

The Implementation of the caption package^{*}

Axel Sommerfeldt
`caption@sommerfee.de`

2007/11/04

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 v3.1e, last revised 2007/11/04.

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	17
1.11	Text formats	18
1.12	Fonts	19
1.13	Justifications	20
1.13.1	The ragged2e package	21
1.14	Vertical spaces before and after captions	22
1.15	Positioning	22
1.16	Hooks	24
1.17	Lists	24
1.18	Debug option	25
1.19	Document classes & Babel support	25
1.19.1	The standard L ^A T _E X classes	25
1.19.2	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	25
1.19.3	The beamer class	26
1.19.4	The KOMA-Script classes	27
1.19.5	The NTG Dutch classes	28
1.19.6	The frenchb Babel option	28
1.19.7	The frenchle/pro package	29
1.20	Execution of options	29
1.21	Making an ‘List of’ entry	30
1.22	Typesetting the caption	30
1.23	Bonus features	34
1.24	subfig package adaption	35
2	Main package	37
2.1	Identification	37
2.2	Loading the kernel	37
2.3	Check against incompatible packages	37
2.4	Check document class	37

2.5	Adaption to the \mathcal{AMS} & SMF document classes	38
2.6	Emulation of the KOMA-Script commands	38
2.7	Option for configuration files	40
2.8	Options for figure and table	40
2.9	caption v1.x compatibility options	40
2.10	caption2 v2.x compatibility options	41
2.11	Obsolete caption v3.0 options	41
2.12	ftpage package support options	42
2.13	hyperref package support options	42
2.14	Processing of options	42
2.15	Option ‘type’, \captionof and \ContinuedFloat	42
2.16	Internal helpers	45
2.17	\caption, \@caption, and \@makecaption	47
2.18	Document class & Babel package support	54
2.18.1	The \mathcal{AMS} & SMF classes	54
2.18.2	The beamer class	54
2.18.3	The KOMA-Script classes	54
2.18.4	The frenchb Babel option	54
2.18.5	The frenchle/pro package	54
2.19	Package support	55
2.19.1	The float package	57
2.19.2	The floatflt package	60
2.19.3	The ftpage package	61
2.19.4	The hyperref package	63
2.19.5	The hypcap package	67
2.19.6	The listings package	67
2.19.7	The longtable package	68
2.19.8	The picinpar package	70
2.19.9	The picins package	71
2.19.10	The rotating package	73
2.19.11	The sidecap package	74
2.19.12	The subfigure package	75
2.19.13	The supertabular package	76
2.19.14	The threeparttable package	77
2.19.15	The wrapfig package	77

1 Kernel

1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2007/11/04 v3.1e 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{\langle cmd\rangle}{\langle list entry\rangle}{\langle yes\rangle}{\langle no\rangle}
32 \newcommand*\caption@ifin@list[2]{%
33 \caption@ifempty@list#1%
34 {\@secondoftwo}%
35 {\@expandtwoargs\caption@@ifinlist{#2}{#1}}}

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

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

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

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

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

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

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

```

```

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

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

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

\caption@teststar \caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
\caption@teststar@{<cmd>}{<star arg/token>}{<non-star arg/token>}
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 \newcommand*\caption@ifatletter{%
78   \ifnum\the\catcode`\@=11
79     \expandafter\@firstoftwo
80   \else
81     \expandafter\@secondoftwo
82   \fi}

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

```

```

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

107 \let\caption@CheckCommand\caption@DoCheckCommand
108 \@onlypreamble\caption@CheckCommand

109 \newcommand*\caption@ifCheckCommand{%
110 \ifx\caption@CheckCommand\@gobbletwo
111 \let\next\@firstoftwo
112 \else
113 \let\next\@secondoftwo
114 \fi
115 \let\caption@CheckCommand\caption@DoCheckCommand
116 \next}
117 \@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.

```

118 \let\caption@begindocumenthook\@empty
119 \let\caption@@begindocumenthook\@empty

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

124 \AtBeginDocument{%
125 \PackageInfo{caption}{Begin \noexpand\AtBeginDocument code\@gobble}%

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

130 \def\caption@AtBeginDocument{%
131 \@ifstar\@firstofone\@firstofone}%
132 \caption@@begindocumenthook
133 \let\caption@@begindocumenthook\@undefined

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

```

1.3 Errors

```

\caption@Warning \caption@Error{<message>}
\caption@WarningNoLine 135 \newcommand*\caption@Warning[1]{%
\caption@Error 136 \caption@WarningNoLine{#1\on@line}}
\caption@KV@err 137 \newcommand*\caption@WarningNoLine[1]{%

```

```

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

```

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

```

\caption@eh 142 \newcommand*\caption@wh{%
143 See the caption package documentation for explanation.}
144 \newcommand*\caption@eh{%
145 If you do not understand this error, please take a closer look\MessageBreak
146 at the documentation of the 'caption' package, especially the\MessageBreak
147 section about errors.\MessageBreak\@ehc}

```

1.4 Using the keyval package

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

```

148 \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.

