the float IS a figure...
901 \addvspace\abovecaptionskip
902 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
903 \else % if the float IS NOT a figure...
904 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
905 \nobreak
906 \vskip\belowcaptionskip
907 \fi
908 \relax
909 }}
910 \caption@CheckCommand\makecaption{%
911 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
912 \long\def\makecaption#1#2{%
913 \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
914 \setbox\@tempboxa\vbox{\color@setgroup
915 \advance\hsize-2\captionindent\noindent
916 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
917 {\@cdr#2\@nil}{.\@addpunct{.}\@captionfont\upshape\enspace#2}%
918 \unskip\kern-2\captionindent\par
919 \global\setbox\@ne\lastbox\color@endgroup}%
920 \ifhbox\@ne % the normal case
921 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
922 \fi
923 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
924 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
925 \@tempdima\wd\@ne\advance\@tempdima-\captionindent
926 \wd\@ne\@tempdima
927 \else % tempboxa contained more than one line
928 \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
929 \unvbox\@tempboxa\parskip\z@skip
930 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
931 \fi
932 \ifnum\@tempcnta<64 % if the float IS a figure...
933 \addvspace\abovecaptionskip
934 \noindent\kern\captionindent\box\@ne
935 \else % if the float IS NOT a figure...
936 \noindent\kern\captionindent\box\@ne
937 \nobreak
938 \vskip\belowcaptionskip

```

```

939     \fi
940     \relax
941   }}

942   \let\captionmargin\captionindent % set to 3pc by AMS class
943   \begingroup\edef\@tempa{\endgroup
944     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
945       {margin=\the\captionmargin
946         \caption@ifundefined\smf@makecaption{{},maxmargin=.1\linewidth}}}}
947   \@tempa
948   \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
949   \DeclareCaptionLabelSeparator{default}{{.\enspace}
950   \DeclareCaptionDefaultFont{font}{{\@captionfont}
951   \DeclareCaptionDefaultFont{labelfont}{{\@captionheadfont}
952   \DeclareCaptionDefaultFont{textfont}{{\@captionfont\upshape}
953   \captionsetup[figure]{position=b}
954   \captionsetup[table]{position=t}

955 }

```

19.3 The beamer class (Part one)

\caption@ifbeamerclass

```

956 \providecommand*\caption@ifbeamerclass{%
957   \@ifclassloaded{beamer}\@firstofone\@gobble}
958 \@onlypreamble\caption@ifbeamerclass

959 \caption@ifbeamerclass{%
960   \caption@CheckCommand\beamer@makecaption{%
961     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
962     \long\def\beamer@makecaption#1#2{%
963       \def\insertcaptionname{\csname#1name\endcsname}%
964       \def\insertcaptionnumber{\csname the#1\endcsname}%
965       \def\insertcaption{#2}%
966       \nobreak\vskip\abovecaptionskip\nobreak
967       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
968       \ifdim \wd\@tempboxa >\hsize
969         \usebeamertemplate**{caption}\par
970       \else
971         \global \@minipagefalse
972         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
973       \fi
974       \nobreak\vskip\belowcaptionskip\nobreak}}

```

\caption@ifbeamertemplate

```

975 \newcommand*\caption@ifbeamertemplate[1]{%
976   \begingroup
977     \let\beamer@@tmpl\caption@ORI\beamer@@tmpl\caption
978     \@nameuse{beamer@@tmpop\caption@#1}%
979     \ifx\beamer@@tmpl\caption@ORI\beamer@@tmpl\caption
980       \endgroup\expandafter\@firstoftwo
981     \else
982       \endgroup\expandafter\@secondoftwo
983     \fi}

```

