

The Implementation of the caption package^{*}

Axel Sommerfeldt
`caption@sommerfee.de`

2008/03/20

Abstract

The caption package 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 L^AT_EX 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 L^AT_EX document – it just provides functionality which can be used by L^AT_EX 2_ε packages which typesets captions, for example the caption and subfig packages.

The caption package itself redefines the L^AT_EX commands `\caption`, `\@caption`, and `\@makecaption` and maps the latter one to `\caption@@make`, giving the user the possibility to control the look & feel of the captions from floating environments like `figure` and `table`. Furthermore it does similar to the caption stuff coming from other packages (like the `longtable` or `supertabular` package): Mapping the appropriate internal commands (like `\LT@makecaption` or `\ST@caption`) to the ones offered by the `caption3` kernel. So you can think of the caption package as a layer package, it simply provides adaptation layers between the caption stuff coming from L^AT_EX 2_ε or packages, and the caption stuff offered by the `caption3` kernel.

User manuals

This document is describing the code implementation only. The user documentation can be found in

<code>caption-eng.pdf</code>	The English documentation
<code>caption-rus.pdf</code>	The Russian documentation ¹
<code>caption-deu.pdf</code>	The German documentation

^{*}This package has version number v3.1g, last revised 2008/03/20.

¹Thanks a lot to Olga Lapko for this translation

Contents

1	Kernel	4
1.1	Identification	4
1.2	Generic helpers	4
1.3	Errors	7
1.4	Using the keyval package	8
1.5	Margin resp. width	12
1.6	Indentions	14
1.7	Styles	15
1.8	Formats	16
1.9	Label formats	17
1.10	Label separators	18
1.11	Text formats	19
1.12	Fonts	19
1.13	Justifications	21
1.13.1	The ragged2e package	22
1.14	Vertical spaces before and after captions	23
1.15	Positioning	23
1.16	Hooks	25
1.17	Lists	25
1.18	Debug option	25
1.19	Document classes & Babel support	26
1.19.1	The standard L ^A T _E X classes	26
1.19.2	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	26
1.19.3	The beamer class	27
1.19.4	The KOMA-Script classes	28
1.19.5	The NTG Dutch classes	29
1.19.6	The thesis class	29
1.19.7	The frenchb Babel option	29
1.19.8	The frenchle/pro package	30
1.20	Execution of options	31
1.21	Making an ‘List of’ entry	31
1.22	Typesetting the caption	31
1.23	Types & sub-types	35
1.24	subfig package adaption	39
2	Main package	40
2.1	Identification	40
2.2	Loading the kernel	40
2.3	Check against incompatible packages	40

2.4	Check document class	40
2.5	Adaption to the \mathcal{AMS} & SMF document classes	41
2.6	Emulation of the KOMA-Script commands	41
2.7	Declaration of options	43
2.7.1	Options for figure and table	43
2.7.2	Miscellaneous options	44
2.7.3	caption v1.x compatibility options	44
2.7.4	caption2 v2.x compatibility options	45
2.7.5	Obsolete caption v3.0 options	45
2.7.6	ftpage package support options	45
2.7.7	hyperref package support options	45
2.8	Processing of options	46
2.9	<code>\captionof</code> and <code>\captionlistentry</code>	46
2.10	<code>\ContinuedFloat</code>	48
2.11	Internal helpers	50
2.12	<code>\caption</code> , <code>\@caption</code> , and <code>\@makecaption</code>	52
2.13	Support for sub-captions	58
2.14	Document class & Babel package support	60
2.14.1	The \mathcal{AMS} & SMF classes	60
2.14.2	The beamer class	60
2.14.3	The KOMA-Script classes	61
2.14.4	The frenchb Babel option	61
2.14.5	The frenchle/pro package	61
2.15	Package support	62
2.15.1	The float package	64
2.15.2	The floatflt package	67
2.15.3	The ftpage package	68
2.15.4	The hyperref package	70
2.15.5	The hypcap package	73
2.15.6	The listings package	74
2.15.7	The longtable package	75
2.15.8	The picinpar package	77
2.15.9	The picins package	78
2.15.10	The rotating package	80
2.15.11	The sidecap package	81
2.15.12	The subfigure package	83
2.15.13	The supertabular package	83
2.15.14	The threeparttable package	84
2.15.15	The wrapfig package	85

1 Kernel

1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2008/03/20 v3.1g caption3 kernel (AR)]
```

1.2 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.

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

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

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

would expand to yes.

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

```

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

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

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

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

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

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

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

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

```

```

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

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

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

\caption@teststar \caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
\caption@teststar@{<cmd>}{<star arg>}{<non-star arg>}
74 \newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}
75 \newcommand*\caption@teststar@[3]{%
76   \@ifstar{#1{#2}}{\caption@ifatletter{#1{#2}}{#1{#3}}}}
77 \AtBeginDocument{\let\caption@teststar@\caption@teststar}

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

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

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

```

```

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

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

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

```

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

```

120 \let\caption@begindocumenthook\@empty
121 \let\caption@@begindocumenthook\@empty

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

126 \AtBeginDocument{%
127   \PackageInfo{caption}{Begin \noexpand\AtBeginDocument code\@gobble}%

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

132   \def\caption@AtBeginDocument{%
133     \@ifstar\@firstofone\@firstofone}%
134   \caption@@begindocumenthook
135   \let\caption@@begindocumenthook\@undefined

136   \PackageInfo{caption}{End \noexpand\AtBeginDocument code\@gobble}}

```

1.3 Errors

```

\caption@Warning \caption@Error{<message>}
\caption@WarningNoLine 137 \newcommand*\caption@Warning[1]{%
  \caption@Error 138   \caption@WarningNoLine{#1\on@line}}
\caption@KV@err

```

```

139 \newcommand*\caption@WarningNoLine[1]{%
140   \PackageWarning{caption}{#1.^J\caption@wh\@gobbletwo}}
141 \newcommand*\caption@Error[1]{%
142   \PackageError{caption}{#1}\caption@eh}
143 \let\caption@KV@err\caption@Error

```

\caption@wh At the moment we only offer these two simple warning resp. error helpers.

```

\caption@eh 144 \newcommand*\caption@wh{%
145   See the caption package documentation for explanation.}
146 \newcommand*\caption@eh{%
147   If you do not understand this error, please take a closer look\MessageBreak
148   at the documentation of the 'caption' package, especially the\MessageBreak
149   section about errors.\MessageBreak\@ehc}

```

1.4 Using the keyval package

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

```
150 \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.

```

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

```

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

Analogous to \@onlypreamble from L^AT_EX 2_ε.

```

153 \providecommand*\@preamble@keys{}
154 \providecommand*\@onlypreamble@key[2]{\@cons\@preamble@keys{{#1}{#2}}}
155 \@onlypreamble\@onlypreamble@key
156 \@onlypreamble\@preamble@keys
157 \providecommand*\@notprerr@key[1]{\KV@err{Can be used only in preamble}}
158 \caption@AtBeginDocument{%
159   \def\@elt#1#2{\expandafter\let\csname KV@#1@#2\endcsname\@notprerr@key}%
160   \@preamble@keys
161   \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.

```

162 \newcommand*\DeclareCaptionOption{%
163   \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
164 \@onlypreamble\DeclareCaptionOption
165 \newcommand*\caption@declareoption[2]{%
166   #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
167 \@onlypreamble\caption@declareoption

```



```

\DeclareCaptionOptionNoValue \DeclareCaptionOptionNoValue{<option>}{<code>}
\DeclareCaptionOptionNoValue*{<option>}{<code>}
Same as \DeclareCaptionOption but issues an error if a value is given.

168 \newcommand*\DeclareCaptionOptionNoValue{%
169   \caption@teststar\caption@declareoption@novalue\AtEndOfPackage\@gobble}
170 \@onlypreamble\DeclareCaptionOptionNoValue

171 \newcommand\caption@declareoption@novalue[3]{%
172   \caption@declareoption{#1}{#2}[\KV@err]{%
173     \caption@option@novalue{#2}{##1}{#3}}
174 \@onlypreamble\caption@declareoption@novalue

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

183 \newcommand*\captionsetup{%
184   \caption@teststar\@captionsetup\@gobble\@firstofone}

185 \newcommand*\@captionsetup[1]{%
186   \captionsetup@startrue#1\captionsetup@starfalse
187   \@ifnextchar[\caption@setup@options\caption@setup}

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

189 \def\caption@setup@options[#1]#2{%
190   \@bsphack
191   \ifcaptionsetup@star\captionsetup@starfalse\else\caption@addtooptlist{#1}\fi
192   \expandafter\caption@l@addto@list\csname caption@opt@#1\endcsname{#2}%
193   \@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.

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

196 \newcommand*\@clearcaptionsetup[1]{%
197   \let\caption@tempa#1%
198   \@testopt\@clearcaptionsetup{}}

```

```

199 \def\@@clearcaptionsetup[#1]#2{%
200   \@bsphack
201   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
202   {\caption@tempa{\caption@Warning{Option list `#2' undefined}}}%
203   {\ifx,#1,%
204     \caption@clearsetup{#2}%
205   \else
206     \caption@@removefromsetup{#1}{#2}%
207   \fi}%
208   \@esphack}

209 \newcommand*\caption@clearsetup[1]{%
210   \caption@removefromoptlist{#1}%
211   \@nameundef{caption@opt@#1}}

212 \newcommand*\caption@removefromsetup{%
213   \let\caption@tempa\@gobble
214   \caption@@removefromsetup}

215 \newcommand*\caption@@removefromsetup[2]{%
216   \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
217   \expandafter\let\csname caption@opt@#2\endcsname\@undefined
218   \def\@tempb##1=##2\@nil{##1}%
219   \edef\@tempc{#1}%
220   \@for\@tempa:=\@tempa\do{%
221     \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
222     \ifx\@tempd\@tempc
223       \let\caption@tempa\@gobble
224     \else
225       \expandafter\expandafter\expandafter\caption@l@addto@list
226       \expandafter\csname caption@opt@#2\expandafter\endcsname
227       \expandafter{\@tempa}%
228     \fi}%
229   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
230   {\caption@removefromoptlist{#2}}}%
231   \caption@tempa{\caption@Warning{%
232     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>*.

```

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

244 \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.

```

245 \newcommand*\caption@setoptions[1]{%
246   \caption@Debug{options=#1}%
247   \expandafter\let\expandafter\caption@opt\csname caption@opt@#1\endcsname
248   \ifx\caption@opt\relax \else
249     \caption@xsetup\caption@opt
250     \caption@clearsetup{#1}%
251   \fi}

252 \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”).

```

253 \newcommand*\caption@addtooptlist[1]{%
254   \ifundefined{caption@opt@#1@lineno}{%
255     \caption@dooptlist\caption@g@addto@list{#1}%
256     \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
257   }{}}

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

261 \newcommand*\caption@dooptlist[2]{%
262   \begingroup
263     \edef\@tempa{#2}\@onelevel@sanitize\@tempa
264     \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
265   \endgroup}

266 \AtEndDocument{%
267   \caption@for@list\caption@optlist{%
268     \PackageWarningNoLine{caption}{%
269       Unused \string\captionsetup[#1]
270       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.

```

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

272 \long\def\caption@@setkeys[#1]#2#3{%
273   \@bsphack

274   \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
275   \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
276   \expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
277   \let\KV@errx\KV@err
278   \edef\caption@keydepth{\caption@keydepth i}%

279   \caption@Debug{\protect\setkeys{#2}{#3}}%
280   \setkeys{#2}{#3}%

```

```

281 \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
282 \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
283 \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
284 \ifx\caption@keydepth\@empty \captionsetup@starfalse \fi
285 \@esphack}
286 \let\caption@keydepth\@empty

\caption@ExecuteOptions \caption@ExecuteOptions{\family}{\key-values}
We execute our options using the keyval interface, so we use this one instead of
\ExecuteOptions offered by LATEX 2ε.
287 \newcommand*\caption@ExecuteOptions[2]{%
288 \@expandtwoargs\caption@setkeys{#1}{#2}}%
289 \@onlypreamble\caption@ExecuteOptions

\caption@ProcessOptions \caption@ProcessOptions*{\family}
We process our options using the keyval package, so we use this one instead of
\ProcessOptions offered by LATEX 2ε. The starred variant do not process the global
options. (This code was taken from the hyperref package[9] v6.74 and improved.)
290 \newcommand*\caption@ProcessOptions{%
291 \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
292 \@onlypreamble\caption@ProcessOptions

293 \newcommand*\caption@@ProcessOptions[2]{%
294 \let\@tempc\relax
295 \let\caption@tempa\@empty
296 #1{% \@firstofone -or- \@gobble
297 \@for\CurrentOption:=\@classoptionslist\do{%
298 \@ifundefined{KV@#2@}\CurrentOption}{}%
299 \@ifundefined{KV@#2@\CurrentOption @default}{%
300 \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
301 }{%
302 \PackageInfo{#2}{Global option '\CurrentOption' processed}%
303 \edef\caption@tempa{\caption@tempa,\CurrentOption,}%
304 \@expandtwoargs\@removeelement\CurrentOption
305 \@unusedoptionlist\@unusedoptionlist
306 }%
307 }%
308 }%
309 \let\CurrentOption\@empty
310 }%
311 \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
312 \AtEndOfPackage{\let\@unprocessedoptions\relax}}
313 \@onlypreamble\caption@@ProcessOptions

```

1.5 Margin resp. width

`\captionmargin` and `\captionwidth` contain the extra margin resp. the total width used for captions. Please never set these values in a direct way, they are just accessible in user documents to provide compatibility to *v1.x*. Note that we can only set one value at a time, ‘margin’ or ‘width’. If `\captionwidth` is not zero we will take this value afterwards, otherwise `\captionmargin` and `\captionmargin@`.

```

314 \newdimen\captionmargin
315 \newdimen\captionmargin@
316 \newdimen\captionwidth

317 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
318 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
319 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
320 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}
321 \DeclareCaptionOptionNoValue{oneside}{\caption@set@bool\caption@iftwoside0}

322 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
323 \DeclareCaptionOption{maxmargin}{\caption@setoptcmd\caption@maxmargin{#1}}

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

324 \newcommand*\setcaptionmargin{%
325   \caption@teststar\caption@setmargin\@gobble\@firstofone}

326 \newcommand*\caption@setmargin[2]{%
327   #1{\captionwidth\z@}%
328   \caption@@setmargin#2,#2,\@nil}

329 \def\caption@@setmargin#1,#2,#3\@nil{%
330   \setlength\captionmargin@{#2}%
331   \setlength\captionmargin{#1}%
332   \addtolength\captionmargin@{-\captionmargin}}

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

333 \newcommand*\setcaptionwidth{%
334   \captionmargin\z@
335   \captionmargin@\z@
336   \setlength\captionwidth}

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

337 \newcommand*\caption@thecounter{0}

338 \newcommand*\caption@stepcounter{%
339   \@tempcnta\caption@thecounter
340   \advance\@tempcnta\@ne
341   \xdef\caption@thecounter{\the\@tempcnta}}

\caption@newlabel This command is a modified version of \newlabel from LATEX2ε. 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.)

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

\caption@thepage This command is a modified version of \thepage from LATEX2ε. It will be used inside
\caption@ifoddpage only.

343 \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`.

```

344 \newcommand*\caption@label[1]{%
345   \caption@@label
346   \protected@write\@auxout{\let\caption@thepage\relax}%
347     {\string\caption@newlabel{#1}{\caption@thepage}}}
348 \newcommand*\caption@@label{%
349   \global\let\caption@@label\relax
350   \protected@write\@auxout{}%
351     {\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`.

```

352 \newcommand*\caption@pageref[1]{%
353   \expandafter\ifx\csname caption@r@#1\endcsname\relax
354     \G@refundefinedtrue % => 'There are undefined references.'
355     \caption@Warning{Reference on page \thepage \space undefined}%
356   \else
357     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
358   \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[23] and `\ifodd\getpagerefnumber{...}`.
Note: This macro re-defines itself so the .aux file will only be used once per group.

```

359 \newcommand*\caption@ifoddpage{%
360   \caption@iftwoside{%
361     \caption@label\caption@thecounter
362     \caption@pageref\caption@thecounter
363     \ifodd\caption@thepage
364       \let\caption@ifoddpage\@firstoftwo
365     \else
366       \let\caption@ifoddpage\@secondoftwo
367     \fi
368   }{\let\caption@ifoddpage\@firstoftwo}%
369   \caption@ifoddpage}
```

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

```

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

1.6 Indentions

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

`\caption@parindent` 372 \newdimen\caption@indent

`\caption@hangindent` 373 \newdimen\caption@parindent

374 \newdimen\caption@hangindent

```

375 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
376   \setlength\caption@indent{#1}}
377 \DeclareCaptionOption{indentation}[\leftmargini]{%
378   \setlength\caption@indent{#1}}
379 \DeclareCaptionOption{parindent}{%
```

```

380      \setlength\caption@parindent{#1}}
381 \DeclareCaptionOption{hangindent}{%
382      \setlength\caption@hangindent{#1}}
383 \DeclareCaptionOption{parskip}{%
384      \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).²

```

385 \@ifundefined{scr@caption}{}{%
386   \let\caption@KV@parindent\KV@caption@parindent
387   \DeclareCaptionOption{parindent}[]{%
388     \ifx,#1,%
389       \caption@Debug{Option 'parindent' ignored}%
390     \else
391       \caption@KV@parindent{#1}%
392     \fi}%
393   \let\caption@KV@parskip\KV@caption@parskip
394   \DeclareCaptionOption{parskip}[]{%
395     \ifx,#1,%
396       \caption@Debug{Option 'parskip' ignored}%
397     \else
398       \caption@KV@parskip{#1}%
399     \fi}%
400 }

```

1.7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
401 \newcommand*\DeclareCaptionStyle[1]{%
402   \@testopt{\caption@declarestyle{#1}}{}%
403   \@onlypreamble\DeclareCaptionStyle
404   \def\caption@declarestyle#1[#2]#3{%
405     \global\@namedef{caption@sls@#1}{#2}%
406     \global\@namedef{caption@sty@#1}{#3}%
407     \@onlypreamble\caption@declarestyle
408   }
409   \DeclareCaptionOption{style}{\caption@setstyle{#1}}
409   \DeclareCaptionOption{style*}{\caption@setstyle*{#1}}
410   \DeclareCaptionOption{singlelinecheck}{\caption@set@bool\caption@ifslc{#1}}

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

```

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

²This problem was completely solved due a change of `\caption@ProcessOptions` in the caption package v3.0j, but we still need this workaround since these options would otherwise still collide with the current version 1.3 of the subfig package (Sigh!)

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.

```

411 \newcommand*\caption@setstyle{%
412   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

413 \newcommand*\caption@@setstyle[2]{%
414   \@ifundefined{caption@sty@#2}%
415     {#1{\caption@Error{Undefined style `#2'}}}%
416     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
417     \ifx\caption@setstyle@flag\@undefined
418       \let\caption@setstyle@flag\relax
419       \caption@resetstyle
420       \caption@xsetup\caption@sty
421       \let\caption@setstyle@flag\@undefined
422     \else
423       \caption@xsetup\caption@sty
424     \fi
425     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
426     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
427     \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!*

```

428 \newcommand*\caption@resetstyle{%
429   \caption@setup{%
430     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
431     justification=justified,font=,size=,labelfont=,textfont=,%
432     margin=0pt,minmargin=0,maxmargin=0,%
433     indent=0pt,parindent=0pt,hangindent=0pt,%
434     singlelinecheck=1,strut=1}%
435   \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 package v3.0), the second one matches the document class actually used.

```

436 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
437 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{}
438   format=default,labelsep=default,textformat=default,%
439   justification=default,font=default,labelfont=default,textfont=default}
```

1.8 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.