```

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

```

```

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

```

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

```

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

```

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

```

```

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

```

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


```

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

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

173 \newcommand*\caption@option@novalue[2]{%
174   \ifx\KV@err#2%
175     \expandafter\@firstofone
176   \else
177     \KV@err{No value allowed for #1}%
178     \expandafter\@gobble
179   \fi}

```

`\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.

```

180 \newcommand*\captionsetup{%
181   \caption@teststar@\@captionsetup\@gobble\@firstofone}

182 \newcommand*\@captionsetup[1]{%
183   \ifnextchar[{\caption@setup@options#1}\caption@setup}

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

185 \def\caption@setup@options#1[#2]{%
186   % #1 is \@gobble in star form, and \@firstofone otherwise
187   #1{\caption@addtooptlist{#2}}%
188   \expandafter\caption@l@addto@list\csname caption@opt@#2\endcsname}

```

`\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.

```

189 \newcommand*\clearcaptionsetup{%
190   \caption@teststar@\@clearcaptionsetup\@gobble\@firstofone}

191 \newcommand*\@clearcaptionsetup[1]{%
192   \let\caption@tempa#1%
193   \@testopt\@clearcaptionsetup{}}

194 \def\@clearcaptionsetup[#1]#2{%
195   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
196   {\caption@tempa{\caption@Warning{Option list ‘#2’ undefined}}}%
197   {\ifx,#1,%
198     \caption@clearsetup{#2}%
199   \else
200     \caption@@removefromsetup{#1}{#2}%
201   \fi}}

```

```

202 \newcommand*\caption@clearsetup[1]{%
203   \caption@removefromoptlist{#1}%
204   \@nameundef{caption@opt@#1}}

205 \newcommand*\caption@removefromsetup{%
206   \let\caption@tempa\@gobble
207   \caption@@removefromsetup}

208 \newcommand*\caption@@removefromsetup[2]{%
209   \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
210   \expandafter\let\csname caption@opt@#2\endcsname\@undefined
211   \def\@tempb##1=##2\@nil{##1}%
212   \edef\@tempc{#1}%
213   \@for\@tempa:=\@tempa\do{%
214     \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
215     \ifx\@tempd\@tempc
216       \let\caption@tempa\@gobble
217     \else
218       \expandafter\expandafter\expandafter\caption@l@addto@list
219       \expandafter\csname caption@opt@#2\expandafter\endcsname
220       \expandafter{\@tempa}%
221     \fi}%
222   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
223   {\caption@removefromoptlist{#2}}}%
224   \caption@tempa{\caption@Warning{%
225     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>*.

```

226 \newcommand*\showcaptionsetup[2][\@firstofone]{%
227   \GenericWarning{}{%
228     #1 Caption Info: Option list on '#2'\MessageBreak
229     #1 Caption Data: \@ifundefined{caption@opt@#2}{%
230       -none-%
231     }{%
232       {\expandafter\expandafter\expandafter\strip@prefix
233         \expandafter\meaning\csname caption@opt@#2\endcsname}%
234     }}}

235 \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.

```

236 \newcommand*\caption@setoptions[1]{%
237   \caption@Debug{options=#1}%
238   \expandafter\let\expandafter\@tempa\csname caption@opt@#1\endcsname
239   \ifx\@tempa\relax \else
240     \let\caption@iftypewarning\@firstoftwo % = true
241     \expandafter\caption@setup\expandafter{\@tempa}%
242     \caption@clearsetup{#1}%
243   \fi}

```

```

\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”).