```

984 \DeclareCaptionLabelFormat{default}{%
985   #1\caption@ifbeamertemplate{numbered}{~#2}{}}
986 \caption@declarelabeleseparator
987   {\caption@ifbeamertemplate{caption name own line}\@gobble\@firstofone}
988   {default}
989   {\caption@ifbeamertemplate{caption name own line}{\}\{:\ }}
990 \DeclareCaptionDefaultFont{font}{%
991   \usebeamerfont*{caption}%
992   \usebeamercolor[fg]{caption}}
993 \DeclareCaptionDefaultFont{labelfont}{%
994   \usebeamercolor[fg]{caption name}%
995   \usebeamerfont*{caption name}}
996 \DeclareCaptionDefaultJustification{\raggedright}
997 \DeclareOption{beamerclass}{%
998   \renewcommand\caption@ifslc{%
999     \caption@ifbeamertemplate{caption name own line}\@secondoftwo\@firstoftwo}
1000   % Since the beamer class do not offer a 'list of figures' we switch this supp
1001   \captionsetup{list=0}}
1002 \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.)

```

1003 \defbeamertemplate{caption}{caption3}{%
1004   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
1005 \DeclareOption{beamer}{%
1006   % \usebeamertemplate*{caption} will set font
1007   \DeclareCaptionDefaultFont{font}{}%
1008   \setbeamertemplate{caption}[caption3]}
1009 %
1010 %   \begin{macrocode]
1011 }

```

19.4 The KOMA-Script classes

`\caption@ifkomaclass`

```

1012 \providecommand*\caption@ifkomaclass{%
1013   \caption@ifundefined\scr@caption\@gobble\@firstofone}
1014 \@onlypreamble\caption@ifkomaclass
1015 \caption@ifkomaclass{%
1016   \caption@CheckCommand\@makecaption{%
1017     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
1018     \long\def\@makecaption#1#2{%
1019       \if@captionabove
1020         \vskip\belowcaptionskip
1021       \else
1022         \vskip\abovecaptionskip
1023       \fi
1024       \@@makecaption\@firstofone{#1}{#2}%
1025       \if@captionabove
1026         \vskip\abovecaptionskip
1027       \else

```



```

1028         \vskip\belowcaptionskip
1029     \fi}}

1030 \DeclareCaptionFormat{default}[#1#2#3\par]{%
1031     \ifdofullc@p
1032         \caption@ifin@list\caption@lsepclist\caption@lsepname
1033         {\caption@Error{%
1034             The option 'labelsep=\caption@lsepname' does not work\MessageBreak
1035             with \noexpand\setcaphanging (which is set by default)}}%
1036         {\caption@fmt@hang{#1}{#2}{#3}}%
1037     \else
1038         #1#2%
1039         \ifdim\cap@indent<\z@
1040             \par
1041             \noindent\hspace*{ -\cap@indent}%
1042         \else\if@capbreak
1043             \par
1044             \fi\fi
1045         #3\par
1046     \fi}
1047 \DeclareCaptionLabelSeparator{default}{\captionformat}
1048 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
1049 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
1050 }

```

19.5 The NTG Dutch classes

\caption@ifntgclass

```

1051 \providecommand*\caption@ifntgclass{%
1052     \caption@ifundefined\CaptionFonts\@gobble\@firstofone}
1053 \onlypreamble\caption@ifntgclass

1054 \caption@ifntgclass{%
1055     \caption@CheckCommand\@makecaption{%
1056         % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
1057         \long\def\@makecaption#1#2{%
1058             \vskip\abovecaptionskip
1059             \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
1060             \ifdim \wd\@tempboxa >\hsize
1061                 {\CaptionLabelFont#1:} \CaptionTextFont#2\par
1062             \else
1063                 \global \@minipagefalse
1064                 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1065             \fi
1066             \vskip\belowcaptionskip}}
1067 \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
1068 \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
1069 }

```

19.6 The thesis class

\caption@ifthesisclass

```

1070 \providecommand*\caption@ifthesisclass{%
1071   \caption@ifundefined\cph@font
1072   {\@gobble}%
1073   {\caption@ifundefined\cpb@font\@gobble\@firstofone}}

1074 \caption@ifthesisclass{%

1075   \caption@CheckCommand\@makecaption{%
1076     % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
1077     \long\def\@makecaption#1#2{%
1078       \vskip\abovecaptionskip
1079       \setbox\@tempboxa\hbox{{\cph@font #1:} {\cpb@font #2}}%
1080       \ifdim \wd\@tempboxa >\hsize
1081         \@hangfrom{\cph@font #1:} {\cpb@font #2\par}%
1082       \else
1083         \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
1084       \fi
1085       \vskip\belowcaptionskip}}

1086   \DeclareCaptionDefaultFormat{hang}
1087   \DeclareCaptionDefaultFont{labelfont}{\cph@font}
1088   \DeclareCaptionDefaultFont{textfont}{\cpb@font}

1089 }