```

440 \newcommand*\DeclareCaptionFormat{%
441   \caption@teststar\caption@declareformat\@gobble\@firstofone}
442 \@onlypreamble\DeclareCaptionFormat
```



```

443 \newcommand*\caption@declareformat[2]{%
444   \@dblarg{\caption@@declareformat#1{#2}}}%
445 \@onlypreamble\caption@declareformat

446 \long\def\caption@@declareformat#1#2[#3]#4{%
447   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
448   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
449   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}%
450 \@onlypreamble\caption@@declareformat

451 \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`).

```

452 \newcommand*\caption@setformat[1]{%
453   \@ifundefined{caption@fmt@#1}%
454     {\caption@Error{Undefined format `#1'}}%
455     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
456      \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
457      \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

`\DeclareCaptionDefaultFormat`

```

458 \newcommand*\DeclareCaptionDefaultFormat[1]{%
459   \expandafter\def\expandafter\caption@fmt@default\expandafter
460     {\csname caption@fmt@#1\endcsname}%
461   \expandafter\def\expandafter\caption@slfmt@default\expandafter
462     {\csname caption@slfmt@#1\endcsname}%
463   \expandafter\def\expandafter\caption@ifh@default\expandafter
464     {\csname caption@ifh@#1\endcsname}}
465 \@onlypreamble\DeclareCaptionDefaultFormat

```

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

```

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

467 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
468   \caption@ifin@list\caption@lsepclist\caption@lsepname
469   {\caption@Error{%
470     The option ‘labelsep=\caption@lsepname’ does not work\MessageBreak
471     with ‘format=hang’}}%
472   {\@hangfrom{#1#2}%
473    \advance\caption@parindent\hangindent
474    \advance\caption@hangindent\hangindent
475    \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

476 \DeclareCaptionDefaultFormat{plain}

```

1.9 Label formats

`\DeclareCaptionLabelFormat` `\DeclareCaptionLabelFormat{<name>}{<code with #1 and #2>}`

```

477 \newcommand*\DeclareCaptionLabelFormat[2]{%
478   \global\@namedef{caption@lfmt@#1}##1##2{#2}}%
479 \@onlypreamble\DeclareCaptionLabelFormat

```

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

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

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

```
481 \newcommand*\caption@setlabelformat[1]{%
```

```
482 \ifundefined{caption@lfmt@#1}%
```

```
483 {\caption@Error{Undefined label format `#1'}}%
```

```
484 {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}
```

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

```
485 \DeclareCaptionLabelFormat{empty}{}%
```

```
486 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
```

```
487 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace} (#2) }
```

```
488 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace} #2) }
```

‘default’ usually maps to ‘simple’.

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

1.10 Label separators

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

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

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

```
490 \newcommand\DeclareCaptionLabelSeparator{%
```

```
491 \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
```

```
492 \@onlypreamble\DeclareCaptionLabelSeparator
```

```
493 \newcommand\caption@declarelabelseparator[3]{%
```

```
494 \global\expandafter\let\csname caption@iflf@#2\endcsname#1%
```

```
495 \global\long\@namedef{caption@lsep@#2}{#3}%
```

```
496 \caption@@declarelabelseparator{#2}#3\\@nil}
```

```
497 \@onlypreamble\caption@declarelabelseparator
```

```
498 \long\def\caption@@declarelabelseparator#1#2\\#3\\@nil{%
```

```
499 \def\@tempa{#3}\ifx\@tempa\empty \else
```

```
500 \caption@g@addto@list\caption@lsep@#1%
```

```
501 \fi}
```

```
502 \@onlypreamble\caption@@declarelabelseparator
```

```
503 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
```

```
504 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

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

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

```
505 \newcommand*\caption@setlabelseparator[1]{%
```

```
506 \ifundefined{caption@lsep@#1}%
```

```
507 {\caption@Error{Undefined label separator `#1'}}%
```

```
508 {\edef\caption@lsepname{#1}%
```

```
509 \expandafter\let\expandafter\caption@iflf\csname caption@iflf@#1\endcsname
```

```
510 \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’.

```
511 \DeclareCaptionLabelSeparator{none}{}%
```

```

512 \DeclareCaptionLabelSeparator{colon}{: }
513 \DeclareCaptionLabelSeparator{period}{. }
514 \DeclareCaptionLabelSeparator{space}{ }
515 \DeclareCaptionLabelSeparator*{quad}{\quad}
516 \DeclareCaptionLabelSeparator*{newline}{\\}
517 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}
‘default’ usually maps to ‘colon’.
518 \def\caption@lsep@default{\caption@lsep@colon}
519 \def\caption@iflf@default{\caption@iflf@colon}

```

1.11 Text formats

```

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

523 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
524 \DeclareCaptionOption{strut}{\caption@set@bool\caption@ifstrut{#1}}

```

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

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

```

525 \newcommand*\caption@settextformat[1]{%
526   \ifundefined{caption@tfmt@#1}%
527     {\caption@Error{Undefined text format `#1'}}%
528     {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

529 \DeclareCaptionTextFormat{simple}{#1}
530 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

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

1.12 Fonts

```

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

DeclareCaptionDefaultFont \DeclareCaptionDefaultFont{<name>}{<code>}
535 \newcommand*\DeclareCaptionDefaultFont[2]{%
536   \global\@namedef{caption#1@default}{#2}}
537 \@onlypreamble\DeclareCaptionDefaultFont

538 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
539 \DeclareCaptionOption{font+}{\caption@addtofont{font}{#1}}
540 \DeclareCaptionDefaultFont{font}{}

541 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
542 \DeclareCaptionOption{labelfont+}{\caption@addtofont{labelfont}{#1}}
543 \DeclareCaptionDefaultFont{labelfont}{}

```

```

544 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
545 \DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
546 \DeclareCaptionDefaultFont{textfont}{}

```

`\caption@setfont` `\caption@setfont{<name>}{<keyval-list of names>}`

Selecting a caption font means saving all the code snippets in `\caption<name>`.

```

547 \newcommand*\caption@setfont[1]{%
548   \expandafter\let\csname caption#1\endcsname\@empty
549   \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.

```

550 \newcommand*\caption@addtofont[2]{%
551   \begingroup
552     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
553     \define@key{caption@fnt}{default}[]{%
554       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
555     \caption@setkeys[caption]{caption@fnt}{#2}%
556     \global\let\caption@tempa\caption@fnt
557   \endgroup
558   \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`.

```

559 \newcommand*\caption@font{%
560   \caption@teststar\caption@@font\@firstofone
561   {\caption@setkeys[caption]{caption@fnt}}%
562 \newcommand*\caption@@font[2]{%
563   \begingroup
564   \def\caption@fnt{\endgroup}%
565   #1{#2}%
566   \caption@fnt}

```

These are the pre-defined font code snippets.

```

567 \DeclareCaptionFont{normalcolor}{\normalcolor}
568 \DeclareCaptionFont{color}{\color{#1}}

569 \DeclareCaptionFont{normalfont}{\normalfont}
570 \DeclareCaptionFont{up}{\upshape}
571 \DeclareCaptionFont{it}{\itshape}
572 \DeclareCaptionFont{sl}{\slshape}
573 \DeclareCaptionFont{sc}{\scshape}
574 \DeclareCaptionFont{md}{\mdseries}
575 \DeclareCaptionFont{bf}{\bfseries}
576 \DeclareCaptionFont{rm}{\rmfamily}
577 \DeclareCaptionFont{sf}{\sffamily}
578 \DeclareCaptionFont{tt}{\ttfamily}

579 \DeclareCaptionFont{scriptsize}{\scriptsize}
580 \DeclareCaptionFont{footnotesize}{\footnotesize}
581 \DeclareCaptionFont{small}{\small}
582 \DeclareCaptionFont{normalsize}{\normalsize}

```

```

583 \DeclareCaptionFont{large}{\large}
584 \DeclareCaptionFont{Large}{\Large}

585 \DeclareCaptionFont{singlespacing}{\setstretch\setstretch@singlespacing}% normally 1
586 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
587 \DeclareCaptionFont{doublespacing}{\doublespacing}
588 \DeclareCaptionFont{stretch}{\setstretch{#1}}
589 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }

590 %\DeclareCaptionFont{normal}{%
591 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
592 \DeclareCaptionFont{normal}{%
593   \caption@font*{%
594     \KV@caption@fnt@normalcolor\@unused
595     \KV@caption@fnt@normalfont\@unused
596     \KV@caption@fnt@normalsize\@unused
597     \KV@caption@fnt@singlespacing\@unused}}

```

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

```

598 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
599 \DeclareCaptionDefaultFont{size}{}

```

1.13 Justifications

```

\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
600 \newcommand*\DeclareCaptionJustification[2]{%
601   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
602   \DeclareCaptionFont{#1}{#2}}
603 \@onlypreamble\DeclareCaptionJustification

\captionDefaultJustification \DeclareCaptionDefaultJustification{<code>}
604 \newcommand*\DeclareCaptionDefaultJustification[1]{%
605   \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
606   \DeclareCaptionDefaultFont{@hj}{#1}}
607 \@onlypreamble\DeclareCaptionDefaultJustification

608 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
609 \DeclareCaptionDefaultJustification{}

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

```

These are the pre-defined justification code snippets.

```

611 \DeclareCaptionJustification{justified}{}
612 \DeclareCaptionJustification{centering}{\centering}
613 \DeclareCaptionJustification{centerfirst}{\centerfirst}
614 \DeclareCaptionJustification{centerlast}{\centerlast}
615 \DeclareCaptionJustification{raggedleft}{\raggedleft}
616 \DeclareCaptionJustification{raggedright}{\raggedright}

```

```

\centerfirst Please blame Frank Mittelbach for the code of \centerfirst :-)
617 \providecommand\centerfirst{%

```

```

618 \let\\\@centercr
619 \edef\caption@normaladjust{%
620   \leftskip\the\leftskip
621   \rightskip\the\rightskip
622   \parfillskip\the\parfillskip\relax}%
623 \leftskip\z@\@plus -1fil%
624 \rightskip\z@\@plus 1fil%
625 \parfillskip\z@skip
626 \noindent\hskip\z@\@plus 2fil%
627 \setpar{\@par\@restorepar\caption@normaladjust}}

```

\centerlast This is based on code from Anne Brüggemann-Klein[22]

```

628 \providecommand\centerlast{%
629   \let\\\@centercr
630   \leftskip\z@\@plus 1fil%
631   \rightskip\z@\@plus -1fil%
632   \parfillskip\z@\@plus 2fil\relax}

```

1.13.1 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.

```

633 \DeclareCaptionJustification{Centering}{%
634   \caption@ragged\Centering\centering}
635 \DeclareCaptionJustification{RaggedLeft}{%
636   \caption@ragged\RaggedLeft\raggedleft}
637 \DeclareCaptionJustification{RaggedRight}{%
638   \caption@ragged\RaggedRight\raggedright}

```

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

```

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

```

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

```

639 \newcommand*\caption@ragged{%
640   \caption@Debug{We need ragged2e}%
641   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
642   \global\let\caption@ragged\caption@@ragged
643   \caption@ragged}

644 \caption@AtBeginDocument{%
645   \@ifundefined{caption@r@ragged2e}{%
646     \newcommand*\caption@@ragged{%
647       \caption@Warning{%
648         'ragged2e' support has been changed.\MessageBreak
649         Rerun to get captions right}%
650     \global\let\caption@ragged\@secondoftwo % suppress further warnings
651     \caption@ragged}%
652   }{%
653     \caption@Debug{We load ragged2e}%

```

```

654 \IfFileExists{ragged2e.sty}{%
655 \RequirePackage{ragged2e}%
656 \let\caption@@ragged\@firstoftwo
657 }{%
658 \newcommand*\caption@@ragged[2]{%
659 \ifundefined{caption\string#1}{%
660 \caption@Warning{%
661 'ragged2e' package not loaded, therefore\MessageBreak
662 substituting \string#2 for \string#1\MessageBreak}%
663 \global\@namedef{caption\string#1}}}%
664 #2}%
665 }%
666 }}

```

1.14 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`

```

667 \ifundefined{abovecaptionskip}{%
668 \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
669 \ifundefined{belowcaptionskip}{%
670 \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

671 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
672 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
673 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

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.

```

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

```

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

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

```

675 \newcommand*\caption@setposition[1]{%
676 \caption@ifinlist{#1}{d,default}{%
677 \let\caption@position\caption@defaultpos
678 }{\caption@ifinlist{#1}{t,top,above}{%
679 \let\caption@position\@firstoftwo
680 }{\caption@ifinlist{#1}{b,bottom,below}{%
681 \let\caption@position\@secondoftwo
682 }{\caption@ifinlist{#1}{a,auto}{%
683 \let\caption@position\@undefined
684 }{%
685 \caption@Error{Undefined position `#1'}%
686 }}}}}

```

`\caption@defaultpos` The default ‘position’ is ‘auto’, this means that the caption package 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 L^AT_EX document classes, but ‘auto’ should give better results in most cases.

```
687 %\caption@setdefaultpos{a}% default = auto
688 \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.)

```
689 \newcommand*\caption@iftop{%
690   \ifx\caption@position\@undefined
691     \let\caption@position\@secondoftwo
692 %   = \caption@setposition b%
693   \fi
694   \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.

```
695 \newcommand*\caption@fixposition{%
696   \ifx\caption@position\@undefined
697     \caption@autoposition
698   \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[20] 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.

```
699 \newcommand*\caption@autoposition{%
700   \ifvmode
701     \edef\caption@tempa{\the\prevdepth}%
702     \caption@Debug{\protect\prevdepth=\caption@tempa}%
703     \ifdim\prevdepth>-\p@
704       \let\caption@position\@secondoftwo
705     \else
706       \let\caption@position\@firstoftwo
707     \fi
708 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
709   \else
710     \caption@Debug{no \protect\prevdepth}%
711     \let\caption@position\@secondoftwo
712 %   = \caption@setposition b%
713   \fi}
```

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

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


1.16 Hooks

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

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

```
716 \newcommand*\caption@beginhook{}
717 \newcommand*\caption@endhook{}
718 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
719 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}
```

1.17 Lists

```
720 \DeclareCaptionOption{list}{\caption@setlist{#1}}
721 \DeclareCaptionOption{listof}{\caption@setlist{#1}}
```

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

```
\DeclareCaptionListFormat \DeclareCaptionListFormat {<name>} {<code with #1 and #2>}
723 \newcommand*\DeclareCaptionListFormat[2]{%
724   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
725 \@onlypreamble\DeclareCaptionListFormat

726 \DeclareCaptionOption{listformat}{\caption@setlistformat{#1}}
```

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

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

```
727 \newcommand*\caption@setlistformat[1]{%
728   \@ifundefined{caption@lstfmt@#1}%
729     {\caption@Error{Undefined list format `#1'}}%
730     {\expandafter\let\expandafter\caption@lstfmt
731       \csname caption@lstfmt@#1\endcsname}}
```

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

```
732 \DeclareCaptionListFormat{empty}{}
733 \DeclareCaptionListFormat{simple}{#1#2}
734 \DeclareCaptionListFormat{parens}{#1(#2)}
735 \DeclareCaptionListFormat{subsimple}{#2}
736 \DeclareCaptionListFormat{subparens}{(#2)}

737 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}
```

1.18 Debug option

```
738 \DeclareCaptionOption{debug}[1]{%
739   \caption@set@bool\caption@ifdebug{#1}%
740   \caption@ifdebug
741     {\def\caption@Debug{\PackageInfo{caption}}}%
742     {\let\caption@Debug@gobble}}

743 \DeclareOption{debug}{\setkeys{caption}{debug}}
744 \setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

1.19.1 The standard L^AT_EX classes

```
745 \caption@CheckCommand\@makecaption{%
746 % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
747 \long\def\@makecaption#1#2{%
748   \vskip\abovecaptionskip
749   \sbox\@tempboxa{#1: #2}%
750   \ifdim \wd\@tempboxa >\hsize
751     #1: #2\par
752   \else
753     \global \@minipagefalse
754     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
755   \fi
756   \vskip\belowcaptionskip}}
```

1.19.2 The $\mathcal{A}_M\mathcal{S}$ & SMF classes

```
757 \ifundefined{@captionheadfont}{}{%
758   \caption@CheckCommand\@makecaption{%
759     % amsart|amsproc|amsbook [2004/08/06 v2.20]
760     \long\def\@makecaption#1#2{%
761       \setbox\@tempboxa\vbox{\color@setgroup
762         \advance\hsize-2\captionindent\noindent
763         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
764           {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
765         \unskip\kern-2\captionindent\par
766         \global\setbox\@ne\lastbox\color@endgroup}%
767       \ifhbox\@ne % the normal case
768         \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
769       \fi
770       \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
771         \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
772       \else % tempboxa contained more than one line
773         \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
774           \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
775       \fi
776       \ifnum\@tempcnta<64 % if the float IS a figure...
777         \addvspace\abovecaptionskip
778         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
779       \else % if the float IS NOT a figure...
780         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
781         \nobreak
782         \vskip\belowcaptionskip
783       \fi
784       \relax
785     }}
786 \caption@CheckCommand\@makecaption{%
787 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publiés par
788 \long\def\@makecaption#1#2{%
789   \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
790   \setbox\@tempboxa\vbox{\color@setgroup
791     \advance\hsize-2\captionindent\noindent
792     \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
```

```

793         {\@cdr#2\@nil}{\@addpunct{.}\@captionfont\upshape\enspace#2}%
794         \unskip\kern-2\captionindent\par
795         \global\setbox\@ne\lastbox\color@endgroup}%
796     \ifhbox\@ne % the normal case
797         \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
798     \fi
799     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
800         \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
801         \@tempdima\wd\@ne\advance\@tempdima-\captionindent
802         \wd\@ne\@tempdima
803     \else % tempboxa contained more than one line
804         \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
805             \unvbox\@tempboxa\parskip\z@skip
806             \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
807     \fi
808     \ifnum\@tempcnta<64 % if the float IS a figure...
809         \addvspace\abovecaptionskip
810         \noindent\kern\captionindent\box\@ne
811     \else % if the float IS NOT a figure...
812         \noindent\kern\captionindent\box\@ne
813         \nobreak
814         \vskip\belowcaptionskip
815     \fi
816     \relax
817 }

818 \let\captionmargin\captionindent % set to 3pc by AMS class
819 \begingroup\edef\@tempa{\endgroup
820     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
821     {margin=\the\captionmargin
822     \@ifundefined{smf@makecaption}{},{,maxmargin=.1\linewidth}}}}
823 \@tempa
824 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
825 \DeclareCaptionLabelSeparator{default}{.\enspace}
826 \DeclareCaptionDefaultFont{font}{\@captionfont}
827 \DeclareCaptionDefaultFont{labelfont}{\@captionheadfont}
828 \DeclareCaptionDefaultFont{textfont}{\@captionfont\upshape}
829 \captionsetup[figure]{position=b}
830 \captionsetup[table]{position=t}
831 }

```

1.19.3 The beamer class

```

832 \@ifclassloaded{beamer}{%
833     \caption@CheckCommand\beamer@makecaption{%
834         % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
835         \long\def\beamer@makecaption#1#2{%
836             \def\insertcaptionname{\csname#1name\endcsname}%
837             \def\insertcaptionnumber{\csname the#1\endcsname}%
838             \def\insertcaption{#2}%
839             \nobreak\vskip\abovecaptionskip\nobreak
840             \sbox\@tempboxa{\usebeamertemplate**{caption}}}%
841             \ifdim\wd\@tempboxa>\hsize
842                 \usebeamertemplate**{caption}\par
843             \else

```

```

844     \global \@minipagefalse
845     \hb@xt@ \hsize { \hfil \box \@tempboxa \hfil } %
846     \fi
847     \nobreak \vskip \belowcaptionskip \nobreak } }

848 \DeclareCaptionLabelFormat {default} {#1}
849 \DeclareCaptionDefaultJustification { \raggedright }
850 \DeclareCaptionDefaultFont {font} { %
851     \usebeamerfont * {caption} %
852     \usebeamercolor [fg] {caption} }
853 \DeclareCaptionDefaultFont {labelfont} { %
854     \usebeamercolor [fg] {caption name} %
855     \usebeamerfont * {caption name} }

```

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.)

```

856 \defbeamertemplate{caption}{caption3}{ %
857     \caption@make \insertcaptionname \insertcaptionnumber \insertcaption }
858 \DeclareOption{beamer}{ %
859     % \usebeamertemplate*{caption} will set font
860     \DeclareCaptionDefaultFont{font}{} %
861     \setbeamertemplate{caption}[caption3] }
862 } { }

```

1.19.4 The KOMA-Script classes

```

863 \@ifundefined{scr@caption}{}{ %
864     \caption@CheckCommand \@makecaption { %
865         % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
866         \long \def \@makecaption #1 #2 { %
867             \if@captionabove
868                 \vskip \belowcaptionskip
869             \else
870                 \vskip \abovecaptionskip
871             \fi
872             \@@makecaption \@firstofone {#1} {#2} %
873             \if@captionabove
874                 \vskip \abovecaptionskip
875             \else
876                 \vskip \belowcaptionskip
877             \fi } }
878 \DeclareCaptionFormat {default} [#1 #2 #3 \par] { %
879     \ifdofullc@p
880         \caption@ifin@list \caption@lsep@crlist \caption@lsep@name
881         { \caption@Error { %
882             The option ‘labelsep=\caption@lsep@name’ does not work \MessageBreak
883             with \noexpand\setcaphanging (which is set by default) } } %
884         { \caption@fmt@hang {#1} {#2} {#3} } %
885     \else
886         #1 #2 %
887         \ifdim \cap@indent < \z@
888             \par
889             \noindent \hspace * { - \cap@indent } %

```

```

890     \else\if@capbreak
891     \par
892     \fi\fi
893     #3\par
894     \fi}
895 \DeclareCaptionLabelSeparator{default}{\captionformat}
896 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
897 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
898 }

```

1.19.5 The NTG Dutch classes

```

899 \@ifundefined{CaptionFonts}{}{%
900   \caption@CheckCommand\@makecaption{%
901     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
902     \long\def\@makecaption#1#2{%
903       \vskip\abovecaptionskip
904       \sbox\@tempboxa{\captionlabelfont#1:} \captionfont#2}%
905       \ifdim \wd\@tempboxa >\hsize
906         \captionlabelfont#1: \captionfont#2\par
907       \else
908         \global \@minipagefalse
909         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
910       \fi
911       \vskip\belowcaptionskip}}
912 \DeclareCaptionDefaultFont{labelfont}{\captionlabelfont}
913 \DeclareCaptionDefaultFont{textfont}{\captionfont}
914 }

```

1.19.6 The thesis class

```

915 \@ifclassloaded{thesis}{%
916   \caption@CheckCommand\@makecaption{%
917     % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
918     \long\def\@makecaption#1#2{%
919       \vskip\abovecaptionskip
920       \setbox\@tempboxa\hbox{\captionfont #1:} {\captionfont #2}}%
921       \ifdim \wd\@tempboxa >\hsize
922         \@hangfrom{\captionfont #1:} {\captionfont #2\par}%
923       \else
924         \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
925       \fi
926       \vskip\belowcaptionskip}}
927 \DeclareCaptionDefaultFormat{hang}
928 \DeclareCaptionDefaultFont{labelfont}{\captionfont}
929 \DeclareCaptionDefaultFont{textfont}{\captionfont}
930 }{}

```

1.19.7 The frenchb Babel option

```

931 \@ifundefined{FB@makecaption}{}{%
932   \caption@CheckCommand\@makecaption{%
933     % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]

```

```

934 % frenchb.ldf [2007/10/05 v2.0e French support from the babel system]
935 \long\def\@makecaption#1#2{%
936   \vskip\abovecaptionskip
937   \sbox\@tempboxa{#1\CaptionSeparator #2}%
938   \ifdim \wd\@tempboxa >\hsize
939     #1\CaptionSeparator #2\par
940   \else
941     \global \@minipagefalse
942     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
943   \fi
944   \vskip\belowcaptionskip}}
945 \ifx\@makecaption\STD@makecaption
946   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
947   \def\caption@frenchb{% supress frenchb warning
948     \let\STD@makecaption\@makecaption
949     \let\FB@makecaption\@makecaption}
950 \else
951   \ifx\@makecaption\@undefined\else
952     \PackageInfo{caption}{%
953       The definition of \protect\@makecaption\space
954       has been changed,\MessageBreak
955       frenchb will NOT customize it}%
956   \fi
957 \fi
958 }

```

1.19.8 The frenchle/pro package

```

959 \@ifundefined{frenchTeXmods}{}{%
960   \caption@CheckCommand\@makecaption{%
961     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
962     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
963     \def\@makecaption#1#2{%
964       \ifFTY%
965         \def\@secondofmany##1##2\void{##2}%
966         \def\@tempa{\@secondofmany#2\void}%
967         \ifx\@tempa\empty%
968           \let\captionseparator\empty%
969         \fi%
970         \@mcORI{#1}{\relax\captionfont{#2}}%
971       \else
972         \@mcORI{#1}{#2}%
973       \fi}}
974   \caption@CheckCommand\@makecaption{%
975     % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
976     \def\@makecaption#1#2{%
977       \ifFTY%
978         \def\@secondofmany##1##2\void{##2}%
979         \protected@edef\@tempa{\@secondofmany#2\void}%
980         \ifx\@tempa\empty%
981           \let\captionseparator\empty%
982         \fi%
983         \@mcORI{#1}{\relax\captionfont{#2}}%

```

```

984         \else
985             \@mcORI{#1}{#2}%
986         \fi}}
987 \DeclareCaptionDefaultFont{textfont}{\itshape}%
988 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
989 }

```

1.20 Execution of options

```

990 \captionsetup{style=default,position=default,list=1,listformat=default,%
991             twoside=\if@twoside 1\else 0\fi}
992 \ProcessOptions*
993 \caption@IfCheckCommand{%
994     \caption@setbool{documentclass}{1}%
995 }{%
996     \caption@setbool{documentclass}{0}%
997     \PackageInfo{caption}{%
998         Unknown document class (or package),\MessageBreak
999         standard defaults will be used}%
1000     \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1001 }

```

1.21 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`.