```

244 \newcommand*\caption@addtooptlist[1]{%
245   \ifundefined{caption@opt@#1@lineno}{%
246     \caption@dooptlist\caption@g@addto@list{#1}%
247     \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
248   }{}}

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

252 \newcommand*\caption@dooptlist[2]{%
253   \begingroup
254     \edef\@tempa{#2}\@onelevel@sanitize\@tempa
255     \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
256   \endgroup

257 \AtEndDocument{%
258   \caption@for@list\caption@optlist{%
259     \PackageWarningNoLine{caption}{%
260       Unused \string\captionsetup[#1]
261       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.

```

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

263 \long\def\caption@@setkeys[#1]#2#3{%
264   \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
265   \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
266   \ifundefined{#1@KV@err}%
267     {\let\KV@err\ORI@KV@err\let\KV@errx\ORI@KV@errx}%
268     {\expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
269       \let\KV@errx\KV@err}%
270   \edef\caption@keydepth{\caption@keydepth i}%

271   \caption@Debug{\protect\setkeys{#2}{#3}}%
272   \setkeys{#2}{#3}%

273   \edef\caption@keydepth{\expandafter\@cdr\caption@keydepth\@nil}%
274   \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
275   \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname

276 \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 L^AT_EX 2_ε.

```

277 \newcommand*\caption@ExecuteOptions[2]{%
278   \@expandtwoargs\caption@setkeys{#1}{#2}%
279 \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.)

280 \newcommand*\caption@ProcessOptions{%
281   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
282 \@onlypreamble\caption@ProcessOptions

283 \newcommand*\caption@@ProcessOptions[2]{%
284   \let\@tempc\relax
285   \let\caption@tempa\@empty
286   #1{% \@firstofone -or- \@gobble
287     \@for\CurrentOption:=\@classoptionslist\do{%
288       \ifundefined{KV@#2@\CurrentOption}{}{%
289         \ifundefined{KV@#2@\CurrentOption @default}{%
290           \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
291         }{%
292           \PackageInfo{#2}{Global option '\CurrentOption' processed}%
293           \edef\caption@tempa{\caption@tempa,\CurrentOption,}%
294           \@expandtwoargs\@removeelement\CurrentOption
295           \@unusedoptionlist\@unusedoptionlist
296         }%
297       }%
298     }%
299     \let\CurrentOption\@empty
300   }%
301   \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
302   \AtEndOfPackage{\let\@unprocessedoptions\relax}
303 \@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 `caption.sty 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@`.

```

304 \newdimen\captionmargin
305 \newdimen\captionmargin@
306 \newdimen\captionwidth

307 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
308 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
309 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
310 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}

311 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
312 \DeclareCaptionOption{maxmargin}{\caption@setoptcmd\caption@maxmargin{#1}}

```

```

\setcaptionmargin \setcaptionmargin{\family}
\setcaptionmargin*{\family}

```

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

```

313 \newcommand*\setcaptionmargin{%
314   \caption@teststar\caption@setmargin\@gobble\@firstofone}

315 \newcommand*\caption@setmargin[2]{%
316   #1{\captionwidth\z@}%
317   \caption@@setmargin#2,#2,\@nil}

318 \def\caption@@setmargin#1,#2,#3\@nil{%
319   \setlength\captionmargin@{#2}%
320   \setlength\captionmargin{#1}%
321   \addtolength\captionmargin@{-\captionmargin}}

```

`\setcaptionwidth` `\setcaptionwidth{<amount>}`

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

```

322 \newcommand*\setcaptionwidth{%
323   \captionmargin\z@
324   \captionmargin@\z@
325   \setlength\captionwidth}

```

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

```

326 \newcommand*\caption@thecounter{0}

327 \newcommand*\caption@stepcounter{%
328   \@tempcnta\caption@thecounter
329   \advance\@tempcnta\@ne
330   \xdef\caption@thecounter{\the\@tempcnta}}

```

`\caption@newlabel` This command is a modified version of `\newlabel` from \LaTeX 2 ϵ . It will be written to the `.aux` file to pass label information from one run to another. (We use it inside `\caption@ifoddpage` and `\caption@ragged`.)

```

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

```

`\caption@thepage` This command is a modified version of `\thepage` from \LaTeX 2 ϵ . It will be used inside `\caption@ifoddpage` only.

```

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

```

`\caption@label` This command is a modified version of `\label` from \LaTeX 2 ϵ . It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

333 \newcommand*\caption@label[1]{%
334   \protected@write\@auxout{\let\caption@thepage\relax}%
335   {\string\caption@newlabel{#1}\caption@thepage}}

```

`\caption@pageref` This command is a modified version of `\pageref` from \LaTeX 2 ϵ . It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

336 \newcommand*\caption@pageref[1]{%
337   \expandafter\ifx\csname caption@r@#1\endcsname\relax
338     \G@refundefinedtrue % => 'There are undefined references.'
339     \caption@Warning{Reference on page \thepage \space undefined}%
340   \else
341     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
342   \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.