```

19.7 The frenchb Babel option

```

1090 \caption@ifundefined\FB@makecaption{}{%

1091   \caption@CheckCommand\@makecaption{%
1092     % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
1093     % frenchb.ldf [2007/10/05 v2.0e French support from the babel system]
1094     \long\def\@makecaption#1#2{%
1095       \vskip\abovecaptionskip
1096       \sbox\@tempboxa{#1\CaptionSeparator #2}%
1097       \ifdim \wd\@tempboxa >\hsize
1098         #1\CaptionSeparator #2\par
1099       \else
1100         \global \@minipagefalse
1101         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1102       \fi
1103       \vskip\belowcaptionskip}}

1104   \ifx\@makecaption\STD@makecaption
1105     \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
1106     \def\caption@frenchb{% supress frenchb warning
1107       \let\STD@makecaption\@makecaption
1108       \let\FB@makecaption\@makecaption}
1109   \else
1110     \ifx\@makecaption\@undefined\else
1111       \caption@InfoNoLine{%
1112         The definition of \protect\@makecaption\space
1113         has been changed,\MessageBreak
1114         frenchb will NOT customize it}%
1115     \fi
1116   \fi

1117 }

```

19.8 The frenchle/pro package

```

1118 \caption@ifundefined\frenchTeXmods{}{%
1119   \caption@CheckCommand\@makecaption{%
1120     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
1121     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
1122     \def\@makecaption#1#2{%
1123       \ifFTY%
1124         \def\@secondofmany##1##2\void{##2}%
1125         \def\@tempa{\@secondofmany#2\void}%
1126         \ifx\@tempa\empty%
1127           \let\captionseparator\empty%
1128         \fi%
1129         \@mcORI{#1}{\relax\captionfont{#2}}%
1130       \else
1131         \@mcORI{#1}{#2}%
1132       \fi}%
1133   \caption@CheckCommand\@makecaption{%
1134     % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
1135     \def\@makecaption#1#2{%
1136       \ifFTY%
1137         \def\@secondofmany##1##2\void{##2}%
1138         \protected@edef\@tempa{\@secondofmany#2\void}%
1139         \ifx\@tempa\empty%
1140           \let\captionseparator\empty%
1141         \fi%
1142         \@mcORI{#1}{\relax\captionfont{#2}}%
1143       \else
1144         \@mcORI{#1}{#2}%
1145       \fi}%
1146   \DeclareCaptionDefaultFont{textfont}{\itshape}%
1147   \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
1148 }

```

19.9 The hungarian and magyar Babel option

```

1149 \DeclareCaptionListFormat{subperiod}{#2.}
1150 \caption@ifundefined\hunnewlabel{}{%
1151   \caption@CheckCommand\@makecaption{%
1152     % magyar.ldf [2005/03/30 v1.4j Magyar support from the babel system]
1153     \def\@makecaption#1#2{%
1154       \vskip\abovecaptionskip
1155       \sbox\@tempboxa{#1. #2}%
1156       \ifdim \wd\@tempboxa >\hsize
1157         {#1. #2\csize par\endcsize}
1158       \else
1159         \global \@minipagefalse
1160         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1161       \fi
1162       \vskip\belowcaptionskip}}
1163   \def\caption@tempa#1{\@ifundefined{extras#1}}{%
1164     \expandafter\addto\csize extras#1\endcsize}%
1165     % change default labelsep and listformat

```

```

1166 \caption@setdefaultlabelsep{period}%
1167 \caption@setdefaultlistformat{subperiod}}}%
1168 \expandafter\addto\csname noextras#1\endcsname{%
1169 % change default labelsep and listformat
1170 \caption@setdefaultlabelsep{colon}%
1171 \caption@setdefaultlistformat{subsimple}}}%
1172 }}
1173 \caption@tempa{hungarian}
1174 \caption@tempa{magyar}

```

19.10 Unknown document class (or package)

```

1175 \caption@ifcheckcommand{%
1176 \caption@setbool{documentclass}{1}%
1177 }{%
1178 \caption@setbool{documentclass}{0}%
1179 \caption@info{no line}%
1180 Unknown document class (or package), \MessageBreak
1181 standard defaults will be used}%
1182 \caption@debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1183 }

```

20 Execution of options

```

1184 \captionsetup{style=default,position=default,%
1185 list,listformat=default,twoside=\if@twoside 1\else 0\fi}
1186 \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`.