```

1002 \newcommand*\caption@addcontentsline[2]{%
1003     \caption@iflist
1004     {\def\@tempa{#2}}%
1005     {\let\@tempa\@empty}%
1006     \ifx\@tempa\@empty \else
1007         {\let\\space
1008             \addcontentsline{\csname ext@#1\endcsname}{#1}%
1009             {\protect\numberline
1010              {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1011              {\ignorespaces #2}}}%
1012     \fi}

```

1.22 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.)
1013 \newif\ifcaption@star

\caption@fnum \caption@fnum{<float type>}
Typesets the caption label; as replacement for \fnum{<float type>}.
1014 \newcommand*\caption@fnum[1]{\caption@lstfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}

\caption@make \caption@make{<float name>}{<ref. number>}{<text>}
Typesets the caption.
1015 \newcommand\caption@make[2]{\caption@@make{\caption@lstfmt{#1}{#2}}}

```

```

\caption@@make \caption@@make{\caption label}{\caption text}
1016 \newcommand\caption@@make[2]{%
1017 \begingroup
1018 \caption@stepcounter
1019 \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
1020 \ifx\caption@maxmargin\undefined \else
1021 \ifdim\captionmargin>\caption@maxmargin\relax
1022 \captionmargin\caption@maxmargin\relax
1023 \fi
1024 \fi
1025 \ifx\caption@minmargin\undefined \else
1026 \ifdim\captionmargin<\caption@minmargin\relax
1027 \captionmargin\caption@minmargin\relax
1028 \fi
1029 \fi

Special single-line treatment (option singlelinecheck=)
1030 \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%

Typeset the left margin (option margin=)
1031 \caption@calcmargin
1032 \@tempdima\captionmargin
1033 \ifdim\captionmargin@=\z@ \else
1034 \caption@ifoddpagel{\advance\@tempdima\captionmargin@}%
1035 \fi
1036 \caption@ifh{\advance\@tempdima\caption@indent}%
1037 \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*.
1038 \@tempdima\captionwidth
1039 \caption@ifh{\advance\@tempdima-\caption@indent}%
1040 \caption@parbox\@tempdima{%

Typeset the indentation (option indentation=)
Bugfix 04-05-05: \hskip-\caption@indent replaced by \ifdim\caption@indent=\z@...
1041 \caption@ifh{%
1042 \ifdim\caption@indent=\z@
1043 \leavevmode
1044 \else
1045 \hskip-\caption@indent
1046 \fi}%

Typeset the caption itself and close the \caption@parbox
1047 \caption@@@make{#1}{#2}}%

Typeset the right margin (option margin=)
1048 \@tempdima\captionmargin
1049 \ifdim\captionmargin@=\z@ \else
1050 \caption@ifoddpagel{\advance\@tempdima\captionmargin@}{}%
1051 \fi
1052 \hspace\@tempdima

```



```

1053 \caption@endhook
1054 \endgroup
1055 \global\caption@starfalse}

\caption@calcmargin \caption@calcmargin
Calculate \captionmargin & \captionwidth, so both contain valid values.
1056 \newcommand*\caption@calcmargin{%
1057 \ifdim\captionwidth=\z@
1058 \captionwidth\linewidth
1059 \advance\captionwidth by -2\captionmargin
1060 \advance\captionwidth by -\captionmargin@
1061 \else
1062 \captionmargin\linewidth
1063 \advance\captionmargin by -\captionwidth
1064 \divide\captionmargin by 2
1065 \captionmargin@\z@
1066 \fi

1067 \caption@Debug{%
1068 \string\hsize=\the\hsize,
1069 \string\linewidth=\the\linewidth,\MessageBreak
1070 \string\leftmargin=\the\leftmargin,
1071 \string\rightmargin=\the\rightmargin,\MessageBreak
1072 \string\margin=\the\captionmargin,
1073 \string\margin@=\the\captionmargin@,
1074 \string\width=\the\captionwidth}%
1075 }

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.
1076 \newcommand\caption@slc[4]{%
1077 \caption@Debug{Begin SLC}%
1078 \begingroup
1079 \caption@singleline
1080 \let\caption@hj\@empty
1081 \caption@calcmargin % calculate #3 if necessary
1082 \caption@prepareslc
1083 \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1084 \ifdim\wd\@tempboxa>#3%
1085 \endgroup
1086 \else
1087 \endgroup
1088 \caption@singleline
1089 #4%
1090 \fi
1091 \caption@Debug{End SLC}}

1092 \newcommand*\caption@singleline{%
1093 \caption@xsetup\caption@opt@singleline
1094 \let\caption@fmt\caption@slfmt}

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1095 \newcommand*\caption@prepareslc{%

```

```

1096 \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1097 \def\label{\caption@withoptargs\@gobbletwo}%
1098 \let\stepcounter\caption@l@stepcounter
1099 \let\refstepcounter\stepcounter\let\H\refstepcounter\stepcounter}

1100 \newcommand*\caption@l@stepcounter[1]{\advance\csname c@#1\endcsname\@ne\relax}

\caption@parbox \caption@parbox{\langle width\rangle}{\langle contents\rangle}
This macro defines the box which surrounds the caption paragraph.
1101 \newcommand*\caption@parbox{\parbox[b]}

\caption@@@make \caption@@@make{\langle caption label\rangle}{\langle caption text\rangle}
This one finally typesets the caption paragraph, without margin and indentation.
1102 \newcommand\caption@@@make[2]{%
If the label is empty, we use no caption label separator.
1103 \sbox\@tempboxa{#1}%
1104 \ifdim\wd\@tempboxa=\z@
1105 \let\caption@lsep\relax
1106 % \@capbreakfalse
1107 \fi
If the text is empty, we use no caption label separator, too.
1108 \caption@ifempty{#2}{%
1109 \let\caption@lsep\relax
1110 % \@capbreakfalse
1111 % \let\caption@ifstrut\@secondoftwo
1112 }%
Take care that \caption@parindent and \caption@hangindent will be used
to typeset the paragraph.
1113 \@setpar{\@@par\caption@@par}\caption@@par
Finally typeset the caption.
1114 \caption@hj\captionfont\captionsize\caption@fmt
1115 {\ifcaption@star\else\captionlabelfont#1\fi}%
1116 {\ifcaption@star\else\caption@iflf\captionlabelfont\caption@lsep\fi}%
1117 {\captiontextfont
1118 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1119 \nobreak\hskip\z@skip % enable hyphenation
1120 \caption@tfmt{#2}%
1121 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1122 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1123 \par}}

\caption@ifempty \caption@ifempty{\langle text\rangle}{\langle true\rangle} (no \langle false\rangle)
This one tests if the \langle text\rangle is actually empty.
Note: This will be done without expanding the text, therefore this is far away from being bullet-
proof.
Note: This macro is re-defining itself so only the first test (in a group) will actually be done.
1124 \newcommand\caption@ifempty[1]{%
1125 \caption@ifempty{#1}%
1126 \caption@ifempty\@unused}

```

```

1127 \newcommand\caption@if@empty[1]{%
1128   \def\caption@tempa{#1}%
1129   \ifx\caption@tempa\@empty
1130     \let\caption@ifempty\@secondoftwo
1131   \else
1132     \expandafter\def\expandafter\caption@tempa\expandafter{%
1133       \caption@car#1\caption@if@empty\caption@nil}%
1134     \def\caption@tempb{\caption@if@empty}%
1135     \ifx\caption@tempa\caption@tempb
1136       \let\caption@ifempty\@secondoftwo
1137     \else
1138       \def\caption@tempb{\ignorespaces}%
1139       \ifx\caption@tempa\caption@tempb
1140         \expandafter\caption@if@empty\expandafter{\@gobble#1}%
1141       \else
1142         \def\caption@tempb{\label}%
1143         \ifx\caption@tempa\caption@tempb
1144           \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1145         \else
1146           \let\caption@ifempty\@gobbletwo
1147         \fi
1148       \fi
1149     \fi
1150   \fi}
1151 \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.

```

1152 \newcommand*\caption@@par{%
1153   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

1.23 Types & sub-types

```

\DeclareCaptionType \DeclareCaptionType[<options>]{<environment>}[<name>][<list name>]
1154 \newcommand*\DeclareCaptionType{%
1155   \@testopt\@DeclareCaptionType{}%
1156   \@onlypreamble\DeclareCaptionType
1157   \def\@DeclareCaptionType[#1]#2{%
1158     \def\caption@type{#2}%
1159     \caption@Debug{New type `#2'}%
1160     \newcounter{#2}\@namedef{theH#2}{}%
1161     \KV@caption@DCT@within{\@ifundefined{c@chapter}{none}{chapter}}%
1162     \KV@caption@DCT@placement{tbp}%
1163     \@ifundefined{c@float@type}%
1164       {\newcounter{float@type}%
1165        \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1166       {}%
1167     \caption@Debug{float type `#2'=\the\value{float@type}}%
1168     \expandafter\xdef\csname ftype#2\endcsname{\the\value{float@type}}%
1169     \addtocounter{float@type}{\value{float@type}}%
1170     \KV@caption@DCT@fileext{lo#2}%
1171     \@namedef{fnum#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1172     \newenvironment{#2}{\@float{#2}}{\end@float}%

```

```

1173 \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1174 \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1175 \@ifundefined{l@figure}%
1176   {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1177   {\expandafter\let\csname l@#2\endcsname l@figure}%
1178 \expandafter\newcommand\csname #2name\endcsname{}%
1179 \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
1180 \uppercase\expandafter{\@tempa}%
1181 \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
1182 \@tempb
1183 \expandafter\let\csname #2name\endcsname\@tempa
1184 \expandafter\newcommand\csname list#2name\endcsname{}%
1185 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1186 \@cons\caption@typelist{{#2}}%
1187 \caption@setkeys[caption]{caption@DCT}{#1}%
1188 \@ifnextchar[\@@DeclareCaptionType\relax}
1189 \@onlypreamble\@@DeclareCaptionType
1190 \def\@@DeclareCaptionType[#1]{%
1191   \KV@caption@DCT@name{#1}%
1192   \@ifnextchar[\@@@DeclareCaptionType\relax}
1193 \@onlypreamble\@@@DeclareCaptionType
1194 \def\@@@DeclareCaptionType[#1]{%
1195   \KV@caption@DCT@listname{#1}}
1196 \@onlypreamble\@@@DeclareCaptionType
1197 \let\DeclareFloatingEnvironment\DeclareCaptionType % old command name

```

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

```

1198 \newcommand*\caption@listof[1]{%
1199   \begingroup
1200     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1201     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1202     \let\caption@ORI@starttoc\@starttoc
1203     \renewcommand*\@starttoc[1]{%
1204       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1205     \listoffigures
1206   \endgroup}

```

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

```
1207 \newcommand*\caption@typelist{}
```

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

```

1208 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
1209 \@onlypreamble@key{caption@DCT}{fileext}
1210 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1211 \@onlypreamble@key{caption@DCT}{listname}
1212 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1213 \@onlypreamble@key{caption@DCT}{name}
1214 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
1215 \@onlypreamble@key{caption@DCT}{placement}
1216 \define@key{caption@DCT}{within}{%
1217   \@ifundefined{c@chapter}{\@removefromreset\caption@type{chapter}}%

```

```

1218 \@removefromreset\caption@type{section}%
1219 \begingroup
1220 \caption@setkeys[caption]{caption@within}{#1}%
1221 \endgroup
1222 \@onlypreamble@key{caption@DCT}{within}
1223 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1224 \@onlypreamble@key{caption@DCT}{without}

1225 \define@key{caption@within}{none}[]{%
1226 \caption@within{}}{}
1227 \@onlypreamble@key{caption@within}{none}
1228 \@ifundefined{c@chapter}{%
1229 \define@key{caption@within}{section}[]{%
1230 \@addtoreset\caption@type{section}%
1231 \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
1232 }{%
1233 \define@key{caption@within}{chapter}[]{%
1234 \@addtoreset\caption@type{chapter}%
1235 \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
1236 \@onlypreamble@key{caption@within}{chapter}
1237 \define@key{caption@within}{section}[]{%
1238 \@addtoreset\caption@type{chapter}%
1239 \@addtoreset\caption@type{section}%
1240 \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi
1241 \ifnum\c@section>\z@ \thesection.\fi}{%
1242 \theHchapter.\theHsection.}}
1243 }\@onlypreamble@key{caption@within}{section}

```

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

```

1244 \newcommand*\caption@within{%
1245 \expandafter\let\csname c@#1\endcsname\@removefromreset
1246 \@onlypreamble\caption@within
1247 \newcommand*\caption@within@[3]{%
1248 \global\@namedef{the#1}{#2\arabic{#1}}%
1249 \@ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
1250 {\global\@namedef{theH#1}{#3\arabic{#1}}}%
1251 \@onlypreamble\caption@within@

```

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

```

1252 \providecommand*\@removefromreset[2]{%
1253 \expandafter\let\csname c@#1\endcsname\@removefromreset
1254 \def\@elt##1{%
1255 \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1256 \else
1257 \noexpand\@elt{##1}%
1258 \fi}%
1259 \expandafter\xdef\csname cl@#2\endcsname{%
1260 \csname cl@#2\endcsname}}

```

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

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

```

1261 \newcommand*\DeclareCaptionSubType{%
1262   \caption@teststar\@DeclareCaptionSubType\@firstoftwo\@secondoftwo}
1263 \@onlypreamble\DeclareCaptionSubType

1264 \newcommand*\@DeclareCaptionSubType[1]{%
1265   \@testopt{\@@DeclareCaptionSubType{#1}}{alph}}
1266 \@onlypreamble\@DeclareCaptionSubType

1267 \def\@@DeclareCaptionSubType#1[#2]#3{%
1268   \ifundefined{c@#3}%
1269     {\caption@Error{No float type '#3' defined}}%
1270   {\@ifundefined{c@sub#3}%
1271     {\caption@Debug{New subtype `sub#3'}}%
1272     \newcounter{sub#3}%
1273     \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1274     \ifundefined{l@chapter}%
1275       {\edef\@tempa{\expandafter\expandafter\expandafter\noexpand
1276         \expandafter\@car\l@subsubsection\@nil}%
1277       \def\@tempb{\@dottedtocline}%
1278       \ifx\@tempa\@tempb % l@subsubsection starts with \@dottedtocline
1279         \expandafter\edef\csname l@sub#3\endcsname{%
1280           \noexpand\@dottedtocline{2}%
1281           \expandafter\expandafter\expandafter\noexpand
1282           \expandafter\@gobbletwo\l@subsubsection}%
1283       \else
1284         \@namedef{l@sub#3}{\@dottedtocline{2}{3.8em}{3.2em}}%
1285       \fi}%
1286     {\expandafter\let\csname l@sub#3\endcsname\l@subsection}%
1287     \@cons\caption@subtypelist{#3}}%
1288     {\caption@Debug{Modify caption `sub#3'}}}%

1289   \@namedef{sub#3name}{}%
1290   \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
1291   #1 is \@firstoftwo in star form, and \@secondoftwo otherwise
1292   {\@namedef{p@sub#3}}}%
1293   \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1294   {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1295     \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1296   \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}%
1297   }}

```

```

1298 \@onlypreamble\@@DeclareCaptionSubType

```

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

```

1299 \newcommand*\caption@subtypelist{}

```

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

```

1300 \newcommand*\caption@For{\caption@withoptargs\caption@@For}
1301 \@onlypreamble\caption@For

```

```

1302 \newcommand\caption@@For[3]{%
1303   \caption@AtBeginDocument#1{%
1304     \def\@elt##1{#3}%
1305     \@nameuse{caption@#2}%
1306     \let\@elt\relax}%
1307 \@onlypreamble\caption@@For

```

1.24 subfig package adaption

We have to make several adaption to the caption package v3.1 here.

```

1308 \caption@AtBeginDocument{%
1309   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1310   \ifx\subref\@tempa
1311     \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1312     \let\caption@setfloattype\@gobble
1313     \let\@dottedxxxline\sf@NEW@dottedxxxline
1314     \let\sf@subfloat\sf@NEW@subfloat

```

This is a very small bugfix for v1.2 and v1.3 of the subfig package, making \subref robust, so it works in captions, too.

```

1315   \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1316   \fi
1317   \let\sf@NEW@dottedxxxline\@undefined
1318   \let\sf@NEW@subfloat\@undefined}
1319 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1320   \begingroup
1321     \caption@setfloattype{#1}%
1322     \caption@setoptions{subfloat}%
1323     \caption@setoptions{sub#1}%
1324     \ifnum #3>\@nameuse{c@#2depth}\else
1325       \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1326     \fi
1327   \endgroup}
1328 \def\sf@NEW@subfloat{%
1329   \begingroup
1330     \caption@setfloattype\@capttype
1331     \sf@ifpositiontop{%
1332       \maincaptiontoptrue
1333     }{%
1334       \maincaptiontopfalse
1335     }%
1336     \caption@setoptions{subfloat}%
1337     \caption@setoptions{sub\@capttype}%
1338     \let\sf@oldlabel=\label
1339     \let\label=\subfloat@label
1340     \ifmaincaptiontop\else
1341       \advance\@nameuse{c@\@capttype}\@ne
1342     \fi
1343     \refstepcounter{sub\@capttype}%
1344     \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1345     \@ifnextchar [% %] match left bracket
1346       {\sf@@subfloat}%
1347       {\sf@@subfloat[\@empty]}%

```

2 Main package

2.1 Identification

```
1348 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1349 \ProvidesPackage{caption}[2008/03/20 v3.1g Customizing captions (AR)]
1350 %\@ifundefined{PackageRedefines}{}{\PackageRedefines{caption}{caption}}

\caption@Info 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.

1351 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1352 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1353 \RequirePackage{caption3}[2008/03/01] % needs v3.1g
```

2.3 Check against incompatible packages

```
1354 \@ifpackageloaded{caption2}{%
1355   \caption@Error{%
1356     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1357     the (current) caption package}%
1358   \endinput
1359 }{}

1360 \caption@AtBeginDocument{%
1361   \@ifpackageloaded{ftcap}{%
1362     \caption@IncompatiblePackage{ftcap}{}%
1363   }{}%
1364   \@ifpackageloaded{nonfloat}{%
1365     \caption@IncompatiblePackage{nonfloat}{.\MessageBreak
1366       Furthermore you should use \noexpand\captionof instead of\MessageBreak
1367       \noexpand\figcaption or \string\tabcaption}%
1368   }{}%
1369   \@ifpackageloaded{topcapt}{%
1370     \caption@IncompatiblePackage{topcapt}{}%
1371   }{}%
1372 }
```