```

343 \newcommand*\caption@ifoddpage{%
344   \caption@iftwoside{%
345     \caption@label\caption@thecounter
346     \caption@pageref\caption@thecounter
347     \ifodd\caption@thepage
348       \let\caption@ifoddpage\@firstoftwo
349     \else
350       \let\caption@ifoddpage\@secondoftwo
351     \fi
352   }\let\caption@ifoddpage\@firstoftwo}%
353 \caption@ifoddpage}

```

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

```

354 \newcommand*\caption@setoptcmd[2]{%
355   \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` 356 \newdimen\caption@indent
`\caption@hangindent` 357 \newdimen\caption@parindent
358 \newdimen\caption@hangindent

```

359 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
360   \setlength\caption@indent{#1}}
361 \DeclareCaptionOption{indentation}[\leftmargini]{%
362   \setlength\caption@indent{#1}}
363 \DeclareCaptionOption{parindent}{%
364   \setlength\caption@parindent{#1}}
365 \DeclareCaptionOption{hangindent}{%
366   \setlength\caption@hangindent{#1}}
367 \DeclareCaptionOption{parskip}{%
368   \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

```

369 \@ifundefined{scr@caption}{}{%
370   \let\caption@KV@parindent\KV@caption@parindent
371   \DeclareCaptionOption{parindent}[]{%
372     \ifx,#1,%
373       \caption@Debug{Option 'parindent' ignored}%
374     \else

```

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 otherwise would still collide with the current version 1.3 of the subfig package (Sigh!)

```

375     \caption@KV@parindent{#1}%
376   \fi}%

377   \let\caption@KV@parskip\KV@caption@parskip
378   \DeclareCaptionOption{parskip}[]{%
379     \ifx,#1,%
380       \caption@Debug{Option 'parskip' ignored}%
381     \else
382       \caption@KV@parskip{#1}%
383     \fi}%
384 }

```

1.7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
385 \newcommand*\DeclareCaptionStyle[1]{%
386   \@testopt{\caption@declarestyle{#1}}{}%
387   \@onlypreamble\DeclareCaptionStyle

388   \def\caption@declarestyle#1[#2]#3{%
389     \global\@namedef{caption@sls@#1}{#2}%
390     \global\@namedef{caption@sty@#1}{#3}%
391     \@onlypreamble\caption@declarestyle

392   \DeclareCaptionOption{style}{\caption@setstyle{#1}}
393   \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 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.

```

394 \newcommand*\caption@setstyle{%
395   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

396 \newcommand*\caption@@setstyle[2]{%
397   \ifundefined{caption@sty@#2}%
398     {#1{\caption@Error{Undefined style '#2'}}}%
399     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
400     \ifx\caption@setstyle@flag\@undefined
401       \let\caption@setstyle@flag\caption@setstyle
402       \let\caption@iftypewarning\@firstoftwo % = true
403       \caption@resetstyle
404       \expandafter\caption@setup\expandafter{\caption@sty}%
405       \let\caption@setstyle@flag\@undefined
406     \else
407       \expandafter\caption@setup\expandafter{\caption@sty}%
408     \fi
409     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
410     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
411     \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!*

```

412 \newcommand*\caption@resetstyle{%
413   \caption@setup{%
414     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
415     justification=justified,font=,labelfont=,textfont=,%
416     margin=0pt,minmargin=0,maxmargin=0,%
417     indent=0pt,parindent=0pt,hangindent=0pt,%
418     singlelinecheck=1,strut=1}%
419   \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.

```

420 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
421 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
422   format=default,labelsep=default,textformat=default,%
423   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.

```

424 \newcommand*\DeclareCaptionFormat{%
425   \caption@teststar\caption@declareformat\@gobble\@firstofone}
426 \@onlypreamble\DeclareCaptionFormat

427 \newcommand*\caption@declareformat[2]{%
428   \@dblarg{\caption@@declareformat#1{#2}}}%
429 \@onlypreamble\caption@declareformat

430 \long\def\caption@@declareformat#1#2[#3]#4{%
431   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
432   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
433   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}%
434 \@onlypreamble\caption@@declareformat

435 \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`).