```

1187 \newcommand\caption@addcontentsline[2]{%
1188 \caption@ifcontentsline{#2}{%
1189 \begingroup
1190 \let\@tempa\@gobble
1191 \caption@ifundefined\caption@listtype
1192 {\edef\caption@listtype{#1}}%
1193 {\let\@tempa\@firstofone}%
1194 \caption@ifundefined\caption@listtype@ext
1195 }}%
1196 {\edef\caption@listtype{\caption@listtype\caption@listtype@ext}%
1197 \let\@tempa\@firstofone}%
1198 \@tempa
1199 {\caption@debug{addcontentsline: #1 => \caption@listtype}%
1200 % \caption@setoptions\caption@listtype
1201 \@namedef{the\caption@listtype}{\@nameuse{the#1}}}%
1202 \expandafter\caption@@addcontentsline\expandafter{\caption@listtype}{#2}%
1203 \endgroup}}
1204 \newcommand\caption@@addcontentsline[2]{%
1205 {\let\\\space
1206 \@ifundefined{ext@#1}%

```

```

1207     {\caption@Error{No float type ‘#1’ defined}}%
1208     {\caption@@@addcontentsline
1209      {\csname ext@#1\endcsname}%
1210      {#1}%
1211      {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1212      {\ignorespaces #2}}}%
1213 \newcommand*\caption@@@addcontentsline[4]{%
1214   \addcontentsline{#1}{#2}{\protect\numberline{#3}{#4}}}
1215 \newcommand\caption@ifcontentsline[1]{%
1216   \caption@iflist
1217     {\def\@tempa{#1}}%
1218     {\let\@tempa\@empty}%
1219   \ifx\@tempa\@empty
1220     \expandafter\@gobble
1221   \else
1222     \expandafter\@firstofone
1223   \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.)
1224 \newif\ifcaption@star

\caption@fnum \caption@fnum{<float type>}
Typesets the caption label; as replacement for \fnum{<float type>}.
1225 \newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}

\caption@make \caption@make{<float name>}{<ref. number>}{<text>}
Typesets the caption.
1226 \newcommand\caption@make[2]{\caption@@make{\caption@lfmt{#1}{#2}}}

\caption@@make \caption@@make{<caption label>}{<caption text>}
1227 \newcommand\caption@@make[2]{%
1228   \begingroup
1229   \caption@stepcounter
1230   \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
1231 % TODO: Move this to \caption@calcmargin!?
1232   \ifx\caption@maxmargin\undefined \else
1233     \ifdim\captionmargin>\caption@maxmargin\relax
1234     \captionmargin\caption@maxmargin\relax
1235   \fi
1236 \fi
1237   \ifx\caption@minmargin\undefined \else
1238     \ifdim\captionmargin<\caption@minmargin\relax
1239     \captionmargin\caption@minmargin\relax
1240   \fi
1241 \fi

Special single-line treatment (option singlelinecheck=)
1242   \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}}}%

```

Typeset the left margin (option margin=)

```

1243 \caption@calcmargin
1244 \@tempdima\captionmargin
1245 \ifdim\captionmargin@=\z@ \else
1246 \caption@ifoddpage{}{\advance\@tempdima\captionmargin@}%
1247 \fi
1248 \caption@ifh{\advance\@tempdima\caption@indent}%
1249 \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*.

```

1250 \@tempdima\captionwidth
1251 \caption@ifh{\advance\@tempdima-\caption@indent}%
1252 \caption@parbox\@tempdima{%

```

Typeset the indentation (option indentation=)

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

```

1253 \caption@ifh{%
1254 \ifdim\caption@indent=\z@
1255 \leavevmode
1256 \else
1257 \hskip-\caption@indent
1258 \fi}%

```

Typeset the caption itself and close the \caption@parbox

```

1259 \caption@@@make{#1}{#2}}%