\caption@Incompatible *Issue an ‘incompatible’ warning.*

```
1373 \newcommand*\caption@IncompatiblePackage[2]{%
1374   \PackageWarningNoLine{caption}{%
1375     Usage of the #1 package together with the\MessageBreak
1376     caption package is strongly not recommended.\MessageBreak
1377     Instead of loading the #1 package you should\MessageBreak
1378     use the caption package option\MessageBreak
1379     \space\space'tableposition=top'#2}%
1380   \caption@Info{Setting position=b for compatibility reasons}%
1381   \caption@setposition b}

1382 \@onlypreamble\caption@IncompatiblePackage
```

2.4 Check document class

```
1383 \caption@ifbool{documentclass}{}{}%
```



```

1384 \PackageWarningNoLine{caption}{%
1385   Unsupported document class (or package) detected,\MessageBreak
1386   usage of the caption package is not recommended}%
1387 \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1388 }

```

2.5 Adaption to the $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes

```

1389 \@ifundefined{@captionheadfont}{}{%
1390   \caption@Info{AMS or SMF document class}%
1391   \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1392 }

```

2.6 Emulation of the KOMA-Script commands

```

1393 \@ifundefined{scr@caption}{}{%
1394   \caption@Info{KOMA-Script document class}%

```

Here we emulate the caption related commands and take over the caption related settings from the KOMA-Script classes.

```

\@tablecaptionabovetrue
\@tablecaptionabovefalse 1395 \g@addto@macro\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1396 \g@addto@macro\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1397 \if@tablecaptionabove
1398   \caption@AtBeginDocument{%
1399     \def\caption@tempa#1{%
1400       \ifpackageloaded{#1}{\caption@Error{%
1401         You can't use the global option\MessageBreak
1402         \space\space'tablecaptionabove'\MessageBreak
1403         and use the #1 package, too}}{}}%
1404       \caption@tempa{ftcap}%
1405       \caption@tempa{nonfloat}%
1406       \caption@tempa{topcapt}}%
1407   \@tablecaptionabovetrue
1408 \else
1409   \@tablecaptionabovefalse
1410 \fi

\onelinecaptionstrue
\onelinecaptionfalse 1411 \g@addto@macro\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1412 \g@addto@macro\onelinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1413 \ifonelinecaptions
1414   \onelinecaptionstrue
1415 \else
1416   \onelinecaptionfalse
1417 \fi

\@captionabovetrue
\@captionabovefalse 1418 \g@addto@macro\@captionabovetrue{\let\caption@position\@firstoftwo}
1419 \g@addto@macro\@captionabovefalse{\let\caption@position\@secondoftwo}

```

\setcapindent

```
1420 \let\caption@KOMA@setcapindent\@setcapindent
1421 \renewcommand*\@setcapindent[1]{%
1422   \caption@KOMA@setcapindent{#1}\caption@setcapindent}
1423 \let\caption@KOMA@@setcapindent\@@setcapindent
1424 \renewcommand*\@@setcapindent[1]{%
1425   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1426 \newcommand*\caption@setcapindent{%
1427   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1428 \@ifundefined{cap@indent}{\caption@setcapindent}
```

\setcapwidth *Note: The optional argument of \setcapwidth if not supported (yet), so we issue a warning if used. (Since this does not seem to have an negative effect when used by the captionbeside environment, we suppress the warning here.)*

```
1429 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1430   \csname\string\setcapwidth\endcsname
1431 \@namedef{\string\setcapwidth}[1]#2{%
1432   \caption@KOMA@setcapwidth{#1}{#2}\caption@setcapwidth{#1}}
1433 \newcommand*\caption@setcapwidth[1]{%
1434   \ifx\#1\\\else
1435     \@ifundefined{cap@margin}{\caption@Warning{
1436       \def\@tempa{captionbeside}%
1437       \ifx\@tempa\@currenvir\else\caption@Warning{
1438         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1439       \fi}%
1440     \fi
1441     \captionsetup{width=\cap@width}}
1442 \def\caption@tempa{\hsize}%
1443 \ifx\caption@tempa\cap@width \else
1444   \caption@setcapwidth{?}
1445 \fi
```

\setcapmargin

```
1446 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1447   \csname\string\@setcapmargin\endcsname
1448 \@namedef{\string\@setcapmargin}[1]#2{%
1449   \caption@KOMA@setcapmargin{#1}{#2}\caption@setcapmargin}
1450 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1451   \csname\string\@@setcapmargin\endcsname
1452 \@namedef{\string\@@setcapmargin}[1]#2{%
1453   \caption@KOMA@@setcapmargin{#1}{#2}\caption@setcapmargin}
1454 \newcommand*\caption@setcapmargin{%
1455   \begingroup
1456   \let\onelinecaptionsfalse\relax
1457   \def\@twoside{0}%
1458   \def\if@twoside{\def\@twoside{1}\iffalse}%
1459   \cap@margin
1460   \def\@tempa{\endgroup}%
1461   \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
```

```

1462         \def\hspace##1##{\@firstofone}%
1463         \edef\@tempa{\endgroup
1464             \noexpand\captionsetup{%
1465                 twoside=\@twoside,singlelinecheck=0,%
1466                 margin={\cap@left,\cap@right}}}%
1467         \fi\fi
1468         \@tempa}

1469 \ifx\cap@margin\relax \else
1470     \caption@setcapmargin
1471 \fi
1472 }

```

2.7 Declaration of options

2.7.1 Options for figure and table

```

1473 \DeclareCaptionOption{figureposition}{%
1474     \captionsetup*[figure]{position=#1}}
1475 \@onlypreamble@key{caption}{figureposition}

1476 \DeclareCaptionOption{tableposition}{%
1477     \caption@AtBeginDocument{%
1478         \def\caption@tempa##1{%
1479             \ifpackageloaded{##1}{\caption@Error{%
1480                 You can't use the caption package option\MessageBreak
1481                 \space\space`tableposition=#1'\MessageBreak
1482                 and use the ##1 package, too}}{}}}%
1483         \caption@tempa{ftcap}%
1484         \caption@tempa{nonfloat}%
1485         \caption@tempa{topcapt}}}%
1486     \captionsetup*[table]{position=#1}}
1487 \@onlypreamble@key{caption}{tableposition}

1488 \DeclareCaptionOption{figurename}{\caption@SetName{figure}{#1}}
1489 \DeclareCaptionOption{tablename}{\caption@SetName{table}{#1}}
1490 \DeclareCaptionOption{name}{\caption@setname\@captype{#1}}

1491 \DeclareCaptionOption{listfigurename}{\caption@SetName{listfigure}{#1}}
1492 \DeclareCaptionOption{listtablename}{\caption@SetName{listtable}{#1}}

\caption@SetName \caption@SetName{<cmd>}{<value>}

1493 \newcommand*\caption@SetName[2]{%
1494     \begingroup
1495     \def\@tempa{\endgroup\@firstofone}%
1496     \ifundefined{language\@tempa}{\@tempa}{%
1497         \ifundefined{captions\language\@tempa}{\@tempa}{%
1498             \def\@tempa{\endgroup
1499                 \expandafter\addto\csname captions\language\@tempa\endcsname}}}%
1500     \@tempa{\caption@setname{#1}{#2}}}

1501 \newcommand*\caption@setname[2]{\@namedef{#1name}{#2}}

1502 \caption@AtBeginDocument{\let\caption@SetName\caption@setname}

1503 \DeclareCaptionOption{figurewithin}{\caption@Within{figure}{#1}}
1504 \@onlypreamble@key{caption}{figurewithin}

```

```

1505 \DeclareCaptionOption{figurewithout}{\caption@Within{figure}{none}}
1506 \@onlypreamble@key{caption}{figurewithout}

1507 \DeclareCaptionOption{tablewithin}{\caption@Within{table}{#1}}
1508 \@onlypreamble@key{caption}{tablewithin}
1509 \DeclareCaptionOption{tablewithout}{\caption@Within{table}{none}}
1510 \@onlypreamble@key{caption}{tablewithout}

```

\caption@within

```

1511 \newcommand*\caption@Within[1]{\def\caption@type{#1}\KV@caption@DCT@within}
1512 \@onlypreamble\caption@Within

```

2.7.2 Miscellaneous options

```

1513 \DeclareCaptionOption*{config}[caption]{%
1514   \InputIfFileExists{#1.cfg}%
1515   {\typeout{*** Local configuration file #1.cfg used ***}}%
1516   {\caption@Warning{Configuration file #1.cfg not found}}}%

1517 \DeclareCaptionOption{@minipage}{%
1518   \caption@ifinlist{#1}{auto,default}%
1519   {\let\caption@if@minipage\@gobbles}%
1520   {\caption@set@bool\caption@if@minipage{#1}}}%
1521 \captionsetup{@minipage=default}

```

2.7.3 caption v1.x compatibility options

```

1522 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1523 \@onlypreamble@key{caption}{compatibility}

1524 \DeclareCaptionOptionNoValue*{normal}{%
1525   \caption@setformat{plain}%
1526   \caption@setjustification{justified}}
1527 \DeclareCaptionOptionNoValue*{isu}{%
1528   \caption@setformat{hang}%
1529   \caption@setjustification{justified}}
1530 \DeclareCaptionOptionNoValue*{hang}{%
1531   \caption@setformat{hang}%
1532   \caption@setjustification{justified}}
1533 \DeclareCaptionOptionNoValue*{center}{%
1534   \caption@setformat{plain}%
1535   \caption@setjustification{centering}}
1536 \DeclareCaptionOptionNoValue*{anne}{%
1537   \caption@setformat{plain}%
1538   \caption@setjustification{centerlast}}
1539 \DeclareCaptionOptionNoValue*{centerlast}{%
1540   \caption@setformat{plain}%
1541   \caption@setjustification{centerlast}}

1542 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1543 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1544 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1545 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1546 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1547 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1548 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1549 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}

```

```

1550 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1551 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1552 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1553 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1554 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1555 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1556 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}
1557 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}
1558 \caption@setbool{ruled}{0}
1559 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}

```

2.7.4 caption2 v2.x compatibility options

```

1560 \DeclareCaptionOptionNoValue*{flushleft}{%
1561   \caption@setformat{plain}%
1562   \caption@setjustification{raggedright}}
1563 \DeclareCaptionOptionNoValue*{flushright}{%
1564   \caption@setformat{plain}%
1565   \caption@setjustification{raggedleft}}
1566 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}
1567 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
1568   \caption@WarningNoLine{Obsolete option 'ignoreLTcapwidth' ignored}}

```

2.7.5 Obsolete caption v3.0 options

```

1569 \DeclareCaptionOption*{caption}{%
1570   \caption@setbool{temp}{#1}%
1571   \caption@ifbool{temp}{{}%
1572     \caption@Error{%
1573       The package option 'caption=#1' is obsolete.\MessageBreak
1574       Please pass this option to the subfig package instead\MessageBreak
1575       and do *not* load the caption package anymore}}}%

```

2.7.6 fltpage package support options

With these options is controlled where the list-of entry and `\ref` resp. `\pageref` or `\autoref` will link to. Defaults are `FPlist=caption` and `FPref=figure` which is inconsistent, but compatible to the usual behaviour of the `fltpage` package.

```

1576 \DeclareCaptionOption{FPlist}[1]{\caption@setFFOption{list}{#1}}
1577 \DeclareCaptionOption{FPref}[1]{\caption@setFFOption{ref}{#1}}
1578 \@onlypreamble@key{caption}{FPlist}
1579 \@onlypreamble@key{caption}{FPref}
1580 \newcommand*\caption@setFFOption[2]{%
1581   \edef\@tempa{\@car#2\@nil}%
1582   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1583 \@onlypreamble\caption@setFFOption
1584 \captionsetup{FPlist=caption,FPref=figure}

```

2.7.7 hyperref package support options

With `hypcap=off` one can turn the `hypcap` support off (default is on).

```

1585 \DeclareCaptionOption{hypcap}[1]{\caption@setbool{hypcap}{#1}}
1586 \DeclareCaptionOption{hypcapSPACE}{\def\caption@hypcapSPACE{#1}}
1587 \captionsetup{hypcap=1,hypcapSPACE=.5\baselineskip}

```

2.8 Processing of options

```
1588 \caption@ProcessOptions*{caption}
```

2.9 \captionof and \captionlistentry

```
1589 \caption@AtBeginDocument{%
1590   \DeclareCaptionOption{type}{\caption@settype{#1}}%
1591   \DeclareCaptionOption{type*}{\caption@settype*{#1}}%
1592   \DeclareCaptionOption{subtype}[sub\@capttype]{\caption@setsubtype{#1}}%
1593   \DeclareCaptionOption{subtype*}[sub\@capttype]{\caption@setsubtype*{#1}}%
1594 }
```

Important Note: Like `\captionof` the option `type=` should only be used inside a group, box, or environment and does not check if the argument is a valid floating environment or not.

`\caption@settype` `\caption@settype*{<type>}`
 sets `\@capttype` and executes the options associated with it (using `\caption@setoptions`). Furthermore we check `\currentgrouplevel` (if avail), redefine `\@currentlabel` so a `\label` before `\caption` will result in a hint instead of a wrong reference, and use the macro `\caption@(sub)typehook` (which will be used by our float package support).

The non-starred version sets a `hyperref` anchor additionally (if `hypcap=true` and the `hypcap` package is not loaded).

```
1595 \newcommand*\caption@settype{%
1596   \caption@@settype{}}
1597 \newcommand*\caption@setsubtype{%
1598   \caption@iftype
1599   {\caption@@settype{sub}}%
1600   {\caption@Error{Option 'subtype=' outside float}}}%
1601 \newcommand*\caption@@settype[1]{%
1602   \caption@teststar{\caption@@@settype{#1}}{\@firstoftwo\@secondoftwo}
1603 \newcommand*\caption@@@settype[3]{%
1604   % #1 = "" or "sub"
1605   % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1606   % #3 = <type>, e.g. "figure" or "table"
1607   \@ifundefined{c@#3}%
1608     {\caption@Error{No float type '#3' defined}}%
1609     {\caption@Debug{#1type=#3}}%
1610     \caption@checkgrouplevel{#1}{%
1611       \captionsetup{#1type#2*\@empty=...}#2{ or
1612         \@backslashchar#1captionof}{}}%
1613     \edef\@tempa{#3}%
1614     \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1615       \ifcaptionsetup@star\else\@nameuse{caption@#1type@warning}\fi
1616     \fi
1617     \expandafter\let\csname @#1capttype\endcsname\@tempa
1618     \@nameuse{caption@#1typehook}%
1619     \caption@setoptions{#3}%
1620     \ifx\caption@opt\relax
1621       \@nameundef{caption@#1type@warning}%

```

```

1622 \else
1623 \namedef{caption@#1type@warning}{\caption@Warning{%
1624 The #1caption type was already set to
1625 '\csname @#1capttype\endcsname'\MessageBreak}}%
1626 \fi
1627 \let\caption@ifrefstepcounter\@secondoftwo
1628 #2{}{%
1629 \let\@currentlabel\caption@undefinedlabel
1630 % \let\@currentHlabel\@undefined
1631 \ifx\caption@ORI@label\@undefined
1632 \let\caption@ORI@label\label
1633 \let\label\caption@xlabel
1634 \fi
1635 \caption@start}}

```

`\caption@typehook` Hook, will be extended later on, e.g. by our float package support.

```
1636 \newcommand*\caption@typehook{}
```

`\caption@iftype` Since we often need to check if `\@capttype` is defined (means: we are inside a floating environment) this helper macro was introduced.

```

1637 \newcommand*\caption@iftype{%
1638 \@ifundefined{@capttype}{\let\@capttype\@undefined\@secondoftwo}\@firstoftwo}

```

`\caption@checkgrouplevel` Checks if `\captionsetup{type=...}` or `\caption` is done inside a group or not – in the latter case a warning message will be issued. (needs \mathcal{E} - \TeX)

```

1639 \begingroup\expandafter\expandafter\expandafter\endgroup
1640 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1641 \caption@Debug{TeX engine: TeX}
1642 \let\caption@checkgrouplevel\@gobbletwo
1643 \else
1644 \caption@Debug{TeX engine: e-TeX}
1645 \newcommand*\caption@checkgrouplevel[2]{%
1646 \@ifundefined{#1caption@grouplevel}{%
1647 \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}}{%
1648 \ifnum\currentgrouplevel>\caption@grouplevel\relax
1649 \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1650 \the\currentgrouplevel}%
1651 \else
1652 \caption@Warning{\string#2\MessageBreak outside box or environment}%
1653 \fi
1654 }{} }
1655 \fi

```

`\caption@undefinedlabel` This label will be used for `\currentlabel` inside (floating) environments as default. (see above)

```
1656 \newcommand*\caption@undefinedlabel{??}
```

`\caption@xlabel` The new code of `\label` inside floating environments. `\label` will be redefined using `\caption@withoptargs`, so #1 are the optional arguments (if any), and #2 is the mandatory argument here.

```

1657 \newcommand*\caption@xlabel{%
1658 \ifx\@currentlabel\caption@undefinedlabel
1659 \caption@Warning{\noexpand\label before \noexpand\caption ignored}%

```

```

1660     \expandafter\@gobble
1661   \else
1662     \expandafter\caption@ORI@label
1663   \fi}

\captionof \captionof{<type>}[<lst.entry>]{<heading>}
\captionof* [<lst.entry>]{<heading>}
Note: This will be defined with \AtBeginDocument so \usepackage{caption,capt-of}
will still work. (Compatibility to v1.x)

1664 \caption@AtBeginDocument{%
1665   \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}

1666 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

\captionlistentry \captionlistentry[<float type>]{<list entry>}
\captionlistentry* [<float type>]{<list entry>}

1667 \newcommand*\captionlistentry{%
1668   \caption@teststar\@captionlistentry\@firstoftwo\@secondoftwo}

1669 \newcommand*\@captionlistentry[1]{%
1670   \@testopt{\caption@listentry{#1}}\@capttype}

1671 \def\caption@listentry#1[#2]#3{%
1672   \@bsphack
1673   #1{\def\@currentlabelname{#3}}%
1674   {\caption@refstepcounter{#2}}%
1675   \caption@makecurrent{#2}{#3}}%
1676   \caption@addcontentsline{#2}{#3}%
1677   \@esphack}

```

2.10 \ContinuedFloat

```

\ContinuedFloat \ContinuedFloat
\ContinuedFloat*

```

This mainly decrements the appropriate counter and increments the continuation counter instead. Furthermore we set `\caption@resetContinuedFloat` to `\@gobble` so the continuation counter will not be reset to zero inside `\caption@refstepcounter`. Please forget about the optional argument, it was never working well, is incompatible to the subfig package, but is still there for compatibility reasons.

Note: The definition of `\ContinuedFloat` itself is compatible to the one inside the subfig package, except for the starred variant and the optional argument.

When the `hyperref` package is used we have the problem that the usage of `\ContinuedFloat` will create duplicate hyper links – `\@currentHref` will be the same for the main float and the continued ones. So we have to make sure unique labels and references will be created each time. We do this by extending `\theHfigure` and `\theHtable`, so for continued floats the scheme

$$\langle type \rangle . \langle type \# \rangle \backslash \alpha \{ \langle continued \# \rangle \}$$

will be used instead of

$$\langle type \rangle . \langle type \# \rangle \quad .$$

(This implementation follows an idea from Steven Douglas Cochran.)