```

436 \newcommand*\caption@setformat[1]{%
437   \ifundefined{caption@fmt@#1}%
438     {\caption@Error{Undefined format `#1'}}%
439     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
440      \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
441      \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

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

```

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

```



```

443 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
444   \caption@ifin@list\caption@lsep@list\caption@lsepname
445   {\caption@Error{%
446     The option 'labelsep=\caption@lsepname' does not work\MessageBreak
447     with 'format=hang'}}}%
448   {\@hangfrom{#1#2}%
449     \advance\caption@parindent\hangindent
450     \advance\caption@hangindent\hangindent
451     \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

452 \def\caption@fmt@default{\caption@fmt@plain}
453 \def\caption@ifh@default{\caption@ifh@plain}

```

1.9 Label formats

```

\DeclareCaptionLabelFormat \DeclareCaptionLabelFormat{<name>}{<code with #1, #2, and #3>}
454 \newcommand*\DeclareCaptionLabelFormat[2]{%
455   \global\@namedef{caption@lfmt@#1}##1##2##3{#2}}
456 \@onlypreamble\DeclareCaptionLabelFormat

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

```

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

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

```

458 \newcommand*\caption@setlabelformat[1]{%
459   \ifundefined{caption@lfmt@#1}%
460     {\caption@Error{Undefined label format '#1'}}%
461     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}

```

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

```

462 \DeclareCaptionLabelFormat{empty}{}
463 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
464 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace}{#2}}

```

‘default’ usually maps to ‘simple’.

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

1.10 Label separators

```

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

```

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

```

466 \newcommand\DeclareCaptionLabelSeparator{%
467   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
468 \@onlypreamble\DeclareCaptionLabelSeparator

469 \newcommand\caption@declarelabelseparator[3]{%
470   \global\expandafter\let\csname caption@iflf@#2\endcsname#1%
471   \global\long\@namedef{caption@lsep@#2}{#3}%
472   \caption@@declarelabelseparator{#2}#3\\@nil}
473 \@onlypreamble\caption@declarelabelseparator

```

```

474 \long\def\caption@@declarelabelseparator#1#2\\#3\@nil{%
475   \def\@tempa{#3}\ifx\@tempa\@empty \else
476     \caption@g@addto@list\caption@lsep@crlist{#1}%
477   \fi}
478 \@onlypreamble\caption@@declarelabelseparator

479 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
480 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}

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

```

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

```

481 \newcommand*\caption@setlabelseparator[1]{%
482   \@ifundefined{caption@lsep@#1}%
483     {\caption@Error{Undefined label separator `#1'}}%
484     {\edef\caption@lsepname{#1}%
485       \expandafter\let\expandafter\caption@ifl@csname\caption@ifl@#1\endcsname
486       \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’.

```

487 \DeclareCaptionLabelSeparator{none}{}
488 \DeclareCaptionLabelSeparator{colon}{: }
489 \DeclareCaptionLabelSeparator{period}{. }
490 \DeclareCaptionLabelSeparator{space}{ }
491 \DeclareCaptionLabelSeparator*{quad}{\quad}
492 \DeclareCaptionLabelSeparator*{newline}{\\}
493 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

```

‘default’ usually maps to ‘colon’.

```

494 \def\caption@lsep@default{\caption@lsep@colon}
495 \def\caption@ifl@default{\caption@ifl@colon}

```

1.11 Text formats

```

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

496 \newcommand*\DeclareCaptionTextFormat[2]{%
497   \global\long\@namedef{caption@tfmt@#1}##1{#2}}
498 \@onlypreamble\DeclareCaptionTextFormat

499 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
500 \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`).

```

501 \newcommand*\caption@settextformat[1]{%
502   \@ifundefined{caption@tfmt@#1}%
503     {\caption@Error{Undefined text format `#1'}}%
504     {\expandafter\let\expandafter\caption@tfmt\csname\caption@tfmt@#1\endcsname}}

```

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

```

505 \DeclareCaptionTextFormat{simple}{#1}
506 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

1.12 Fonts

```

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

DeclareDefaultCaptionFont \DeclareDefaultCaptionFont{<name>}{<code>}
511 \newcommand*\DeclareDefaultCaptionFont[2]{%
512   \global\@namedef{caption#1@default}{#2}}
513 \@onlypreamble\DeclareDefaultCaptionFont

514 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
515 \DeclareDefaultCaptionFont{font}{}

516 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
517 \DeclareDefaultCaptionFont{labelfont}{}

518 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
519 \DeclareDefaultCaptionFont{textfont}{}

\caption@setfont \caption@setfont{<name>}{<keyval-list of names>}
Selecting a caption font means saving all the code snippets (in \caption#1). Because we use \setkeys recursive here we need to do this inside an extra group and collect all the code snippets in \caption@tempa first.
520 \newcommand*\caption@setfont[2]{%
521   \begingroup
522     \let\caption@fnt\@empty
523     \define@key{caption@fnt}{default}[]{%
524       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
525     \caption@setkeys[caption]{caption@fnt}{#2}%
526     \global\let\caption@tempa\caption@fnt
527   \endgroup
528   \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.
529 \newcommand*\caption@font{%
530   \caption@teststar\caption@@font\@firstofone
531     {\caption@setkeys[caption]{caption@fnt}}}
532 \newcommand*\caption@@font[2]{%
533   \begingroup
534     \def\caption@fnt{\endgroup}%
535     #1{#2}%
536   \caption@fnt}

```

These are the pre-defined font code snippets.