```

Typeset the right margin (option margin=)

```

1260 \@tempdima\captionmargin
1261 \ifdim\captionmargin@=\z@ \else
1262 \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
1263 \fi
1264 \hspace\@tempdima

1265 \caption@endhook
1266 \endgroup

1267 \global\caption@starfalse}

```

\caption@calcmargin \caption@calcmargin

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

```

1268 \newcommand*\caption@calcmargin{%
1269 \caption@calcmargin@hook
1270 \ifdim\captionwidth=\z@
1271 \captionwidth\linewidth
1272 \advance\captionwidth by -2\captionmargin
1273 \advance\captionwidth by -\captionmargin@
1274 \else
1275 \captionmargin\linewidth
1276 \advance\captionmargin by -\captionwidth
1277 \divide\captionmargin by 2
1278 \captionmargin@\z@
1279 \fi

```

```

1280 \caption@Debug{%
1281   \string\hsize=\the\hsize,
1282   \string\linewidth=\the\linewidth,\MessageBreak
1283   \string\leftmargin=\the\leftmargin,
1284   \string\rightmargin=\the\rightmargin,\MessageBreak
1285   \string\margin=\the\captionmargin,
1286   \string\margin@=\the\captionmargin@,
1287   \string\width=\the\captionwidth}%
1288 }

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.
1289 \newcommand\caption@slc[4]{%
1290   \caption@@slc{#1}{#2}{#3}{\caption@singleline#4}{}}
1291 \newcommand\caption@@slc[5]{%
1292   \caption@Debug{Begin SLC}%
1293   \begingroup
1294   \caption@singleline
1295   \let\caption@hj\@empty
1296   \caption@calcmargin % calculate #3 if necessary
1297   \caption@prepareslc
1298   \sbox\@tempboxa{\caption@@make{#1}{#2}}%
1299   \ifdim\wd\@tempboxa>#3%
1300     \endgroup
1301     #5%
1302   \else
1303     \endgroup
1304     #4%
1305   \fi
1306   \caption@Debug{End SLC}}
1307 \newcommand*\caption@singleline{%
1308   \caption@xsetup\caption@opt@singleline
1309   \let\caption@fmt\caption@slfmt}

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1310 \newcommand*\caption@prepareslc{%
1311   \let\label\caption@gobble
1312   \let\caption@footnotemark@ORI\footnotemark
1313   \def\footnote{\caption@withoptargs\caption@footnote}%
1314   \def\footnotemark{\caption@withoptargs\caption@footnotemark}%
1315   \let\@footnotetext\caption@gobble
1316   \let\@endnotetext\caption@gobble
1317   \let\pagenote\caption@gobble
1318 }
1319 \newcommand\caption@footnote[2]{%
1320   \caption@footnotemark{#1}}
1321 \newcommand\caption@footnotemark[1]{%
1322   \begingroup
1323     \let\stepcounter\caption@l@stepcounter
1324     \caption@footnotemark@ORI#1%
1325   \endgroup}

```

```

1326 \newcommand*\caption@l@stepcounter[1]{%
1327   \advance\cscname c@#1\endcsname\@ne\relax}

\caption@parbox \caption@parbox{<width>}{<contents>}
This macro defines the box which surrounds the caption paragraph.
1328 \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=.
1329 \newcommand*\caption@applyfont{%
1330   \caption@hj\captionfont\captionsize}

\caption@@@make \caption@@@make{<caption label>}{<caption text>}
This one finally typesets the caption paragraph, without margin and indentation.
1331 \newcommand\caption@@@make[2]{%
If the label is empty, we use no caption label separator.
1332   \sbox\@tempboxa{#1}%
1333   \ifdim\wd\@tempboxa=\z@
1334     \let\caption@lsep\relax
1335 %   \@capbreakfalse
1336   \fi

If the text is empty, we use no caption label separator, too. (And no text format either.)
1337   \caption@ifempty{#2}{%
1338     \let\caption@lsep\@empty
1339     \let\caption@tfmt\@firstofone
1340 %   \@capbreakfalse
1341 %   \let\caption@ifstrut\@secondoftwo
1342 }%

Take care that \caption@parindent and \caption@hangindent will be used
to typeset the paragraph.
1343   \@setpar{\@par\caption@@par}\caption@@par

Finally typeset the caption.
1344   \caption@applyfont
1345   \caption@fmt
1346   {\ifcaption@star\else{\captionlabelfont#1}\fi}%
1347   {\ifcaption@star\else{\caption@iflfl\captionlabelfont\caption@lsep}\fi}%
1348   {{\captiontextfont
1349     \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1350     \nobreak\hskip\z@skip % enable hyphenation
1351     \caption@tfmt{#2}%
1352 %   \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1353     \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1354   \par}}

\caption@ifempty \caption@ifempty{<text>}{<true>}(no <false>))
This one tests if the <text> is actually empty.
Note: This will be done without expanding the text, therefore this is far away from being bullet-
proof.
Note: This macro is re-defining itself so only the first test (in a group) will actually be done.