Note: This does not help if the `hyperref` package option `naturalnames=true` is set.

```

1678 \def\ContinuedFloat{%
1679   \@ifnextchar{\@Continued@Float\@ContinuedFloat}

1680 \def\@Continued@Float[#1]{\addtocounter{#1}\m@ne}

1681 \def\@ContinuedFloat{%
1682   \caption@iftype
1683     {\addtocounter\@c@ptype\m@ne
1684     \caption@ContinuedFloat\@c@ptype}%
1685     {\caption@Error{\noexpand\ContinuedFloat outside float}}}}

1686 \def\caption@ContinuedFloat#1{%
1687   \ifstar{\caption@Continued@Float@{#1}}{\caption@Continued@Float{#1}}}

1688 \def\caption@Continued@Float@{%
1689   \addtocounter\@c@ptype\@ne
1690   \@stpelt{ContinuedFloat}\stepcounter{ContinuedFloat}%
1691   \def\caption@resetContinuedFloat##1{\xdef\caption@CFtype{##1}}%
1692   \caption@@ContinuedFloat}

1693 \def\caption@Continued@Float#1{%
1694   \edef\@tempa{#1}%
1695   \ifx\@tempa\caption@CFtype
1696     \stepcounter{ContinuedFloat}%
1697     \let\caption@resetContinuedFloat\@gobble
1698     \caption@@ContinuedFloat{#1}%
1699     \sf@ContinuedFloat{#1}%
1700   \else
1701     \caption@Error{Continued `#1' after `\caption@CFtype'}%
1702   \fi}

1703 \def\caption@@ContinuedFloat#1{%
1704   \expandafter\l@addto@macro\c@name the#1\endcsname\theContinuedFloat
1705   \@ifundefined{theH#1}{}{%
1706     \expandafter\l@addto@macro\c@name theH#1\endcsname{%
1707       \@alph@c@ContinuedFloat}}%
1708   \caption@setoptions{ContinuedFloat}%
1709   \caption@setoptions{continued#1}}

1710 \providecommand*\sf@ContinuedFloat[1]{}

1711 \newcommand*\caption@CFtype{??}

```

`\theContinuedFloat` Its preset to `\@empty`, so usually the continuation counter is not included in the caption label or references.

```

1712 \newcounter{ContinuedFloat}
1713 \let\theContinuedFloat\@empty

```

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

If a continuation counter is defined, we reset it. (This one will be called inside `\@caption`.)

```

1714 \newcommand*\caption@resetContinuedFloat[1]{%
1715   \@stpelt{ContinuedFloat}\xdef\caption@CFtype{#1}}

```

2.11 Internal helpers

`\caption@refstepcounter` Resets the continuation counter, increments the float (i.e. figure or table) counter, and sets the `refstepcounter` flag.

```
1716 \newcommand*\caption@refstepcounter[1]{%
1717   \caption@resetContinuedFloat{#1}%
1718   \caption@@refstepcounter{#1}%
1719   \let\caption@ifrefstepcounter\@firstoftwo}

1720 \newcommand*\caption@@refstepcounter{\refstepcounter}
1721 \let\caption@ifrefstepcounter\@secondoftwo
```

`\caption@dblarg` A `\relax` was added compared to `\@dblarg` so `\caption{}` will be expanded to `\caption[\relax]{}` (and not to `\caption[]{}).`

```
1722 \@ifundefined{kernel@ifnextchar}%
1723   {\newcommand\caption@dblarg[1]{\@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1724   {\newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1725   \newcommand\caption@xdblarg[2]{#1[{\#2\relax}]{\#2}}}%
```

`\caption@begin` Our handling of `\caption` will always be surrounded by `\caption@begin` (or `\caption@beginex`) and `\caption@end`. `\caption@begin{<type>}` performs these tasks:

1. Start a new group.
2. Define `\fnum{<type>}` if the caption label format is set to non-default.
3. Override the `position=` setting, if necessary. (for example if set to `auto` or used inside a `supertabular`)

```
1726 \newcommand*\caption@begin[1]{%
1727   \begingroup
1728   \caption@setfnum{#1}%
1729   \caption@fixposition
1730   \global\let\caption@fixedposition\caption@position}
```

`\caption@beginex` `\caption@beginex{<type>}{<list entry>}{<heading>}` performs the same tasks as `\caption@begin` and additionally:

4. Make an entry in the list-of-whatever.
5. Set `\caption@ifempty` according argument `<heading>`.

```
1731 \newcommand\caption@beginex[3]{%
1732   \caption@begin{#1}%
1733   \caption@addcontentsline{#1}{#2}%
1734   \caption@ifempty{#3}{}}
```

`\caption@end` `\caption@end` closes the group.

```
1735 \newcommand*\caption@end{%
1736   \endgroup
1737   \let\caption@position\caption@fixedposition}
```

```

\caption@setfnum \caption@setfnum{<type>}
redefines \fnum@<type> according the caption label format set with labelformat=.
But if labelformat=default is set, \fnum@<type> will not be overwritten by us.
1738 \newcommand*\caption@setfnum[1]{%
1739   \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1740   \@namedef{fnum@#1}{\caption@fnum{#1}}}%
1741   \fi}

```

\caption@boxrestore The original code (from latex/base/ltboxes.dtx):

```

\def\@parboxrestore{\@arrayparboxrestore\let\\@normalcr}
\def\@arrayparboxrestore{%
  \let\if@nobreak\iffalse
  \let\if@noskipsec\iffalse
  \let\par\@par
  \let\-\@dischyph
  \let'\@acci\let\'@accii\let\=\@acciii
  \parindent\z@ \parskip\z@skip
  \everypar{}%
  \linewidth\hsize
  \@totalleftmargin\z@
  \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
  \parfillskip\@flushglue \lineskip\normallineskip
  \baselineskip\normalbaselineskip
  \sloppy}

```

This one will be used by \@caption instead of \@parboxrestore.

```

1742 \newcommand*\caption@boxrestore{%
1743   \let\if@nobreak\iffalse
1744   \let\if@noskipsec\iffalse
1745   \let\par\@par
1746 % \let\-\@dischyph
1747 % \let'\@acci\let\'@accii\let\=\@acciii
1748   \parindent\z@ \parskip\z@skip
1749   \everypar{}%
1750 % \linewidth\hsize
1751 % \totalleftmargin\z@
1752   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1753   \parfillskip\@flushglue \lineskip\normallineskip
1754   \baselineskip\normalbaselineskip
1755   \sloppy
1756   \let\\@normalcr
1757 }

```

\caption@normalsize This one will be used by \@caption instead of \normalsize.
Its code is equivalent to

```
\caption@font{normal}%
```

but executes faster (since the starred form of \caption@font does not use \setkeys internally).

```

1758 \newcommand*\caption@normalsize{%
1759   \caption@font*{\KV@caption@fnt@normal\@unused}}

```

`\caption@setfloatcapt` Needed for support of the float package, where the caption will not be typeset directly, but caught in a `\vbox` called `\@floatcapt` instead.

```
1760 \let\caption@setfloatcapt\@firstofone
```

`\caption@makecurrent` All these are needed for support of the hyperref package.

```
\caption@makeanchor 1761 \newcommand*\caption@makecurrent[2]{}
\caption@start      1762 \let\caption@makeanchor\@firstofone
\caption@@start     1763 \let\caption@start\relax
```

```
\caption@freezeHref 1764 \let\caption@@start\relax
```

```
\caption@defrostHref 1765 \let\caption@freezeHref\relax
```

```
\caption@defrostHref 1766 \let\caption@defrostHref\relax
```

2.12 `\caption`, `\@caption`, and `\@makecaption`

We only redefine `\caption` and `\@caption` if the current definitions are well known, so documents written in the old (caption package *v1.x*) days (where `\caption` & `\@caption` were not redefined by us) will still compile fine. For example the usage of the `captcont` package, which brings it's own definition of `\caption*`, was quite common these days.

Some packages (like the `hyperref` package for example) redefines `\caption` and `\@caption`, too. So we have to use `\AtBeginDocument` here, so we can make sure our definition is the one which will be valid at last.

```
1767 \caption@AtBeginDocument{%
```

```
1768   \caption@setbool{incompatible}{0}%
```

```
1769   \caption@CheckCommand\caption{%
```

```
1770     % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
```

```
1771     \def\caption{%
```

```
1772       \ifx\@captive\@undefined
```

```
1773         \@latex@error{\noexpand\caption outside float}\@ehd
```

```
1774         \expandafter\@gobble
```

```
1775       \else
```

```
1776         \refstepcounter\@captive
```

```
1777         \expandafter\@firstofone
```

```
1778       \fi
```

```
1779       {\@dblarg{\@caption\@captive}}%
```

```
1780     } }%
```

```
1781   \caption@CheckCommand\caption{%
```

```
1782     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
```

```
1783     \def\caption{
```

```
1784       \ifx\@captive\@undefined
```

```
1785         \@latex@error{\noexpand\caption outside figure or table}\@ehd
```

```
1786         \expandafter\@gobble
```

```
1787       \else
```

```
1788         \refstepcounter\@captive
```

```
1789         \expandafter\@firstofone
```

```
1790       \fi
```

```
1791       {\@dblarg{\@caption\@captive}}%
```

```
1792     } }%
```

```
1793   \caption@CheckCommand\caption{%
```

```
1794     % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
```

```
1795     \renewcommand\caption{%
```

```

1796 \ifx\@capttype\@undefined
1797 \latexerror{\noexpand\caption outside float}\@ehd
1798 \expandafter\@gobble
1799 \else
1800 \refstepcounter\@capttype
1801 \let\@tempf\@caption
1802 \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1803 \expandafter\expandafter\let
1804 \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1805 \fi
1806 \fi
1807 \@dblarg{\@tempf\@capttype}}}%
1808 \caption@CheckCommand\caption{%
1809 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1810 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1811 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1812 \def\caption{%
1813 \ifx\@capttype\@undefined
1814 \latexerror{\noexpand\caption outside float}\@ehd
1815 \expandafter\@gobble
1816 \else
1817 \H@refstepcounter\@capttype
1818 \@ifundefined{fst@\@capttype}{%
1819 \let\Hy@tempa\@caption
1820 }{%
1821 \let\Hy@tempa\Hy@float@caption
1822 }%
1823 \expandafter\@firstofone
1824 \fi
1825 {\@dblarg{\Hy@tempa\@capttype}}}%
1826 }}%
1827 \caption@CheckCommand\caption{%
1828 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1829 \def\caption{%
1830 \ifx\@capttype\@undefined
1831 \latexerror{\noexpand\caption outside float}\@ehd
1832 \expandafter\@gobble
1833 \else
1834 \H@refstepcounter\@capttype
1835 \let\Hy@tempa\@caption
1836 \@ifundefined{float@caption}{%
1837 }{%
1838 \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
1839 \let\Hy@tempa\Hy@float@caption
1840 \fi
1841 }%
1842 \expandafter\@firstofone
1843 \fi
1844 {\@dblarg{\Hy@tempa\@capttype}}}%
1845 }}%
1846 \caption@IfCheckCommand{}{%
1847 \caption@Info{%
1848 Incompatible package detected (regarding \string\caption).\MessageBreak

```

```

1849     \string\caption\space=\space\meaning\caption}%
1850     \caption@setbool{incompatible}{1}}%

1851 \caption@CheckCommand\@caption{%
1852 % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1853 \long\def\@caption#1[#2]#3{%
1854     \par
1855     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1856     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1857     \begingroup
1858     \@parboxrestore
1859     \if@minipage
1860         \setminipage
1861     \fi
1862     \normalsize
1863     \makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1864     \endgroup}}%

1865 \caption@CheckCommand\@caption{%
1866 % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1867 \long\def\@caption#1[#2]#3{% second argument ignored
1868     \par\nobreak
1869     \begingroup
1870     \@parboxrestore
1871     \if@minipage
1872         \setminipage
1873     \fi
1874     \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
1875     \endgroup}}%

1876 % \caption@CheckCommand\float@caption{%
1877 % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1878 % \long\def\float@caption#1[#2]#3{%
1879 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%
1880 %     {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
1881 %     \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
1882 %     \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
1883 %     \ifnextchar[{\float@ccon}{\egroup}}%
1884 %     \long\def\float@ccon[#1]{#1\par\egroup}}%

1885 \caption@CheckCommand\@caption{%
1886 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1887 \long\def\@caption#1[#2]#3{%
1888     \hyper@makecurrent{\@cuptype}%
1889     \def\@currentlabelname{#2}%
1890     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1891         \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1892     }%
1893     \begingroup
1894     \@parboxrestore
1895     \if@minipage
1896         \setminipage
1897     \fi
1898     \normalsize
1899     \makecaption{\csname fnum@#1\endcsname}{%
1900         \ignorespaces
1901         \ifHy@nesting

```

```

1902         \hyper@@anchor{\@currentHref}{#3}%
1903     \else
1904         \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1905     \fi
1906 }%
1907 \par
1908 \endgroup
1909 } }%

1910 \caption@CheckCommand\@caption{%
1911 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1912 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1913 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1914 \long\def\@caption#1[#2]#3{%
1915     \expandafter\ifx\csname if@capstart\expandafter\endcsname
1916         \csname iftrue\endcsname
1917     \global\let\@currentHref\hc@currentHref
1918 \else
1919     \hyper@makecurrent{\@capttype}%
1920 \fi
1921 \def\@currentlabelname{#2}%
1922 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1923     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1924 }%
1925 \begingroup
1926     \@parboxrestore
1927     \if@minipage
1928         \@setminipage
1929     \fi
1930     \normalsize
1931     \expandafter\ifx\csname if@capstart\expandafter\endcsname
1932         \csname iftrue\endcsname
1933     \global\@capstartfalse
1934     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1935 \else
1936     \@makecaption{\csname fnum@#1\endcsname}{%
1937         \ignorespaces
1938         \ifHy@nesting
1939             \hyper@@anchor{\@currentHref}{#3}%
1940         \else
1941             \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1942         \fi
1943     }%
1944 \fi
1945 \par
1946 \endgroup
1947 } }%

1948 \caption@CheckCommand\@caption{%
1949 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1950 \long\def\@caption#1[#2]{%
1951     \def\@currentlabelname{#2}%
1952     \NR@@caption{#1}[{#2}]%
1953 } }%

1954 \caption@CheckCommand\@caption{%

```

```

1955 % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1956 \long\def\@caption#1[#2]#3{%
1957   \ifundefined{if#1topcap}%
1958     {\subfig@oldcaption{#1}[{#2}]{#3}}%
1959     {\@nameuse{if#1topcap}%
1960       \@listsubcaptions{#1}%
1961       \subfig@oldcaption{#1}[{#2}]{#3}%
1962     }%
1963     {\subfig@oldcaption{#1}[{#2}]{#3}%
1964     \@listsubcaptions{#1}%
1965     \fi}}}%
1966 \caption@CheckCommand\@caption{%
1967   % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1968   \def\@caption{\caption@}%
1969 % \long\def\caption@#1[#2]#3{%
1970 %   \ifundefined{caption@setfloattype}%
1971 %     \caption@settype
1972 %     \caption@setfloattype
1973 %     \@cptype
1974 %     \sf@ifpositiontop{%
1975 %       \@listsubcaptions{#1}%
1976 %       \sf@old@caption{#1}[{#2}]{#3}%
1977 %     }{%
1978 %       \sf@old@caption{#1}[{#2}]{#3}%
1979 %       \@listsubcaptions{#1}%
1980 %     }%
1981 %   }%
1982 \caption@IfCheckCommand{}{%
1983   \caption@Info{%
1984     Incompatible package detected (regarding \string\@caption).\MessageBreak
1985     \string\@caption\space=\space meaning\@caption}%
1986   \caption@setbool{incompatible}{1}}%

```

The option compatibility= will override the compatibility mode.

```

1987 \ifundefined{caption@ifcompatibility}%
1988 {\let\caption@ifcompatibility\caption@ifincompatible
1989 \let\@tempa\caption@WarningNoLine}%
1990 {\let\@tempa\@gobble}% suppress warning
1991 \caption@ifcompatibility{%
1992   \@tempa{%
1993     \noexpand\caption will not be redefined since it's already\MessageBreak
1994     redefined by a document class or package which is\MessageBreak
1995     unknown to the caption package}%

```

\ContinuedFloat is not supported in compatibility mode.

```

1996 \renewcommand*\caption@ContinuedFloat[1]{%
1997   \caption@Error{Not available in compatibility mode}}%

```

\caption@start is not supported in compatibility mode.

```

1998 \caption@AtBeginDocument*{%
1999   \let\caption@start\relax
2000   \ifundefined{caption@ORI@capstart}{}{%
2001     \caption@Debug{%

```



```

2002         Restore hypcap definition of \string\capstart\@gobble}%
2003         \let\capstart\caption@ORI@capstart}%
2004         \@ifundefined{caption@ORI@float@makebox}{}{%
2005             \caption@Debug{%
2006                 Restore hyperref redefinition of \string\float@makebox\@gobble}%
2007             \let\float@makebox\caption@ORI@float@makebox}%
2008         }%

\caption@star We define \caption@star here, too, so it's defined but does not make any harm.
2009     \newcommand*\caption@star[2]{#1#2}%

2010 }{%
2011     \caption@ifincompatible{%
2012         \caption@WarningNoLine{%
2013             Forced redefinition of \noexpand\caption since the\MessageBreak
2014             unsupported(!) package option 'compatibility=false'\MessageBreak
2015             was given}%
2016         }{}%

\caption Here comes our definition of \caption and \caption*. Beside the support of the
starred variant this code was adapted to the various packages we support. We are using
\caption@dblarg instead of \@dblarg so \caption{} (with an empty arg.) will
produce a list-of entry, but \caption[]{} won't.
2017     \def\caption{%
2018         \caption@iftype
2019         {\caption@checkgrouplevel\@empty\caption
2020         \caption@star
2021         {\caption@refstepcounter\@capttype}%
2022         {\caption@dblarg{\@caption\@capttype}}}%
2023         {\caption@Error{\noexpand\caption outside float}}}%

\caption@star A helper macro which processes the optional * after \caption.
Note: We set \caption@startrue globally so it works with the sidecap package, too.
2024     \newcommand*\caption@star[2]{%
2025         \@ifstar{\global\caption@startrue#2[]}{#1#2}}%

\@caption As above, our version has been adapted to the packages we support. Addition-
ally our code is nested by \caption@beginex & \caption@end instead of
\begingroup & \endgroup. Furthermore we use \caption@boxrestore in-
stead of \@parboxrestore so this code also works correctly inside list-based environ-
ments like wide & addmargin. (This, and the fact that we use \linewidth instead
of \hsize inside \@makecaption, solves LATEX PR latex/2472.)
2026     \long\def\@caption#1[#2]#3{%
2027         \ifcaption@star \else
2028             \caption@prepareanchor{#1}{#2}%
2029         \fi
2030         \par
2031         \caption@beginex{#1}{#2}{#3}%
2032         \caption@setfloatcapt{%
2033             \caption@boxrestore
2034             \if@minipage

```

```

2035         \@setminipage
2036     \fi
2037     \caption@normalsize
2038     \ifcaption@star
2039         \let\caption@makeanchor\@firstofone
2040     \fi
2041     \@makecaption{\csname fnum@#1\endcsname}%
2042                 {\ignorespaces\caption@makeanchor{#3}}\par
2043     \caption@if@minipage\@minipagetrue\@minipagefalse}%
2044 \caption@end}%

```

`\caption@prepareanchor`

```

2045 \newcommand*\caption@prepareanchor[2]{%
2046     \caption@makecurrent{#1}{#2}%
2047     \caption@ifhycap\caption@@start{}}

2048 }%

2049 \caption@AtBeginDocument*{%
2050     \let\caption@ORI@capstart\@undefined
2051     \let\caption@ORI@float@makebox\@undefined}%

```

`\@xfloat` We redefine `\@xfloat` so inside floating environments our type-specific options will be used, a `hyperref` anchor will be set etc.

```

2052 \let\caption@ORI@xfloat\@xfloat
2053 \def\@xfloat#1[#2]{%
2054     \caption@ORI@xfloat{#1}[#2]%
2055     \caption@settype{#1}}%

2056 }

```

`\@makecaption` `\@makecaption{<label>}{<text>}`

We do basically the same as the original code (from the standard \LaTeX document classes), but take care of the `position=` setting and use `\caption@@make` from the `caption` kernel to finally typeset the caption.

```

2057 \long\def\@makecaption#1#2{%
2058     \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2059     \caption@@make{#1}{#2}%
2060     \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}}

```

2.13 Support for sub-captions

`\caption@DeclareSubType` `\caption@DeclareSub` initializes the usage of `\caption` in sub-floats.

```

2061 \def\caption@DeclareSubType sub#1\@nil{%
2062     \caption@Debug{Initializing subtype for `#1'\@gobble}%
2063     \@namedef{caption@c@#1}{0}%
2064     \@namedef{caption@beginsub#1}{\caption@beginsubfloat{#1}}
2065 \@onlypreamble\caption@DeclareSubType

```

Initialize the sub-captions defined with `\DeclareCaptionSubType...`

```

2066 \caption@For*{subtypelist}{\caption@DeclareSubType sub#1\@nil}

```

Initialize the sub-captions defined with `\newsubfloat`[18]...

```
2067 \caption@AtBeginDocument*{%
2068   \@ifundefined{sf@counterlist}{}{%
2069     \@for\sf@temp:=\sf@counterlist\do{%
2070       \expandafter\caption@DeclareSubType\sf@temp\@nil}}}
```

`\caption@subtypehook` Hook, will be used inside `\caption@setsubtype`.

```
2071 \newcommand*\caption@subtypehook{%
2072   \ifx\caption\caption@subcaption \else

2073     \caption@ifrefstepcounter{}{%
2074       % no \caption or \subcaption in this (floating) environment yet
2075       \caption@Debug{Increment \@c@type\ counter =\the\value\@c@type}%
2076       \caption@l@stepcounter\@c@type
2077       \let\addcontentsline\caption@addsubcontentsline}%

2078     \ifnum\csname caption@c@\@c@type\endcsname=\value\@c@type \else
2079       \caption@Debug{Reset sub\@c@type\ counter}%
2080       \expandafter\xdef\csname caption@c@\@c@type\endcsname{%
2081         \the\value\@c@type}%
2082       \@stpelt\@subcaption
2083     \fi

2084     \c@ContinuedFloat=0\relax
2085     \let\caption@resetContinuedFloat\@gobble
2086     \let\caption@addcontentsline\caption@kernel@addcontentsline
2087     \let\caption@setfloatcapt\@firstofone

2088     \caption@clearmargin
2089     \caption@iflist{}{\let\caption@setlist\@gobble}%
2090     \caption@setoptions{sub}%
2091     \caption@setoptions{subfloat}% for subfig-package compatibility

2092     \let\caption\caption@subcaption
2093   \fi}%
```

`\caption@subcaption` Makes a sub-caption.

```
2094 \newcommand*\caption@subcaption{%
2095   \caption@iftype
2096     {\caption@checkgrouplevel{sub}\subcaption
2097     \caption@star
2098     {\caption@refstepcounter\@subcaption}%
2099     {\caption@dblarg{\@caption\@subcaption}}}%
2100   {\caption@Error{\noexpand\subcaption outside float}}}
```

`\caption@addcontentsline` We extend `\caption@addcontentsline` so it handles sub-captions, too.