```

537 \DeclareCaptionFont{normalcolor}{\normalcolor}
538 \DeclareCaptionFont{color}{\color{#1}}

539 \DeclareCaptionFont{normalfont}{\normalfont}
540 \DeclareCaptionFont{up}{\upshape}
541 \DeclareCaptionFont{it}{\itshape}
542 \DeclareCaptionFont{sl}{\slshape}

```

```

543 \DeclareCaptionFont{sc}{\scshape}
544 \DeclareCaptionFont{md}{\mdseries}
545 \DeclareCaptionFont{bf}{\bfseries}
546 \DeclareCaptionFont{rm}{\rmfamily}
547 \DeclareCaptionFont{sf}{\sffamily}
548 \DeclareCaptionFont{tt}{\ttfamily}

549 \DeclareCaptionFont{scriptsize}{\scriptsize}
550 \DeclareCaptionFont{footnotesize}{\footnotesize}
551 \DeclareCaptionFont{small}{\small}
552 \DeclareCaptionFont{normalsize}{\normalsize}
553 \DeclareCaptionFont{large}{\large}
554 \DeclareCaptionFont{Large}{\Large}

555 \DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespace}% normally 1
556 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
557 \DeclareCaptionFont{doublespacing}{\doublespacing}
558 \DeclareCaptionFont{stretch}{\setstretch{#1}}
559 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }

560 %\DeclareCaptionFont{normal}{%
561 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
562 \DeclareCaptionFont{normal}{%
563   \caption@font*{%
564     \KV@caption@fnt@normalcolor\@unused
565     \KV@caption@fnt@normalfont\@unused
566     \KV@caption@fnt@normalsize\@unused
567     \KV@caption@fnt@singlespacing\@unused}}

```

`\captionsize` 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.

```

568 \providecommand\captionsize{}

569 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
570 \DeclareDefaultCaptionFont{size}{}

```

1.13 Justifications

```

\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
571 \newcommand*\DeclareCaptionJustification[2]{%
572   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
573   \DeclareCaptionFont{#1}{#2}}
574 \@onlypreamble\DeclareCaptionJustification

\DeclareDefaultCaptionJustification{<code>}
575 \newcommand*\DeclareDefaultCaptionJustification[1]{%
576   \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
577   \DeclareDefaultCaptionFont{@hj}{#1}}
578 \@onlypreamble\DeclareDefaultCaptionJustification

579 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
580 \DeclareDefaultCaptionJustification{}

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

```

These are the pre-defined justification code snippets.

```
582 \DeclareCaptionJustification{justified}{}
583 \DeclareCaptionJustification{centering}{\centering}
584 \DeclareCaptionJustification{centerfirst}{\centerfirst}
585 \DeclareCaptionJustification{centerlast}{\centerlast}
586 \DeclareCaptionJustification{raggedleft}{\raggedleft}
587 \DeclareCaptionJustification{raggedright}{\raggedright}
```

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

```
588 \providecommand\centerfirst{%
589   \let\\@centercr
590   \edef\caption@normaladjust{%
591     \leftskip\the\leftskip
592     \rightskip\the\rightskip
593     \parfillskip\the\parfillskip\relax}%
594   \leftskip\z@\@plus -1fil%
595   \rightskip\z@\@plus 1fil%
596   \parfillskip\z@skip
597   \noindent\hskip\z@\@plus 2fil%
598   \@setpar{\@@par\@restorepar\caption@normaladjust}}
```

`\centerlast` This is based on code from Anne Brüggemann-Klein[\[22\]](#)

```
599 \providecommand\centerlast{%
600   \let\\@centercr
601   \leftskip\z@\@plus 1fil%
602   \rightskip\z@\@plus -1fil%
603   \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.

```
604 \DeclareCaptionJustification{Centering}{%
605   \caption@ragged\Centering\centering}
606 \DeclareCaptionJustification{RaggedLeft}{%
607   \caption@ragged\RaggedLeft\raggedleft}
608 \DeclareCaptionJustification{RaggedRight}{%
609   \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.