```



```

1355 \newcommand\caption@ifempty[1]{%
1356   \caption@if@empty{#1}%
1357   \caption@ifempty\@unused}

1358 \newcommand\caption@if@empty[1]{%
1359   \def\caption@tempa{#1}%
1360   \ifx\caption@tempa\@empty
1361     \let\caption@ifempty\@secondoftwo
1362   \else
1363     \expandafter\def\expandafter\caption@tempa\expandafter{%
1364       \caption@car#1\caption@if@empty\caption@nil}%
1365     \def\caption@tempb{\caption@if@empty}%
1366     \ifx\caption@tempa\caption@tempb
1367       \let\caption@ifempty\@secondoftwo
1368     \else
1369       \def\caption@tempb{\ignorespaces}%
1370       \ifx\caption@tempa\caption@tempb
1371         \expandafter\caption@if@empty\expandafter{\@gobble#1}%
1372       \else
1373         \def\caption@tempb{\label}%
1374         \ifx\caption@tempa\caption@tempb
1375           \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1376         \else
1377           \def\caption@tempb{\index}%
1378           \ifx\caption@tempa\caption@tempb
1379             \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1380           \else
1381             \def\caption@tempb{\glossary}%
1382             \ifx\caption@tempa\caption@tempb
1383               \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1384             \else
1385               \let\caption@ifempty\@gobbletwo
1386             \fi
1387           \fi
1388         \fi
1389       \fi
1390     \fi
1391   \fi}

1392 \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.

```

1393 \newcommand*\caption@@par{%
1394   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

23 Types & sub-types

```

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

1395 \newcommand*\DeclareCaptionType{%
1396   \RequirePackage{newfloat}%
1397   \DeclareFloatingEnvironment}
1398 \@onlypreamble\DeclareCaptionType

```

`\caption@ForEachType` `\caption@ForEachType{<code>}` will execute the given code for all (known) floating environments.

```

1399 \newcommand\caption@ForEachType[1]{%
1400   \caption@ifundefined\ForEachFloatingEnvironment
1401     {\def\@elt##1{##1}%
1402      \caption@ifundefined\c@figure\@gobble\@elt{figure}%
1403      \caption@ifundefined\c@table\@gobble\@elt{table}%
1404      \let\@elt\relax
1405      \newfloat@addtohook{##1}}%
1406   {\ForEachFloatingEnvironment{##1}}}

1407 \providecommand\newfloat@addtohook[1]{%
1408   \toks@=\expandafter{\newfloat@hook{##1}##1}%
1409   \edef\@tempa{\def\noexpand\newfloat@hook###1{\the\toks@}}%
1410   \@tempa}

1411 \providecommand*\newfloat@hook[1]{}

```

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

```

1412 \newcommand*\caption@patch@stpelt{%
1413   \let\caption@stpelt\@stpelt
1414   \def\@stpelt##1{%
1415     \caption@stpelt{##1}%
1416     \begingroup
1417       \let\@elt\caption@stpelt
1418       \csname caption@cl@##1\endcsname
1419     \endgroup}%
1420   \let\caption@patch@stpelt\relax}
1421 \@onlypreamble\caption@patch@stpelt

```

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

```

1422 \newcommand*\caption@addtoreset[2]{%
1423   \caption@patch@stpelt
1424   \@ifundefined{caption@cl@#2}{\@namedef{caption@cl@#2}{}{}}{%
1425     \expandafter\@cons\csname caption@cl@#2\endcsname{{#1}}}%
1426   \@onlypreamble\caption@addtoreset