Note: `\sf@ifpositiontop` & `\@listsubcaptions` are defined by the subfigure & subfig packages.

```
2101 \let\caption@kernel@addcontentsline\caption@addcontentsline
2102 \renewcommand*\caption@addcontentsline[2]{%
2103   \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
2104   \caption@kernel@addcontentsline{#1}{#2}%
2105   \sf@ifpositiontop{}{\@listsubcaptions{#1}}%
2106   \caption@addsubcontentslines}
```

```

2107 \newcommand*\caption@addsubcontentslines{%
2108   \begingroup
2109     \caption@subcontentslines
2110   \endgroup
2111   \caption@clearsubcontentslines}%

2112 \caption@AtBeginDocument*{%
2113   \@ifundefined{sf@ifpositiontop}{\let\sf@ifpositiontop@gobbletwo}{}%
2114   \caption@clearsubcontentslines
2115   \g@addto@macro\caption@typehook{\caption@checksubcontentslines}%
2116   \AtEndDocument{\caption@checksubcontentslines}}%

```

`\caption@addsubcontentsline` Add a pending sub-caption list entry.

```

2117 \newcommand*\caption@addsubcontentsline[3]{%
2118   \protected@edef\@tempa{%
2119     \noexpand\g@addto@macro\noexpand\caption@subcontentslines{%
2120       \noexpand\@namedef{the#2}{\csname the#2\endcsname}%
2121       \ifx\@currentHref\@undefined \else
2122         \noexpand\def\noexpand\@currentHref{\@currentHref}%
2123       \fi
2124       \protect\addcontentsline{#1}{#2}{#3}}}%
2125   \@tempa}

```

`\caption@checksubcontentslines` Checks if the list of pending sub-captions is empty, if not, a warning will be issued.

```

2126 \newcommand*\caption@checksubcontentslines{%
2127   \ifx\caption@subcontentslines\@empty \else
2128     \caption@Error{%
2129       Something's wrong--perhaps a missing \protect\caption\MessageBreak
2130       in the last figure or table}%
2131     \caption@clearsubcontentslines
2132   \fi}

```

`\caption@clearsubcontentslines` Clear pending sub-caption list entries.

```

2133 \newcommand*\caption@clearsubcontentslines{%
2134   \global\let\caption@subcontentslines\@empty}

```

2.14 Document class & Babel package support

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

```

2135 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}

```

2.14.2 The beamer class

```

2136 \@ifclassloaded{beamer}{%
2137   \caption@Info{beamer document class}%

```

Since the beamer class do not offer a ‘list of figures’ we switch this support in the caption package off.

```

2138   \captionsetup{list=false}
2139   \DeclareCaptionOption{list}{}
2140   \DeclareCaptionOption{listof}{}

```

`\figure` We redefine figure & table so our type-specific options will be used, a hyperref
`\table` anchor will be set etc.

```

2141 \expandafter\let\expandafter\caption@ORI@figure
2142 \csname\string\figure\endcsname
2143 \@namedef{\string\figure} [#1] {%
2144 \caption@ORI@figure [#1] %
2145 \caption@settype{figure}}
2146 \expandafter\let\expandafter\caption@ORI@table
2147 \csname\string\table\endcsname
2148 \@namedef{\string\table} [#1] {%
2149 \caption@ORI@table [#1] %
2150 \caption@settype{table}}
2151 {} {}

```

2.14.3 The KOMA-Script classes

KOMA-Script contains the code `\AtBeginDocument{\let\scr@caption\caption}` so we need to update `\scr@caption` here, too.

```

2152 \@ifundefined{scr@caption} {} {%
2153 \caption@AtBeginDocument{\let\scr@caption\caption}}

```

2.14.4 The frenchb Babel option

Suppress “Package frenchb.1df Warning: The definition of `\@makecaption` has been changed, frenchb will NOT customize it.” (but only if we emulate this customization)

```

2154 \@nameuse{caption@frenchb}\@nameundef{caption@frenchb}

```

2.14.5 The frenchle/pro package

```

2155 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods} {} {%
2156 \caption@Info{frenchle/pro package is loaded}%
2157 \let\captionfont@ORI\captionfont
2158 \let\captionlabelfont@ORI\captionlabelfont
2159 \let\@makecaption@ORI\@makecaption

```

If `\GOfrench` is defined as `\relax` all the re-definitions regarding captions have already been done, so we can do our patches immediately. Otherwise we must add our stuff to `\GOfrench`.

```

2160 \@ifundefined{GOfrench}%
2161 {\let\@tempa\@firstofone}%
2162 {\def\@tempa{\g@addto@macro\GOfrench}}%
2163 \@tempa{%
2164 \let\captionfont\captionfont@ORI
2165 \let\captionfont@ORI\undefined
2166 \let\captionlabelfont\captionlabelfont@ORI
2167 \let\captionlabelfont@ORI\undefined
2168 \let\@makecaption\@makecaption@ORI
2169 \let\@makecaption@ORI\undefined

```

`\@cnORI` We update the definition of `\@cnORI` so it actually reflects our definition of `\caption`.

```

2170 \let\@cnORI\caption

```

```

\@tablescaption The frenchle/pro package sets \caption to \@tablescaption at \begin{table}
for special treatment of footnotes. Therefore we have to patch \@tablescaption so
\caption* will work inside the table environment.

2171 \let\caption@tcORI\@tablescaption
2172 \def\@tablescaption{\caption@star\relax\caption@tcORI}%

\ffrench \ffrench and \tfrench reflect \fnum@figure and \fnum@table when
\tfrench used in French mode. These contain additional code which typesets the caption separator
\captionseparator instead of the usual colon. Because this breaks with our
\makecaption code we have to remove this additional code here.

2173 \let\@eatDP\@undefined
2174 \let\@tempa\@empty
2175 \ifx\ffrench\figure
2176 \l@addto@macro\@tempa{\let\fnum@figure\ffrench}%
2177 \fi
2178 \ifx\tfrench\table
2179 \l@addto@macro\@tempa{\let\fnum@table\tfrench}%
2180 \fi
2181 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
2182 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
2183 \@tempa

2184 }%
2185 }

```

2.15 Package support

```

\caption@ifpackageloaded \caption@ifpackageloaded{<package>}[<version>]{<true>}{<false>}
Some kind of combination of \ifpackageloaded and \ifpackagelater. If
the <package> is not loaded yet, the check will be (re-)done \AtBeginDocument, so
the <package> could be loaded later on, too.

2186 \newcommand\caption@ifpackageloaded[1]{%
2187 \testopt{\caption@ifpackageloaded{#1}}{}%
2188 \@onlypreamble\caption@ifpackageloaded

2189 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
2190 \ifpackageloaded{#1}\@firstofone{%
2191 \caption@Debug{#1 package is not loaded (yet)}\@gobble}%
2192 \caption@AtBeginDocument}{%
2193 \caption@ifpackageloaded{#1}[#2]{#3}{#4}}%
2194 \@onlypreamble\caption@ifpackageloaded

2195 \newcommand\caption@ifpackageloaded[1]{%
2196 \testopt{\caption@ifpackageloaded{#1}}{}%
2197 \@onlypreamble\caption@ifpackageloaded

2198 \long\def\caption@ifpackageloaded#1[#2]{%
2199 \ifpackageloaded{#1}{%
2200 \caption@Info{#1 package is loaded}%
2201 \ifpackagelater{#1}{#2}\@firstoftwo{%
2202 \caption@Error{%
2203 For a successful cooperation we need at least version\MessageBreak
2204 `#2' of package #1,\MessageBreak
2205 but only version\MessageBreak

```

```

2206         '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
2207         is available}%
2208         \@secondoftwo}%
2209     }\@secondoftwo}}
2210 \@onlypreamble\caption@ifpackageloaded

```

`\caption@clearmargin` This macro will be used by some package support stuff where the usual margin setting is not welcome, e.g. in the `sidecap` package.

```

2211 \newcommand*\caption@clearmargin{%
2212     \setcaptionmargin\z@
2213     \let\caption@minmargin\@undefined}

2214 \caption@setbool{needfreeze}{0}
2215 \caption@AtBeginDocument*{%
2216     \caption@ifneedfreeze{%

```

`\caption@freeze` `\caption@freeze*`

Used by the `fltpage` & `sidecap` package support.

```

2217 \newcommand*\caption@freeze{%
2218     \caption@teststar\caption@@freeze\@gobble\@firstofone}%

2219 \newcommand*\caption@@freeze[1]{%
2220     \global\let\caption@SCcontinued\relax
2221     \global\let\caption@SCsetup\@undefined
2222     \global\let\caption@SClentry\@undefined
2223     \global\let\caption@SCtext\@undefined
2224     \global\let\caption@SClabel\@undefined

2225     \let\caption@ORI@ContinuedFloat\ContinuedFloat
2226     \def\ContinuedFloat{%
2227         \caption@withoptargs\caption@SC@ContinuedFloat}%
2228     \def\caption@SC@ContinuedFloat##1{%
2229         \let\caption@ORI@setcounter\setcounter
2230         \let\caption@ORI@addtocounter\addtocounter
2231         \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
2232         \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
2233         \caption@ORI@ContinuedFloat##1%
2234         \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
2235         \let\setcounter\caption@ORI@setcounter
2236         \let\addtocounter\caption@ORI@addtocounter}%
2237     \let\caption@ORI@setup\captionsetup
2238     \def\captionsetup{%
2239         \caption@withoptargs\caption@SC@setup}%
2240     \def\caption@SC@setup##1##2{%
2241         \caption@g@addto@list\caption@SCsetup{##2}%
2242         \caption@ORI@setup##1{##2}}}%
2243     \let\caption@ORI\caption
2244     \def\caption{%
2245         \def\caption{\caption@Error{%
2246             Only one \noexpand\caption can be placed in this environment}}%
2247         \let\captionsetup\caption@setup
2248         \let\caption@@refstepcounter\caption@l@stepcounter
2249         \caption@ORI}%
2250     \long\def\@caption##1[##2]##3{%
2251         \@bsphack

```

```

2252         \gdef\caption@SClentry{##2}%
2253         \gdef\caption@SCTest{##3}%
2254         \@esphack}%
2255     #1{% is \@gobble in star form, and \@firstofone otherwise
2256         \def\label##1{\@bsphack\gdef\caption@SCLabel{##1}\@esphack}}%
2257 }%

\caption@defrost \caption@defrost
2258 \newcommand*\caption@defrost{%
2259     \ifx\caption@ORI@ContinuedFloat\@undefined
2260         \caption@defrost@setup
2261         \ifx\caption@SCTest\@undefined \else
2262             \expandafter\expandafter\expandafter\caption
2263             \expandafter\expandafter\expandafter[%
2264             \expandafter\expandafter\expandafter{%
2265             \expandafter\caption@SClentry\expandafter}\expandafter]%
2266             \expandafter{\caption@SCTest}%
2267         \fi
2268         \ifx\caption@SCLabel\@undefined \else
2269             \expandafter\label\expandafter{\caption@SCLabel}%
2270         \fi
2271     \else
2272         \caption@Error{Internal Error:\MessageBreak
2273             \noexpand\caption@defrost in same group as \string\caption@freeze}%
2274     \fi}%

2275 \newcommand*\caption@defrost@setup{%
2276     \caption@SCcontinued
2277     \ifx\caption@SCsetup\@undefined \else
2278         \expandafter\captionsetup\expandafter{\caption@SCsetup}%
2279     \fi}%

2280 }{}%
2281 \caption@undefbool{needfreeze}}

```

2.15.1 The float package

The float package usually do not use the \LaTeX kernel command `\caption` to typeset the caption but `\float@caption` instead. (`\caption` will only be used if the float is re-styled with `\restylefloat*`.)

The main two things `\float@caption` is doing different are:

- The caption will be typeset inside a `\savebox` called `\@floatcapt` so it can be placed above or below the float contents afterwards.
- `\@makecaption` will not be used to finally typeset the caption. Instead `\@fs@capt` will be used which definition is part of the float style. (Note that `\@fs@capt` will not typeset any vertical space above or below the caption; instead this space will be typeset by the float style code itself.)

```

2282 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
2283     \@ifpackageloaded{floatrow}{%
2284         \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
2285     }%

```


`\@float@setevery` `\@float@setevery{<float type>}` is provided by the float package; it's called every time a floating environment defined with `\newfloat` or `\restylefloat` begins. We use this hook to do some adaptations and to setup the proper caption style (if defined) and additional settings declared with `\captionsetup[<float style>]`.

```
2286 \let\caption@ORI@float@setevery\@float@setevery
2287 \def\@float@setevery#1{%
2288     \float@ifcaption{#1}{%
```

First of all we set the caption position to it's proper value by converting `\@fs@iftopcapt` (which is part of a float style and controls where the caption will be typeset, above or below the float contents) to our `position=` setting. Since the spacing above and below the caption will be done by the float style and *not* by us this sounds quite useless. But in fact it isn't, since some packages based on the caption package (like the subfig package) could have an interest for this information and therefore use the `\caption@iftop` macro we provide in our kernel. Furthermore we need this information for ourself in `\captionof` which uses `\@makecaption` to finally typeset the caption with skips.

```
2289     \caption@setposition{\@fs@iftopcapt t\else b\fi}%
```

Afterward we redefine `\caption@setfloatcapt` (which will be used inside `\@caption`) so the caption will be set inside the box `\@floatcapt`, without extra vertical space.

```
2290     \renewcommand\caption@setfloatcapt{%
2291         \let\@makecaption\caption@make
2292         \global\setbox\@floatcapt\ vbox}%
```

To allow different caption styles for different float styles we also determine the current float style (e.g. 'ruled') and select a caption style (and additional settings) with the same name, if defined.

```
2293     \float@getstyle\float@style{#1}%
2294     \caption@setstyle*\float@style
2295     \caption@setoptions\float@style
2296     }{}%
2297     \caption@freezeHref % will be defrosted in \float@makebox
2298     \caption@ORI@float@setevery{#1}}%
```

`\caption@typehook` L^AT_EX and almost every other packages use `\<type>name` to provide a macro for the type resp. environment name – for example the command `\figurename` will usually contain the name of the floating environment figure:

```
\newcommand\figurename{Figure}
```

But the float package doesn't follow this common naming convention: For floats defined with `\newfloat` it uses `\fname@<type>` instead, which breaks with our code (and with `\autoref` and some other things as well). So we have to map the float package name to the common one here.

Note: If the float was not defined with `\newfloat` but with `\restylefloat` instead, `\fname@<type>` is not defined.

```
2299 \g@addto@macro\caption@typehook{%
2300     \expandafter\ifx\csname #1name\endcsname\relax
2301         \expandafter\let\csname #1name\endcsname\expandafter\endcsname
2302         \csname fname@#1\endcsname
2303     \fi}%
```

`\fs@plaintop` Since the float styles `plaintop` and `boxed` don't use `\abovecaptionskip`
`\fs@boxed` which could be set with `skip=` (`plaintop` uses `\belowcaptionskip` instead of
`\abovecaptionskip`, and `boxed` uses a fixed space of 2pt) we patch the according
float style macros here to change this.

```
2304 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2305 \g@addto@macro\fs@boxed{\def\@fs@mid{\kern\abovecaptionskip\relax}}%
```

`\float@ifstyle` `\float@ifstyle{<type>}{<if-clause>}{<else-clause>}`

Checks if the given `<type>` (e.g. `figure`) is associated with a float style (e.g. `boxed`).

```
2306 \providecommand*\float@ifstyle[1]{%
2307   \expandafter\ifx\csname fst@#1\endcsname\relax
2308     \expandafter\@secondoftwo
2309   \else
2310     \expandafter\@firstoftwo
2311   \fi}%
```

`\float@getstyle` `\float@getstyle{<cmd>}{<type>}`

Determining the float style is not so easy because the only hint provided by the float package is the macro `\fst@<float type>` which points to the macro which represents the float style. So for example after

```
\floatstyle{ruled}
\newfloat{Program}{tbp}{lop}
```

`\fst@Program` will be defined as

```
\def\fst@Program{\fs@ruled} .
```

So here is what we do: We make the first level expansion of `\fst@<float type>` a string so we can gobble the first four tokens (= `\fs@`), so only the the name of the float style is left.

TODO: We need to convert the catcodes here.

```
2312 \providecommand*\float@getstyle[2]{%
2313   \edef#1{%
2314     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2315     \expandafter\expandafter\expandafter\noexpand
2316     \csname fst@#2\endcsname}%
2317   \edef#1{#1}%
2318   \caption@Debug{floatstyle{#2} = '#1'}}%
```

`\float@setstyle` `\float@setstyle{<type>}{<style>}`

Sets or changes the float style associated with `<type>`.

```
2319 \providecommand*\float@setstyle[2]{%
2320   \expandafter\edef\csname fst@#1\endcsname{%
2321     \expandafter\noexpand\csname fs@#2\endcsname}}%
```

`\float@dostyle` `\float@dostyle{<type>}`

```
2322 \providecommand*\float@dostyle[1]{%
2323   \nameuse{fst@#1}\@float@setevery{#1}}%
```

`\float@ifcaption` `\float@ifcaption{<type>}{<if-clause>}{<else-clause>}`

Here we determine if the user has used `\newfloat` resp. `\restylefloat`, or `\restylefloat*`. This is quite easy: If `\@float@c@<captype>` is the same as

\float@caption, the user has used \newfloat or \restylefloat, otherwise we assume he has used \restylefloat*. (This test will fail if some package re-defines \float@caption, so we have to assume that there is no one.)

```

2324 \providecommand*\float@ifcaption[1]{%
2325   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2326   \expandafter\@firstoftwo
2327   \else
2328   \expandafter\@secondoftwo
2329   \fi}%

2330 } }{%
2331 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2332 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2333 % \clearcaptionsetup{boxed}% used by the floatrow package?
2334 }

```

The skip between ‘boxed’ floats and their caption defaults to 2pt.

```
2335 \captionsetup[boxed]{skip=2pt} % do not issue a warning when not used
```

To emulate the ‘ruled’ definition of \@fs@capt we provide a caption style ‘ruled’ with appropriate options. But if the package option ruled was specified, we setup some caption parameters to emulate the behavior of the caption package v1.x option ruled instead, i.e., the current caption settings will be used, but without margin and without ‘single-line-check’.

```

2336 \caption@ifbool{ruled}{%
2337   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
2338 }{%
2339   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2340 }
2341 \caption@undefbool{ruled}

```

2.15.2 The floatflt package

```
2342 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

\floatingfigure We patch \floatingfigure so \caption@floatflt will be used.

```

2343 \let\caption@ORI@floatingfigure\floatingfigure
2344 \def\floatingfigure{%
2345   \caption@floatflt{figure}%
2346   \caption@ORI@floatingfigure}%

```

\floatingtable Same with \floatingtable...

```

2347 \let\caption@ORI@floatingtable\floatingtable
2348 \def\floatingtable{%
2349   \caption@floatflt{table}%
2350 % \caption@setautoposition b%
2351   \caption@ORI@floatingtable}%

```

\caption@floatflt Here we do two things:

1. We use \caption@setoptions{floating<type>} so \captionsetup[floating<type>]{...} is supported.

2. `\linewidth` must be set correctly. Usually this is done by `\@parboxrestore` inside `\@caption`, but since we use `\caption@boxrestore` we have to map this to `\@parboxrestore` instead.