```
610 \newcommand*\caption@ragged{%
611   \caption@Debug{We need ragged2e}%
612   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
613   \global\let\caption@ragged\caption@ragged
614   \caption@ragged}
```

```

615 \caption@AtBeginDocument{%
616   \@ifundefined{caption@r@ragged2e}{%
617     \newcommand*\caption@@ragged{%
618       \caption@Warning{%
619         'ragged2e' support has been changed.\MessageBreak
620         Rerun to get captions right}%
621       \global\let\caption@ragged\@secondoftwo % suppress further warnings
622       \caption@ragged}%
623   }{%
624     \caption@Debug{We load ragged2e}%
625     \IfFileExists{ragged2e.sty}{%
626       \RequirePackage{ragged2e}%
627       \let\caption@@ragged\@firstoftwo
628     }{%
629       \newcommand*\caption@@ragged[2]{%
630         \@ifundefined{caption\string#1}{%
631           \caption@Warning{%
632             'ragged2e' package not loaded, therefore\MessageBreak
633             substituting \string#2 for \string#1\MessageBreak}%
634           \global\@namedef{caption\string#1}}{%
635             #2}%
636         }%
637       }%

```

1.14 Vertical spaces before and after captions

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

```

638 \@ifundefined{abovecaptionskip}{%
639   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p}}{}
640 \@ifundefined{belowcaptionskip}{%
641   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p}}{}

642 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
643 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
644 \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.

```

645 \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.*

```

646 \newcommand*\caption@setposition[1]{%
647   \caption@ifinlist{#1}{d,default}{%

```

```

648 \let\caption@position\caption@defaultpos
649 }\caption@ifinlist{#1}{t,top,above}{%
650 \let\caption@position\@firstoftwo
651 }\caption@ifinlist{#1}{b,bottom,below}{%
652 \let\caption@position\@secondoftwo
653 }\caption@ifinlist{#1}{a,auto}{%
654 \let\caption@position\@undefined
655 }{%
656 \caption@Error{Undefined position '#1'}%
657 }}}}

```

`\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.

```

658 %\caption@setdefaultpos{a}% default = auto
659 \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.)

```

660 \newcommand*\caption@iftop{%
661 \ifx\caption@position\@undefined
662 \let\caption@position\@secondoftwo
663 % = \caption@setposition b%
664 \fi
665 \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.

```

666 \newcommand*\caption@fixposition{%
667 \ifx\caption@position\@undefined
668 \caption@autoposition
669 \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.

```

670 \newcommand*\caption@autoposition{%
671 \ifvmode
672 \edef\caption@tempa{\the\prevdepth}%
673 \caption@Debug{\protect\prevdepth=\caption@tempa}%
674 \ifdim\prevdepth>-\p@
675 \let\caption@position\@secondoftwo
676 \else
677 \let\caption@position\@firstoftwo

```

```

678 \fi
679 % = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
680 \else
681 \caption@Debug{no \protect\prevdepth}%
682 \let\caption@position\@secondoftwo
683 % = \caption@setposition b%
684 \fi}

```

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

```

685 \newcommand*\caption@setautoposition[1]{%
686 \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`.

```

687 \newcommand*\caption@beginhook{}
688 \newcommand*\caption@endhook{}
689 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
690 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

1.17 Lists

```

691 \DeclareCaptionOption{listof}{\caption@set@bool\caption@iflof{#1}}
\DeclareCaptionListFormat \DeclareCaptionListFormat {<name>}{<code with #1 and #2>}
692 \newcommand*\DeclareCaptionListFormat [2]{%
693 \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
694 \@onlypreamble\DeclareCaptionListFormat

```

```

695 \DeclareCaptionOption{listofformat}{\caption@setlistofformat{#1}}

```

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

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

```

696 \newcommand*\caption@setlistofformat[1]{%
697 \@ifundefined{caption@lstfmt@#1}%
698 {\caption@Error{Undefined list-of format `#1'}}%
699 {\expandafter\let\expandafter\caption@lstfmt
700 \csname caption@lstfmt@#1\endcsname}}

```

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

```

701 \DeclareCaptionListFormat{empty}{}
702 \DeclareCaptionListFormat{simple}{#1#2}
703 \DeclareCaptionListFormat{parens}{#1(#2)}
704 \DeclareCaptionListFormat{subsimple}{#2}
705 \DeclareCaptionListFormat{subparens}{(#2)}
706 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}

```


1.18 Debug option

```
707 \DeclareCaptionOption{debug}[1]{%
708   \caption@set@bool\caption@ifdebug{#1}%
709   \let\caption@iftypewarning\caption@ifdebug
710   \caption@ifdebug
711   {\def\caption@Debug{\PackageInfo{caption}}}%
712   {\let\caption@Debug\@gobble}}
713 \DeclareOption{debug}{\setkeys{caption}{debug}}
714 \setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