```

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

```

1427 \newcommand*\caption@removefromreset[2]{%
1428   \begingroup
1429     \expandafter\let\csname c@#1\endcsname\caption@removefromreset
1430     \def\@elt##1{%
1431       \expandafter\ifx\csname c@##1\endcsname\caption@removefromreset
1432       \else
1433         \noexpand\@elt{##1}%
1434       \fi}%
1435     \expandafter\xdef\csname caption@cl@#2\endcsname{%
1436       \csname caption@cl@#2\endcsname}%
1437   \endgroup}
1438 \@onlypreamble\caption@removefromreset

```

`\DeclareCaptionSubType` `\DeclareCaptionSubType[<numbering scheme>]{<type>}`
`\DeclareCaptionSubType*[<numbering scheme>]{<type>}`

The starred variant provides the numbering format $\langle type \rangle . \langle subtype \rangle$ while the non-starred variant simply uses $\langle subtype \rangle$.

```

1439 \newcommand*\DeclareCaptionSubType{%
1440   \caption@teststar\caption@declaresubtype\@firstoftwo\@secondoftwo}
1441 \@onlypreamble\DeclareCaptionSubType

1442 \newcommand*\caption@declaresubtype[1]{%
1443   \@testopt{\caption@@declaresubtype{#1}}{\alph}}
1444 \@onlypreamble\caption@declaresubtype

1445 \def\caption@@declaresubtype#1[#2]#3{%
1446   \@ifundefined{c@#3}%
1447     {\caption@Error{No float type ‘#3’ defined}}%
1448     {\@ifundefined{c@sub#3}%
1449       {\caption@Debug{New subtype ‘sub#3’}%
1450         \newcounter{sub#3}%
1451         \caption@addtoreset{sub#3}{#3}%
1452         \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1453         \caption@declaresublistentry{#3}%
1454         \@cons\caption@subtypelist{{#3}}}%
1455       {\caption@Debug{Modify caption ‘sub#3’}}}%

```

Support of titletoc package

```

1456   \caption@ifundefined\contentsuse{}{%
1457     \contentsuse{sub#3}{\csname ext@sub#3\endcsname}}%

1458   \@namedef{sub#3name}{}%
1459   \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
1460   #1 is \@firstoftwo in star form, and \@secondoftwo otherwise
1461   {\@namedef{p@sub#3}}{%
1462     \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1463   {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1464     \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1465   \@namedef{theHsub#3}{\csname theH#3\endcsname.\@arabic{sub#3}}%
1466   }}

1467 \@onlypreamble\caption@@declaresubtype

1468 \newcommand*\caption@declaresublistentry{%
1469   \caption@ifundefined\l@chapter
1470     {\caption@@declaresublistentry\l@subsubsection}%
1471     {\caption@@declaresublistentry\l@section}}
1472 \@onlypreamble\caption@declaresublistentry

1473 \newcommand*\caption@@declaresublistentry[2]{%
1474   \ifx#1\@undefined
1475     \caption@@@declaresublistentry\relax\@dottedtocline\caption@nil{#2}%
1476   \else
1477     \expandafter\caption@@@declaresublistentry#1{}{}\@dottedtocline\caption@nil{#
1478   \fi}
1479 \@onlypreamble\caption@@declaresublistentry

1480 \long\def\caption@@@declaresublistentry#1\@dottedtocline#2\caption@nil#3{%
1481   \def\@tempa{#1}%
1482   % Does \l@ (sub)subsection start with \@dottedtocline?
1483   \ifx\@tempa\@empty
1484     % Yes
1485     \caption@@@declaresublistentry{#3}#2\caption@nil