```

2352 \newcommand*\caption@floatflt[1]{%
2353   \caption@settype{#1}%
2354   \caption@clearmargin
2355   \caption@setoptions{floating#1}%
2356   \let\caption@boxrestore\@parboxrestore}%

2357 }{}

```

2.15.3 The fltpage package

```

2358 \caption@ifpackageloaded{fltpage}[1998/10/29 v.0.3]{%
2359   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

\newcommand{\FP@helpNote}[2]{%
  \typeout{FP#1 is inserted on page \pageref{#2}!}}%

```

```

2360 \renewcommand\FP@helpNote[2]{%
2361   \begingroup % save \caption@thepage
2362     \caption@pageref{#2}%
2363     \typeout{FP#1 is inserted on page \caption@thepage!}%
2364   \endgroup}%

```

`\FP@floatBegin` Original code:

```

\newcommand{\FP@floatBegin}[1]{%
  \gdef\@captype{#1}%
  \global\let\FP@savedCaptionCommand\caption%
  \global\let\FP@savedLabelCommand\label%
  \ifthenelse{\equal{\@captype}{figure}}{
    {\global\let\old@Fnum\fnun@figure}%
    {\global\let\old@Fnum\fnun@table}%
  }%
  \let\FP@LabelText\@empty%
  \let\FP@CaptionText\@empty%
  \let\FP@optionalCaptionText\@empty%
  \renewcommand\label[1]{\gdef\FP@LabelText{##1}}%
  \renewcommand\caption[2][ ]{%
    \gdef\FP@optionalCaptionText{##1}\gdef\FP@CaptionText{##2}}%
  \begin{lrbox}{\FP@floatCorpusBOX}%
} %

```

```

2365 \renewcommand*\FP@floatBegin[1]{%
2366   \def\@captype{#1}%
2367   \let\FP@LabelText\@empty
2368   \begin{lrbox}{\FP@floatCorpusBOX}%
2369     \caption@ifFPrefcap
2370       {\caption@freeze\relax}%
2371       {\def\label##1{\@bsphack\gdef\FP@LabelText{##1}\@esphack}%
2372         \caption@freeze*}}%

```

\FP@floatEnd **Original code:**

```

\newcommand{\FP@floatEnd}{%
  \end{lrbox}%
  \global\setbox\FP@floatCorpusBOX=\box\FP@floatCorpusBOX
  \stepcounter{FP@\@capttype C}%
  \FP@savedLabelCommand{\FP@positionLabel}%
  \FP@helpNote{\@capttype}{\FP@positionLabel}%
  \FP@float
    {\FP@positionLabel}% location label test
    {\begin{\@capttype}[p!]}
      \usebox{\FP@floatCorpusBOX}%
      \refstepcounter{\@capttype}%
      \ifthenelse{\equal{\FP@LabelText}{\@empty}}{
        {}{\FP@savedLabelCommand{\expandafter\protect\FP@LabelText}}}%
      \end{\@capttype}}
    {\addtocounter{\@capttype}{-1}}
    {\begin{\@capttype}[b!]}%
      \ifthenelse{\equal{\FP@guid}{\@empty}}{
        {}{\ifthenelse{\equal{\@capttype}{figure}}{
          {\renewcommand{\fnum@figure}{\old@Fnum\ \FP@guid}}}%
          {\renewcommand{\fnum@table}{\old@Fnum\ \FP@guid}}}%
        }%
      \setlength{\abovecaptionskip}{2pt plus2pt minus 1pt} % length above caption
      \setlength{\belowcaptionskip}{2pt plus2pt minus 1pt} % length above caption
      \FP@separatorCaption%
      \ifthenelse{\equal{\FP@optionalCaptionText}{\@empty}}{
        {\FP@savedCaptionCommand{\expandafter\protect\FP@CaptionText}}}%
        {\FP@savedCaptionCommand[\expandafter\protect\FP@optionalCaptionText]}%
        {\expandafter\protect\FP@CaptionText}}%
      \end{\@capttype}}%
    }%

2373 \renewcommand*\FP@floatEnd{%
2374   \end{lrbox}%
2375   \stepcounter{FP@\@capttype C}%
2376   \caption@label\FP@positionLabel
2377   \FP@helpNote\@capttype\FP@positionLabel
2378   \edef\FP@RestoreCounter{%
2379     \noexpand\setcounter{\@capttype}{\the\value\@capttype}%
2380     \noexpand\setcounter{ContinuedFloat}{\the\value{ContinuedFloat}}}%
2381   \FP@float
2382     {\FP@positionLabel}% location label test
2383     {\begin{\@capttype}[p!]}%
2384       \usebox\FP@floatCorpusBOX
2385       \caption@defrost@setup
2386       \caption@ifFPlistcap
2387         {\caption@refstepcounter\@capttype
2388           \expandafter\caption@makecurrent\expandafter\@capttype
2389             \expandafter{\caption@SClentry}}}%
2390         {\expandafter\captionlistentry\expandafter{\caption@SClentry}}}%
2391       \caption@makeanchor\relax
2392       \ifx\FP@LabelText\@empty \else
2393         \expandafter\label\expandafter{\FP@LabelText}%

```

```

2394         \fi
2395         \end\@capttype}%
2396     {\FP@RestoreCounter
2397     \@ifundefined{theH\@capttype}{}{%
2398         \expandafter\l@addto@macro\csname theH\@capttype\endcsname{.FP}}}%
2399     {\begin\@capttype[b!]}%
2400         \let\FP@savedSetfnumCommand\caption@setfnum
2401         \def\caption@setfnum##1{%
2402             \FP@savedSetfnumCommand{##1}%
2403             \ifx\FP@guide\empty \else
2404                 \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}}%
2405         \fi}%
2406         \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2407         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2408         \caption@setoptions{FP\@capttype}%
2409         \FP@separatorCaption
2410         \caption@ifFPlistcap{}{\let\caption@addcontentsline\@gobbletwo}%
2411         \caption@defrost
2412     \end\@capttype}%
2413 }%

2414 \caption@For{typelist}{%
2415     \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2416 }{%
2417     \let\caption@ifFPlistcap\@undefined
2418     \let\caption@ifFPrefcap\@undefined
2419 }

```

2.15.4 The hyperref package

```

2420 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2421     \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2422         \caption@WarningNoLine{%
2423             Hyperref support is turned off\MessageBreak
2424             because hyperref has stopped early}%
2425     }{%
2426         \g@addto@macro\caption@prepareslc{\measuring@true}%

```

`\caption@@refstepcounter` We redefine `\caption@@refstepcounter` so `\H@refstepcounter` will be used instead of `\refstepcounter` inside `\caption` & `\captionlistentry`.

```

2427     \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

`\caption@makecurrent` We redefine `\caption@makecurrent` so a `hyperref` label will be defined inside `\@caption`.

Note: Will be redefined by `\caption@start`.

```

2428     \renewcommand*\caption@makecurrent[2]{%
2429         \caption@makecurrentHref{#1}%
2430         \caption@Debug{hyperref current=\@currentHref}%
2431         \def\@currentlabelname{#2}}}%
2432     \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

`\caption@makeanchor` We redefine `\caption@makeanchor` so a `hyperref` anchor will be set inside `\@caption`.

Note: Will be redefined by `\caption@start`.

```

2433 \renewcommand\caption@makeanchor[1]{%
2434 \caption@Debug{hyperref anchor: \@currentHref}%
2435 % If we cannot have nesting, the anchor is empty.
2436 \ifHy@nesting
2437 \hyper@@anchor{\@currentHref}{#1}%
2438 \else
2439 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2440 \fi}%
2441 \g@addto@macro\caption@prepareslc{\let\caption@makeanchor\@firstofone}%

```

The hypcap option

`\if@capstart` Like the hypcap package we define the switch `\if@capstart`, too.

```
2442 \newif\if@capstart
```

`\caption@start` While the hypcap package defines a macro called `\capstart` our variant is called `\caption@start` and is controlled by the option `hypcap=false/true`.

```
2443 \def\caption@start{\caption@ifhypcap{%
```

Generate the hyperref label and set the hyperref anchor, usually (if `hypcap=false`) both is done inside `\@caption`.

```
2444 \caption@makestart\@captype
2445 \caption@startanchor\@currentHref
```

Prevent `\@caption` from generating a new hyperref label, use the label we save in `\hc@currentHref` instead. (We also support the `@capstart` flag from the hypcap package.)

```

2446 \global\@capstarttrue
2447 \let\hc@currentHref\@currentHref
2448 \def\caption@makecurrentHref##1{%
2449 \global\@capstartfalse
2450 \global\let\@currentHref\hc@currentHref}%

```

Prevent `\@caption` from generating a hyperref anchor since this has already been done.

```

2451 \let\caption@makeanchor\@firstofone
2452 }{}%

```

`\caption@makestart` `\caption@makestart{<type>}` defines a hyperref anchor inside `\caption@start`. Since we offer `\ContinuedFloat` the float counter can change between ‘now’ and `\caption`, i.e., we simply don’t know the figure or table counter yet and therefore we are not able to generate the ‘right’ hyperref label. Two different solutions of this problem came into my mind:

1. I could use the aux file for this purpose.

-or-

2. I set `hypertexnames=false` locally. Furthermore I use `#1.caption.<counter>` (instead of `#1.<counter>`) as naming scheme for `\@currentHref` to avoid conflicts with other hyper links which are generated with `hypertexnames=true`.

The first idea has the advantage that the ‘right’ anchor name will be generated, but one needs an additional \LaTeX run if figures or tables will be inserted or removed.

The second idea has the advantage that it's very easy to implement, but has some side-effects, e.g. the anchor names don't follow the figure or table label names anymore. Since I'm lazy I implemented the second idea, maybe I will revise this later on.

```

2453 \newcommand*\caption@makestart[1]{%
2454 \begingroup
2455 \Hy@hypertextnamesfalse
2456 % \gdef\@currentHlabel{}%
2457 \hyper@makecurrent{#1.caption}%
2458 \endgroup
2459 \caption@Debug{hycap start=\@currentHref}}%

```

`\caption@startanchor` `\caption@startanchor{<Href>}` sets a **hyperref** anchor inside `\caption@start`. This code was taken from the `hycap` package[10] and adapted.

Note: Since `\hyper@@anchor{<Href>}{\relax}` can cause a change from vertical mode to horizontal mode (design flaw in `hyperref` package!?), and since the workaround `\let\leavevmode\relax` which can be found in the `hycap` package is not always sufficient (for example with “Direct pdfmark support” and `breaklinks=true`), we use `\caption@anchor` instead of `\hyper@@anchor` here.

```

2460 \newcommand*\caption@startanchor[1]{%
2461 \ifvmode\begingroup
2462 \caption@Debug{hycap anchor: #1 (vertical mode)}%
2463 \@tempdima\prevdepth
2464 \nointerlineskip
2465 \vspace*{-\caption@hycap@space}%
2466 \caption@anchor{#1}%
2467 \vspace*{\caption@hycap@space}%
2468 \prevdepth\@tempdima
2469 \endgroup\else
2470 \caption@Debug{hycap anchor: #1 (horizontal mode)}%
2471 \caption@anchor{#1}%
2472 \fi}%

```

`\caption@anchor` `\caption@anchor{<Href>}` sets a **hyperref** anchor.

```

2473 \newcommand*\caption@anchor[1]{%
2474 \ifmeasuring@ \else
2475 \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2476 \fi}%

```

Note: Since `\Hy@raisedlink` change `\@tempdima` we surrounded it by `\ifvmode`, suppressing “LaTeX Warning: Float too large for page by 1.0pt” in sideways floats. (This is not necessary since `hyperref` v6.77.)

```

2477 \ifx\HyperRaiseLinkLength\@tempdima
2478 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2479 \else
2480 \let\caption@raisedlink\Hy@raisedlink
2481 \fi

```

`\caption@@start` Will be used by `\caption@freezeHref`. Apart from that we issue a warning if we expect a saved **hyperref** label coming from `\caption@start`, but there isn't any.

```

2482 \def\caption@@start{%
2483 \@ifundefined{hc@currentHref}{%
2484 \caption@Warning{%
2485 The option 'hycap=true' will be ignored for this\MessageBreak

```



```

2486           particular \string\caption}}{}%

\caption@freezeHref Suppress \caption@start from generating a hyperref label and setting a hyper-
                    ref anchor. Instead if \@caption generates a hyperref label, it will be stored in
                    \caption@currentHref. Furthermore we need to redefine \caption@setfloatcapt
                    so no hyperref anchor will be placed in \@caption.
2487     \def\caption@freezeHref{%
2488         \let\caption@ORI@start\caption@start
2489         \def\caption@start{\let\caption@start\caption@ORI@start}%

2490 %         \let\caption@ORI@@start\caption@@start
2491 %         \l@addto@macro\caption@subtypehook{%
2492 %             \let\caption@@start\caption@ORI@@start}%

2493     \global\let\caption@currentHref\@undefined
2494     \def\caption@@start{\global\let\caption@currentHref\@currentHref}%

2495     \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2496     \renewcommand*\caption@setfloatcapt{%
2497         \ifx\caption@currentHref\@undefined \else
2498             \let\caption@makeanchor\@firstofone
2499         \fi
2500     \caption@ORI@setfloatcapt}%

\caption@defrostHref If there is a frozen \@currentHref, we set the hyperref anchor here.
2501     \def\caption@defrostHref{%
2502         \ifx\caption@currentHref\@undefined \else
2503             \caption@startanchor\caption@currentHref
2504             \global\let\caption@currentHref\@undefined
2505         \fi}%

\float@makebox Do our own redefinition of \float@makebox, if it was redefined by the hyperref pack-
                age.
2506     \@ifundefined{HyOrg@float@makebox}{}{%
2507         \caption@Debug{%
2508             Redefining \noexpand\float@makebox (again)\@gobble}%
2509         \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
2510         \renewcommand\float@makebox[1]{%
2511             \HyOrg@float@makebox{#1\relax \caption@defrostHref}}%
2512     }%

2513 }}{}

```

2.15.5 The hypcap package

```

2514 \caption@IfPackageLoaded{hypcap}{% v1.0
2515     \ifx\caption@start\relax \else % hyperref hasn't stopped early

\capstart We do this simply by mapping \capstart to \caption@start, although our code
            does not behave exactly like the original one: The original \capstart has an effect on
            the next \caption only but our version affects all \captions in the same environ-
            ment, at least unless a new \capstart will be placed.
2516     \let\caption@ORI@capstart\capstart % save for compatibility mode

```

```

2517 \let\capstart\caption@start
2518 \let\caption@start\relax
2519 \let\caption@@start\relax

```

`\caption@hyppcapspace` Furthermore we map our `\caption@hyppcapspace` to `\hyppcapspace` offered by the `hyppcap` package.

```

2520 \caption@set@bool\caption@ifhyppcap1%
2521 \renewcommand*\caption@hyppcapspace{\hyppcapspace}%

2522 \fi}{ }

```

2.15.6 The listings package

```

2523 \caption@ifPackageLoaded{listings}[2004/02/13 v1.2]{%

```

`\lst@MakeCaption` To support the `listings` package we need to redefine `\lst@MakeCaption` so the original stuff is nested with `\caption@begin` and `\caption@end` etc.

Note: This macro is always called twice (with ‘t’ resp. ‘b’ as parameter), therefore we need an extra group here.

```

2524 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2525 \def\lst@MakeCaption#1{% #1 is ‘t’ or ‘b’
2526 \begingroup

```

First of all, we set `position=#1` and if it was set to ‘top’, we swap the skips so the default behavior of the `listings` package will not be changed. (Note that the `listings` package has set its own `\abovecaptionskip` & `\belowcaptionskip` values prior to calling `\lst@MakeCaption`.)

```

2527 \caption@setposition{#1}%
2528 \caption@iftop{%
2529 \@tempdima\belowcaptionskip
2530 \belowcaptionskip\abovecaptionskip
2531 \abovecaptionskip\@tempdima}{ }%

```

Afterwards we set the local ‘`lstlisting`’ options.

```

2532 \caption@setoptions{lstlisting}%

```

If the `position=` is now set to `auto`, we take over the `captionpos=` setting from the `listings` package.

```

2533 \caption@setautoposition{#1}%

```

At the end we do similar stuff as in our `\@caption` code.

```

2534 \caption@begin{lstlisting}%
2535 \caption@ORI@lst@MakeCaption{#1}%
2536 \caption@end
2537 \endgroup}%

```

`\lst@makecaption` Wrapper macros for typesetting the `caption=` resp. `title=` value.

```

\lst@maketitle 2538 \def\lst@makecaption{\caption@starfalse\@makecaption}%
2539 \def\lst@maketitle{\caption@starttrue\@makecaption\@empty}%

```

`\ext@lstlisting` Since the `listings` package do not define `\ext@lstlisting`, but we needed it when `\captionof{lstlisting}` will be done by the end user, we define it here.

```

2540 \providecommand*\ext@lstlisting{lol}%

2541 }{ }

```

2.15.7 The longtable package

```

\LTcapttype \LTcapttype is preset to table.
2542 \providecommand*\LTcapttype{table}

2543 \caption@ifPackageLoaded{longtable}[1995/05/24 v3.14]{%
2544   \RequirePackage{ltcaption}[2007/09/01]%
2545   \let\LT@makecaption\@undefined

\LT@array We redefine \LT@array here to get \captionsetup{<options>} working inside
longtables.
Note: Since the hyperref package patches \LT@array as well and since this only works
with the original definition of \LT@array, we have to do this after the hyperref package,
i.e. \AtBeginDocument.
2546 \caption@AtBeginDocument{%
2547   \let\caption@ORI@LT@array\LT@array
2548   \renewcommand*\LT@array{%

\captionsetup for longtable:
2549   \global\let\caption@opt@@longtable\@undefined
2550   \def\captionsetup{%
2551     \noalign\bgroup
2552     \ifstar\captionsetup\captionsetup}% gobble *
2553   \def\captionsetup##1{\LT@captionsetup{##1}\egroup}%
2554   \def\LT@captionsetup##1{%
2555     \captionsetup@starttrue\caption@setup@options[@longtable]{##1}%
2556     \global\let\caption@opt@@longtable\caption@opt@@longtable}%

\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2557   \def\captionabovetrue{\LT@captionsetup{position=t}}%
2558   \def\captionabovefalse{\LT@captionsetup{position=b}}%

\captionlistentry for longtable:
2559   \def\captionlistentry{%
2560     \noalign\bgroup
2561     \ifstar\egroup\LT@captionlistentry}% gobble *
2562     {\egroup\LT@captionlistentry}}%
2563   \def\LT@captionlistentry##1{%
2564     \caption@listentry\@firstoftwo[\LTcapttype]{##1}}%

\ContinuedFloat for longtable:
(Commented out, since it's not deeply tested and quite useless anyway)
Note: hyperref versions < v6.76j uses 2x \hyper@makecurrent
2565 % \caption@ifhyppcap{%
2566 %   \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2567 %   \def\hyper@makecurrent##1{%
2568 %     \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2569 %     \caption@makestart{##1}%
2570 %   }
2571 %   \let\Hy@LT@currentHlabel\@currentHlabel
2572 %   \let\Hy@LT@currentHref\@currentHref
2573 %   \def\hyper@makecurrent###1{%
2574 %     \let\@currentHlabel\Hy@LT@currentHlabel
2575 %     \let\@currentHref\Hy@LT@currentHref}}%
2576 %   \let\caption@ORI@ContinuedFloat\ContinuedFloat

```

```

2576 %      \def\ContinuedFloat{\noalign{%
2577 %          \gdef\caption@setContinuedFloat{%
2578 %              \let\caption@resetContinuedFloat\@gobble}%
2579 %          \def\caption@setoptions###1{%
2580 %              \g@addto@macro\caption@setContinuedFloat{%
2581 %                  \caption@setoptions{###1}}}%
2582 %          \let\@capttype\LTcapttype
2583 %          \caption@ORI@ContinuedFloat}}%
2584 %      }{%
2585 %          \def\ContinuedFloat{\noalign{%
2586 %              \caption@Error{%
2587 %                  \noexpand\ContinuedFloat inside longtables\MessageBreak
2588 %                  is only available with 'hyccap=true'}}}%
2589 %      }%
2590 %      \global\let\caption@setContinuedFloat\@empty
2591 %      \def\ContinuedFloat{\noalign{%
2592 %          \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2593 %      \caption@ORI@LT@array}}%

```

`\LT@c@ption` The original implementation:

```

\def\LT@c@ption#1[#2]#3{%
  \LT@makecaption#1\fnun@table{#3}%
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else
    {\let\space
     \addcontentsline{lot}{table}{\protect\numberline{\thetable}{#2}}}%
  \fi}

```

Our implementation uses `\LTcapttype` instead of `{table}`:

```

2594 \long\def\LT@c@ption#1[#2]#3{%
2595     \LT@makecaption#1{\csname fnun@\LTcapttype\endcsname}{#3}%
2596     \LT@captionlistentry{#2}}%

```

`\LT@makecaption` `\LT@makecaption{<cmd>}{<label>}{<text>}`

The original definition:

```

\def\LT@makecaption#1#2#3{%
  \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\LTcapwidth{%
    % Based on article class "\@makecaption", "#1" is "\@gobble" in star
    % form, and "\@firstofone" otherwise.
    \sbox\@tempboxa{#1{#2: }#3}%
    \ifdim\wd\@tempboxa>\hsize
      #1{#2: }#3%
    \else
      \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
    \fi
    \endgraf\vskip\baselineskip}%
  \hss}}

```

Our definition:

```

2597 \renewcommand\LT@makecaption[3]{%
2598     \caption@LT@make{%

```

If `\LTcapwidth` is not set to its default value 4in we assume that it shall overwrite our own setting. (But `\captionsetup[longtable]{width=...}` will overwrite `\LTcapwidth`.)

```
2599      \caption@settype*\LTcaptype
2600      \ifdim\LTcapwidth=4in \else
2601        \setcaptionwidth\LTcapwidth
2602      \fi
2603      \caption@setoptions{longtable}%
2604 %      \caption@setContinuedFloat
2605      \caption@setoptions{@longtable}%
```

`position=auto` is a bad idea for longtables, but we do our very best. This works quite well for captions inside the longtable contents, but not for captions inside the longtable (end)foot.

Note: This should be ‘top’ if unclear!

```
2606      \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
```

We set `\ifcaption@star` according the 1st argument.

```
2607      \caption@startrue#1\caption@starfalse
2608      \caption@resetContinuedFloat\LTcaptype
2609      \caption@begin\LTcaptype
2610      \caption@normalsize
```

The following skip has the purpose to correct the height of the `\parbox[t]`. Usually it’s the height of the very first line, but because of our extra skips (`\abovecaptionskip` and `\belowcaptionskip`) it’s always 0pt.

(A different idea would be typesetting the first skip outside the longtable column with `\noalign{\vskip...}`, but this means we have to move `\caption@begin` to some other place because it does not work in tabular mode. And at the moment I have no idea on how to do this in an elegant way...)

```
2611      \vskip-\ht\strutbox
```

The following code should look familiar. We do our skips and use `\caption@@make` to typeset the caption itself.

```
2612      \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2613      \caption@@make{#2}{#3}\endgraf
2614      \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2615      \caption@end}%
```

```
2616 {} {}
```

2.15.8 The picinpar package

```
2617 \caption@ifPackageLoaded{picinpar}{%
```

`\figwindow` The `picinpar` package comes with its own caption code (`\wincaption`, `\@wincaption`,
`\tabwindow` `\@makewincaption`, ...) so we redefine `\figwindow` & `\tabwindow` to use
`\caption` instead.

```
2618 \long\def\figwindow[#1,#2,#3,#4] {%
2619   \caption@window{figure}%
2620   \caption@setoptions{figwindow}%
2621   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
```

```

2622 \long\def\tabwindow[#1,#2,#3,#4] {%
2623   \caption@window{table}%
2624   \caption@setoptions{tabwindow}%
2625   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%

```

`\caption@window` Beside calling `\caption@settype` we redefine `\caption@boxrestore` (as in `floatflt` & `picins` package support) and `\@makecaption` (as in `float` package support) here.

```

2626 \newcommand*\caption@window[1]{%
2627   \let\caption@boxrestore\@parboxrestore
2628   \let\@makecaption\caption@make
2629   \caption@setautoposition b%
2630   \caption@settype{#1}%
2631   \caption@clearmargin}%

```

`\caption@wincaption` This one finally typesets the caption using `\caption`.

```

2632 \newcommand\caption@wincaption[1]{%

```

This will be done twice for every `figwindow` & `tabwindow` caption – on the first run `\picwd` is `0pt`, on the second run `\picwd` is `\hsize`.

```

2633   \ifdim\picwd=\z@
2634     \let\caption@makecurrent\@gobbletwo
2635     \let\caption@start\relax
2636     \caption@prepareslc
2637   \fi

```

The argument `#1` could contain simply the caption text (e.g. A figure caption), but it could also contain an optional argument, the *⟨lst.entry⟩* (e.g. [An entry to the L^{OF}] {A figure caption}). Therefore we have to test if `#1` begins with `[` or not; furthermore we support a starred variant – as in `\caption*` – so we test for `*`, too.

```

2638   \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2639   \if\@tempa*%
2640     \let\@tempa\@firstofone
2641   \else\if\@tempa[%
2642     \let\@tempa\@firstofone
2643   \else
2644     \let\@tempa\@empty
2645   \fi\fi
2646   \expandafter\caption\@tempa{#1}}%

```

```

2647 }{}

```

2.15.9 The `picins` package

`\piccaptiontype` `\piccaptiontype{⟨type⟩}`

We offer this macro for changing the *⟨type⟩* of the caption, so the user doesn't have to redefine `\@capttype`, as proposed in the `picins` documentation.

Note: We define this macro here so it can be used in the preamble of the document, even when the caption package was loaded prior to the `picins` package.

```

2648 \newcommand*\piccaptiontype[1]{\def\@piccaptiontype{#1}}
2649 \caption@ifpackageloaded{picins}{%

```

Initial set \@piccaptiontype and undefine \@capttype which was set to figure by the picins package.

```
2650 \@ifundefined{@piccaptiontype}{%
2651   \caption@iftype{%
2652     \let\@piccaptiontype\@capttype
2653   }{%
2654     \def\@piccaptiontype{figure}%
2655   }%
2656 }{}%
2657 \let\@capttype\@undefined
```

\piccaption The original code:

```
\def\piccaption{\@ifnextchar [{\@piccaption}\@piccaption[]}]}
```

Our code uses \caption@star so \piccaption* works, and \caption@dblarg so \piccaption{} works correctly.

```
2658 \def\piccaption{\caption@star\relax{\caption@dblarg\@piccaption}}%
```

\make@piccaption The original code:

```
\def\make@piccaption{%
[... ]
\setbox\@TEXT=\vbox{\hsize\hsiz@\caption[\sh@rtf@rm]{\capti@nt@xt}}%
}
```

In our code we have to correct several things:

1. \@capttype must be defined, since we have removed the global definition.
2. We use \caption@setoptions{parpic} so \captionsetup[parpic]{...} is supported.
3. \linewidth must be set correctly. Usually this is done by \@parboxrestore inside \caption, but since we use \@caption@boxrestore we have to map this to \@parboxrestore instead.
4. The two arguments of \caption (\sh@rtf@rm & \capti@nt@xt) should be expanded on first level so \caption[] {...} and \caption[...]{ } work correctly.

```
2659 \let\caption@ORI@make@piccaption\make@piccaption
2660 \def\make@piccaption{%
2661   \let\caption@ORI\caption
2662   \long\def\caption[##1]##2{%
2663     \caption@freezeHref % will be defrosted in \ivparpic
2664     \caption@settype\@piccaptiontype
2665 % \ifnum\c@piccaptionpos>2\relax
2666     \caption@clearmargin
2667 % \else
2668 % \captionwidth\z@ % do not use "width=" setting
2669 % \fi
2670     \caption@setoptions{parpic}%
2671     \let\caption@boxrestore\@parboxrestore
2672     \caption@setautoposition b%
```

```

2673     \expandafter\expandafter\expandafter\caption@ORI
2674     \expandafter\expandafter\expandafter[%
2675     \expandafter\expandafter\expandafter{%
2676     \expandafter##1\expandafter}\expandafter]\expandafter{##2}}%

-or- \begingroup
      \toks0\expandafter{##1} \toks2\expandafter{##2}
      \edef\x{\endgroup
        \noexpand\caption@ORI[{\the\toks0}]{\the\toks2}}
      \x

-or- \edef\x{%
      \noexpand\caption@ORI[{\unexpanded\expandafter{##1}}]%
                                {\unexpanded\expandafter{##2}}}%
      \x

2677     \caption@ORI@make@piccaption
2678     \let\caption\caption@ORI}%

```

`\ivparpic` **We need to set our hyperref anchor here. Not bullet-proof since we have to redefine `\noindent` here!**

```

2679     \let\caption@ORI@ivparpic\ivparpic
2680     \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
2681       \let\caption@ORI@noindent\noindent
2682       \def\noindent{%
2683         \caption@defrostHref
2684         \let\noindent\caption@ORI@noindent
2685         \noindent}%
2686       \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2687       \let\noindent\caption@ORI@noindent}%

2688 }{%
2689 \let\piccaptiontype\@undefined
2690 }

```

2.15.10 The rotating package

```

2691 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%

```

`\rotcaption` **Make `\rotcaption*` work.**

```

2692 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
2693 % \let\@rotcaption\@undefined

```

`\rotcaptionof` **Make `\rotcaptionof(*)` work.**

```

2694 \def\rotcaptionof{%
2695   \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

```

`\@makerotcaption` **Original (bugfixed) code:**

```

\long\def\@makerotcaption#1#2{%
  \setbox\@tempboxa\hbox{#1: #2}%
  \ifdim \wd\@tempboxa > .8\vsiz
    \rotatebox{90}{%
      \begin{minipage}{.8\textheight}#1: #2\end{minipage}%
    }% \par % <== \par removed (AR)
  \else%
    \rotatebox{90}{\box\@tempboxa}%
  \fi
}

```



```

\nobreak\hspace{12pt}% <== \nobreak added (AR)
}

```

Our version emulates this behavior, but if `width=` is set, the rotated caption is always typeset as `minipage`. (Note that `margin=` is not supported here.)

```

2696 \long\def\@makerotcaption#1#2{%
2697     \ifdim\captionwidth=\z@
2698         \setcaptionwidth{.8\textheight}%
2699         \caption@slc{#1}{#2}{.8\vsizer}%
2700         \let\caption@makerot\caption@@make
2701         \caption@clearmargin
2702 %     \long\def\caption@parbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2703 %     (not needed because \rotatebox uses an \hbox anyway)
2704     \let\caption@parbox\@secondoftwo}%
2705     \caption@set@bool\caption@ifslc0% been there, done that
2706 \fi
2707 \rotatebox{90}{\caption@makerot{#1}{#2}}%
2708 \nobreak\hspace{12pt}}%
2709 \newcommand\caption@makerot[2]{%
2710     \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
2711 \caption@For{typelist}{%
2712     \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
2713     \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
2714 {}{}

```

2.15.11 The sidecap package

```

2715 \caption@ifpackageloaded{sidecap}[1999/05/11 v1.4d]{%
2716     \caption@setbool{needfreeze}{1}%

```

`\SC@caption` First of all, we let `sidecap` use a current definition of `\caption`.
(This is only required for version 1.5d of the `sidecap` package.)

```

2717 \caption@AtBeginDocument{\let\SC@caption=\caption}%

```

`\SC@zfloat` This macro will be called at the start of the environment, here is a good opportunity to do some adaptations to `\caption` and `\captionsetup`.

```

2718 \let\caption@ORI@SC@zfloat\SC@zfloat
2719 \def\SC@zfloat#1#2#3[#4]{%

```

First we use the original definition, but save & restore `\caption` so `\caption@freeze` will work correctly.

```

2720     \let\caption@ORI\caption
2721     \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
2722     \let\caption\caption@ORI

```

Since the `sidecap` package uses our `\caption` code outside the environment the regular `\captionsetup` will not work. So we need a special version here which saves the given argument list which will be executed later on. Furthermore we need to make `\caption*` work.

```

2723     \caption@settype*{#2}%
2724     \caption@freeze*}%

```

`\endSC@FLOAT` This macro will be called at the end of the environment, here we need to setup our stuff before the `sidecap` package actually typesets its caption.

```
2725 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
2726 \def\endSC@FLOAT{%
```

Note: `\@capttype` isn't defined here, this will be done inside the original definition of `\endSC@FLOAT`. But `\SC@capttype` is defined and can be used here, if needed.

```
2727 \let\caption@ORI@settype\caption@settype
2728 \def\caption@settype##1{% will be done in \@xfloat
2729 \caption@ORI@settype*{##1}% do not change \@currentlabel
2730 \caption@setSC@justify
2731 %%% \caption@setoptions{SCfloat}%
2732 \caption@setoptions{SC\@capttype}%
2733 \caption@start}%
```

Before we can typeset the caption we need to set the margin to zero because any extra margin would only be disturbing here.

(We don't need to take care about the caption position because the `sidecap` package set both `\abovecaptionskip` and `\belowcaptionskip` to a skip of zero anyway.)

Furthermore `\SC@justify` will override the caption justification, if set. The usage of `\SC@justify` differs from version to version of the `sidecap` package:

Version 1.4: `\SC@justify` is not defined

Version 1.5: `\SC@justify` is `\relax` when not set

Version 1.6: `\SC@justify` is `\@empty` when not set

```
2734 \def\caption@setSC@justify{%
2735 \caption@clearmargin
2736 \ifundefined{SC@justify}{}{%
2737 \ifx\SC@justify\@empty \else
2738 \let\caption@hj\SC@justify
2739 \let\SC@justify\@empty
2740 \fi}}%
```

Make the original definition of `\endSC@FLOAT` to use our caption stuff instead of its own.

Note: At this point the `sidecap` definition of `\caption` is valid, not the regular one!

```
2741 \let\caption\SC@orig@caption
2742 \def\SC@orig@caption[##1]##2{\caption@defrost}%
```

Finally we call the original definition of `\endSC@FLOAT`.

```
2743 \caption@setSC@justify % for compatibility mode
2744 \caption@ORI@endSC@FLOAT}%

2745 \newcommand*\caption@For@SC[2]{%
2746 \def#1{b}% = \sidecaptionvpos{#2}{b} (v1.6)
2747 \newenvironment{SC#2}%
2748 {\SC@float[##1]{##2}}{\endSC@float}%
2749 \newenvironment{SC#2*}%
2750 {\SC@dblfloat[##1]{##2}}{\endSC@dblfloat}}%
2751 \@onlypreamble\caption@For@SC
2752 \caption@For{typelist}{%
2753 \expandafter\caption@For@SC\csname SC#1@vpos\endcsname{##1}}%
2754 }{ }
```

2.15.12 The subfigure package

```

2755 \caption@ifPackageLoaded{subfigure}[2002/01/23 v2.1]{%
\sif@ifpositiontop If the subfigure package is loaded, we map \sf@ifpositiontop to \iffiguretopcap
resp. \iftabletopcap, so the subfigure v2.1 options figbotcap etc. will still work.
2756 \def\sif@ifpositiontop{%
2757 \ifx\@capytype\@undefined
2758 \expandafter\@gobbletwo
2759 \else\ifx\@capytype\relax
2760 \expandafter\expandafter\expandafter\@gobbletwo
2761 \else
2762 \expandafter\expandafter\expandafter\sif@if@position@top
2763 \fi\fi}
2764 \def\sif@if@position@top{%
2765 \ifundefined{if\@capytype topcap}%
2766 {\@gobbletwo}%
2767 {\@nameuse{if\@capytype topcap}%
2768 \expandafter\@firstoftwo
2769 \else
2770 \expandafter\@secondoftwo
2771 \fi}}
2772 {}{}

```

2.15.13 The supertabular package

```

2773 \caption@ifPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.
2774 \let\caption@ORI@tablecaption\tablecaption
2775 \def\tablecaption{\caption@star\relax\caption@ORI@tablecaption}%

\ST@caption The original code:

\long\def\ST@caption#1[#2]#3{\par%
\addcontentsline{\csname ext@#1\endcsname}{#1}%
{\protect\numberline{%
\csname the#1\endcsname}{\ignorespaces #2}}
\begingroup
\@parboxrestore
\normalsize
\if@topcaption \vskip -10\p@ \fi
\@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
\if@topcaption \vskip 10\p@ \fi
\endgroup}

2776 \long\def\ST@caption#1[#2]#3{\par%
2777 \caption@settype*{#1}%
2778 \caption@setoptions{supertabular}%

```

The position= setting will be overwritten by the supertabular package: If \topcaption was used, the position will be top automatically, bottom otherwise.

```

2779 \def\caption@fixposition{%
2780 \caption@setposition{\if@topcaption t\else b\fi}}%

```

```

2781 \caption@beginex{#1}{#2}{#3}%
2782 \caption@boxrestore
2783 \caption@normalsize
2784 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2785 \caption@end}%

2786 }{%
2787 \let\caption@setSTposition\@undefined
2788 }

```

2.15.14 The threeparttable package

```

2789 \caption@ifPackageLoaded{threeparttable}[2003/06/13 v3.0]{%
\threeparttable Unfortunately \@capytype is not set when \TPT@common will be used, so we have to
redefine \threeparttable and \measuredfigure instead.
2790 \let\caption@ORI@threeparttable\threeparttable
2791 \renewcommand*\threeparttable{%
2792 \caption@settype{table}%
2793 \caption@setposition a% ?
2794 \caption@clearmargin
2795 \caption@setoptions{threeparttable}%
2796 \caption@ORI@threeparttable}%

\measuredfigure Same here...
2797 \let\caption@ORI@measuredfigure\measuredfigure
2798 \renewcommand*\measuredfigure{%
2799 \caption@settype{figure}%
2800 \caption@setposition a% ?
2801 \caption@clearmargin
2802 \caption@setoptions{measuredfigure}%
2803 \caption@ORI@measuredfigure}%

\TPT@caption The original code:
\def\TPT@caption#1[#2]#3{\gdef\TPT@docapt
{\par\global\let\TPT@docapt\@undefined \TPT@LA@caption{#1}[{#2}]}%
{\strut\ignorespaces#3\ifhmode\unskip\@finalstrut\strutbox\fi}}%
\ifx\TPT@hsize\@empty \let\label\TPT@gatherlabel \abovecaptionskip\z@skip
\else \TPT@docapt \fi \ignorespaces}

2804 \def\TPT@caption#1[#2]#3{%
2805 \gdef\TPT@docapt{%
2806 \global\let\TPT@docapt\@undefined
2807 \caption@setautoposition\caption@TPT@position
2808 \TPT@LA@caption{#1}[{#2}]{#3}}%
2809 \ifx\TPT@hsize\@empty
2810 \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
2811 \gdef\caption@TPT@position{t}%
2812 \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2813 \else
2814 \def\caption@TPT@position{b}%
2815 \TPT@docapt
2816 \fi
2817 \ignorespaces}%

```

```

2818 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2819 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima}%

2820 }{}

```

2.15.15 The wrapfig package

```

2821 \caption@ifpackageloaded{wrapfig}{% ver 3.3 (Oct 12, 1999)

\float@ifstyle \float@ifstyle{<type>}{<if-clause>}{<else-clause>}
(see float package support for details)

```

```

2822 \providecommand*\float@ifstyle[1]{%
2823   \expandafter\ifx\csname fst@#1\endcsname\relax
2824     \expandafter\@secondoftwo
2825   \else
2826     \expandafter\@firstoftwo
2827   \fi}%

```

`\caption@restylewrapfloat` This one redefines the `wrap#1` environment, e.g. `wrapfigure`. Our code uses `\caption@setoptions{wrapfigure}` so `\captionsetup[wrapfigure]{...}` will work.

But first we check if our redefinition was already done, this could happen inside `\float@restyle` when the `wrapfig` support of the `float` package was not installed successfully, so it has not redefined `\wrap#1` there.

```

2828 \newcommand*\caption@restylewrapfloat[1]{%
2829   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2830     \csname wrap#1\endcsname
2831   \caption@Error{%
2832     For a successful cooperation of the 'wrapfig' package\MessageBreak
2833     with the 'float' package you should load the 'wrapfig'\MessageBreak
2834     package *after* the 'float' package}%
2835   \else
2836     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2837       \csname wrap#1\endcsname
2838     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
2839     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2840       \csname wrap#1\endcsname
2841   \fi}%

```

```

\caption@wrapfloat

2842 \newcommand*\caption@wrapfloat[1]{%
2843   \caption@settype*{#1}%
2844   \float@ifstyle{#1}{%
2845     \ifx\WF@floatstyhook\@undefined
2846       \caption@Error{%
2847         For a successful cooperation of the 'wrapfig' package\MessageBreak
2848         with the 'float' package you should use at least\MessageBreak
2849         'wrapfig' version 3.6}%
2850     \else
2851       \float@dostyle{#1}%
2852     \fi}%
2853   \caption@clearmargin
2854   %%% \caption@setoptions{wrapfloat}%
2855   \caption@setoptions{wrap#1}%

```

```
2856 \@nameuse{caption@ORI@wrap#1}}%
```

Now we redefine the `wrapfig` environments we know about.

If someone has placed a `\newfloat` right between `\usepackage{wrapfig}` and `\usepackage{caption}` (or loads the `caption` package first, so all these patches will be done with `\AtBeginDocument`) we have bad luck since the `float` package do not offer a list of (re)styled floats. (This would finally lead to an error in `\caption@setfloatcapt.`)

```
2857 \caption@restylewrapfloat{figure}%
2858 \caption@restylewrapfloat{table}%

2859 \caption@For{typelist}{%
2860   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
2861   \caption@restylewrapfloat{#1}}%

2862 \ifx\WF@floatstyhook\undefined \else % wrapfig v3.6
```

`\float@restyle` If the `wrapfig` package v3.6 is used, we patch `\float@restyle` (if defined), too, so new or restyled floats will be handled correctly, too.

```
2863 \ifundefined{float@restyle}{}{%
2864   \toks@=\expandafter{\float@restyle{#1}}% (env may or may not be defined)
2865   \caption@restylewrapfloat{#1}}%
2866   \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2867   \@tempa}% perform redefinitions
```

`\wrapfloat` An additional check of the package load order: If both, neither the `wrapfig` package nor the `caption` package haven't catch `\float@restyle`, we finally splash down at `\wrapfloat`.

```
2868 \let\caption@ORI@wrapfloat\wrapfloat
2869 \def\wrapfloat#1{%
2870   \float@ifstyle{#1}{%
2871     \caption@Error{%
2872       For a successful cooperation of the 'wrapfig' package\MessageBreak
2873       with the 'float' package you should load the 'wrapfig'\MessageBreak
2874       package *right after* the 'float' package}}{%
2875     \caption@ORI@wrapfloat{#1}}%

2876 \fi % wrapfig v3.6
```

`\WF@rapt` We place our `hyperref` anchor here.
Original code:

```
\def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
\gdef\WF@ovh{#1}% hold overhang for later, when \width is known
\global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
\ifdim\hsize>\z@ \@parboxrestore \else
\setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
\ignorespaces \fi}
```

Our code:

```
2877 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2878   \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2879   \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
2880   \caption@start
```

```
2881 \ifdim\hsize>\z@ \@parboxrestore \else
2882 \setbox\z@\hbox\bgroup \let\wf@@caption\caption \let\caption\wf@caption
2883 \ignorespaces \fi}%
2884 }{ }
```

References

- [1] Frank Mittelbach and Michel Goossens:
The L^AT_EX Companion (2nd. Ed.),
Addison-Wesley, 2004.
- [2] Till Tantau:
User Guide to the Beamer Class, Version 3.07,
March 11, 2007
- [3] Markus Kohm & Jens-Uwe-Morawski:
KOMA-Script – a versatile L^AT_EX 2_ε bundle,
2007-01-09
- [4] Victor Eijkhout:
An introduction to the Dutch L^AT_EX document classes,
3 September 1989
- [5] Anselm Lingnau:
An Improved Environment for Floats,
2001/11/08
- [6] Mats Dahlgren:
Welcome to the floatflt package,
1998/06/05
- [7] Olga Lapko:
The floatrow package documentation,
2007/08/24
- [8] Sebastian Gross:
Welcome to the beta test of fltpage package!,
1998/11/13
- [9] Sebastian Rahtz & Heiko Oberdiek:
Hypertext marks in L^AT_EX,
November 12, 2007
- [10] Heiko Oberdiek:
The hypcap package – Adjusting anchors of captions,
2007/04/09
- [11] Carsten Heinz & Brooks Moses:
The Listings Package,
2007/02/22
- [12] David Carlisle:
The longtable package,
2004/02/01
- [13] Friedhelm Sowa:
Pictures in Paragraphs,
July 13, 1993

- [14] Joachim Bleser and Edmund Lang:
PicIns-Benutzerhandbuch Version 3.0,
September 1992
- [15] Sebastian Rahtz and Leonor Barroca:
A style option for rotated objects in L^AT_EX,
1997/09/26
- [16] Rolf Niepraschk & Hubert Gäßlein:
The sidecap package,
2003/06/06
- [17] Steven D. Cochran:
The subfigure package,
2002/07/02
- [18] Steven D. Cochran:
The subfig package,
2005/07/05
- [19] Johannes Braams and Theo Jurriens:
The supertabular environment,
2002/07/19
- [20] Donald Arseneau:
Three part tables: title, tabular environment, notes,
2003/06/13
- [21] Donald Arseneau:
WRAPFIG.STY ver 3.6,
2003/01/31
- [22] Anne Brüggemann-Klein:
Einführung in die Dokumentverarbeitung,
B.G. Teubner, Stuttgart, 1989
- [23] Heiko Oberdiek:
The refcount package,
2006/02/20