```

```

1486 \else
1487 % No
1488 \caption@@@declaresublistentry{#3}@{3.8em}{3.2em}\caption@nil
1489 \fi}
1490 \@onlypreamble\caption@@@declaresublistentry
1491 \def\caption@@@declaresublistentry#1#2#3#4#5\caption@nil{%
1492 \expandafter\caption@@@declaresublistentry\expandafter
1493 {\csname @dotted\csname ext@#1\endcsname line\endcsname}{#1}{#3}{#4}}
1494 \@onlypreamble\caption@@@declaresublistentry
1495 \newcommand*\caption@@@declaresublistentry[4]{%
1496 \@namedef{l@sub#2}{#1{2}{#3}{#4}}%
1497 \caption@@@declaresublistentry#1{c@\csname ext@#2\endcsname depth}}
1498 \@onlypreamble\caption@@@declaresublistentry
1499 \newcommand*\caption@@@@declaresublistentry[2]{
1500 \ifx#1\relax
1501 \def#1##1{%
1502 \def\next{\@dottedtocline{##1}}%
1503 \@ifundefined{#2}{}{%
1504 \ifnum ##1>\@nameuse{#2}\relax
1505 \let\next\@gobblefour
1506 \fi}%
1507 \next}%
1508 \fi}
1509 \@onlypreamble\caption@@@@declaresublistentry

```

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

```

1510 \newcommand*\caption@subtypelist{}

```

```

\caption@For \caption@For{<elt-list>}{<code with #1>}
\caption@For*{<elt-list>}{<code with #1>}
1511 \newcommand*\caption@For{\caption@withoptargs\caption@For}
1512 %\@onlypreamble\caption@For
1513 \newcommand\caption@@For[3]{%
1514 \caption@AtBeginDocument#1{%
1515 \def\@elt##1{#3}%
1516 \@nameuse{caption@#2}%
1517 \let\@elt\relax}}%
1518 %\@onlypreamble\caption@@For

```

24 subfig package adoptions

Since the subfig package is not maintained anymore, we have to make several adoptions to the caption kernel *v1.1* here. Please note that we only support the version 1.3 of the subfig package here. So older versions do not work with this version of the caption kernel, and never versions are expected to be adapted.

```

1519 \caption@AtBeginDocument{%
1520 \def\@tempa{2005/06/28 ver: 1.3 subfig package}%
1521 \expandafter\ifx\csname ver@subfig.sty\endcsname\@tempa
1522 \caption@InfoNoLine{subfig package v1.3 is loaded}%

```

```

1523 \let\caption@setfloattype\@gobble
1524 \let\@dottedxxxline\sf@NEW@dottedxxxline
1525 \let\sf@subfloat\sf@NEW@subfloat

1526 \fi
1527 \let\sf@NEW@dottedxxxline\@undefined
1528 \let\sf@NEW@subfloat\@undefined}

\@dottedxxxline New version of \@dottedxxxline
1529 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1530 \begingroup
1531 \caption@setfloattype{#1}%
1532 \caption@setoptions{subfloat}%
1533 \caption@setoptions{sub#1}%
1534 \ifnum #3>\@nameuse{c@#2depth}\else
1535 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1536 \fi
1537 \endgroup}

\sf@subfloat New version of \sf@subfloat
1538 \def\sf@NEW@subfloat{%
1539 \begingroup
1540 \@nameuse{caption@warmup}%
1541 \caption@setfloattype\@capttype
1542 \sf@ifpositiontop{%
1543 \maincaptiontoptrue
1544 }{%
1545 \maincaptiontopfalse
1546 }%
1547 \caption@setoptions{subfloat}%
1548 \caption@setoptions{sub\@capttype}%
1549 \let\sf@oldlabel=\label
1550 \let\label=\subfloat@label
1551 \ifmaincaptiontop\else
1552 \advance\@nameuse{c@\@capttype}\@ne
1553 \fi
1554 \refstepcounter{sub\@capttype}%
1555 \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1556 \@ifnextchar [% %] match left bracket
1557 {\sf@@subfloat}%
1558 {\sf@@subfloat[\@empty]}}

```

References

- [1] Frank Mittelbach and Michel Goossens:
The L^AT_EX Companion (2nd. Ed.),
Addison-Wesley, 2004.
- [2] Anne Brüggemann-Klein:
Einführung in die Dokumentverarbeitung,
B.G. Teubner, Stuttgart, 1989
- [3] Sebastian Rahtz & Heiko Oberdiek:
Hypertext marks in L^AT_EX,
November 12, 2007
- [4] Heiko Oberdiek:
The refcount package,
2006/02/20
- [5] Donald Arseneau:
Three part tables: title, tabular environment, notes,
2003/06/13