```
715 \caption@CheckCommand\@makecaption{\let\@makecaption\@undefined}
```

1.19.1 The standard L^AT_EX classes

```
716 \caption@CheckCommand\@makecaption{%
717   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
718   \long\def\@makecaption#1#2{%
719     \vskip\abovcaptionskip
720     \sbox\@tempboxa{#1: #2}%
721     \ifdim \wd\@tempboxa >\hsize
722       #1: #2\par
723     \else
724       \global \@minipagefalse
725       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
726     \fi
727     \vskip\belowcaptionskip}}
```

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

```
728 \ifundefined{@captionheadfont}{}{%
729   \caption@CheckCommand\@makecaption{%
730     % amsart|amsproc|amsbook [2004/08/06 v2.20]
731     \long\def\@makecaption#1#2{%
732       \setbox\@tempboxa\vbox{\color@setgroup
733         \advance\hsize-2\captionindent\noindent
734         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
735           {\@cdr#2\@nil}\@captionfont\upshape\enspace#2}%
736       \unskip\kern-2\captionindent\par
737       \global\setbox\@ne\lastbox\color@endgroup}%
738       \ifhbox\@ne % the normal case
739         \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
740       \fi
741       \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
742         \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
743       \else % tempboxa contained more than one line
744         \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
745           \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
746       \fi
747       \ifnum\@tempcnta<64 % if the float IS a figure...
748         \addvspace\abovcaptionskip
749         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
750       \else % if the float IS NOT a figure...
751         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
752       \fi
753     }%
754   }%
755 }
```

```

752         \nobreak
753         \vskip\belowcaptionskip
754     \fi
755     \relax
756 }
757 \caption@CheckCommand\@makecaption{%
758     % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publiés par
759     \long\def\@makecaption#1#2{%
760         \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
761         \setbox\@tempboxa\vbox{\color@setgroup
762             \advance\hsize-2\captionindent\noindent
763             \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
764                 {\@cdr#2\@nil}\@addpunct{.}\@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             \@tempdima\wd\@ne\advance\@tempdima-\captionindent
773             \wd\@ne\@tempdima
774         \else % tempboxa contained more than one line
775             \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
776                 \unvbox\@tempboxa\parskip\z@skip
777                 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
778         \fi
779         \ifnum\@tempcnta<64 % if the float IS a figure...
780             \addvspace\abovecaptionskip
781             \noindent\kern\captionindent\box\@ne
782         \else % if the float IS NOT a figure...
783             \noindent\kern\captionindent\box\@ne
784             \nobreak
785             \vskip\belowcaptionskip
786         \fi
787     \relax
788 }
789 \let\captionmargin\captionindent % set to 3pc by AMS class
790 \begingroup\edef\@tempa{\endgroup
791     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
792     {margin=\the\captionmargin
793     \@ifundefined{smf@makecaption}{},{,maxmargin=.1\linewidth}}}
794 \@tempa
795 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
796 \DeclareCaptionLabelSeparator{default}{.\enspace}
797 \DeclareDefaultCaptionFont{font}{\@captionfont}
798 \DeclareDefaultCaptionFont{labelfont}{\@captionheadfont}
799 \DeclareDefaultCaptionFont{textfont}{\@captionfont\upshape}
800 \captionsetup[figure]{position=b}
801 \captionsetup[table]{position=t}
802 }

```

1.19.3 The beamer class

```

803 \@ifclassloaded{beamer}{%
804   \caption@CheckCommand\beamer@makecaption{%
805     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
806     \long\def\beamer@makecaption#1#2{%
807       \def\insertcaptionname{\csname#1name\endcsname}%
808       \def\insertcaptionnumber{\csname the#1\endcsname}%
809       \def\insertcaption{#2}%
810       \nobreak\vskip\abovecaptionskip\nobreak
811       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
812       \ifdim \wd\@tempboxa >\hsize
813         \usebeamertemplate**{caption}\par
814       \else
815         \global \@minipagefalse
816         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
817       \fi
818       \nobreak\vskip\belowcaptionskip\nobreak}}
819 \DeclareCaptionLabelFormat{default}{#1}
820 \DeclareDefaultCaptionJustification{\raggedright}
821 \DeclareDefaultCaptionFont{font}{%
822   \usebeamerfont*{caption}%
823   \usebeamerfont*{caption}}
824 \DeclareDefaultCaptionFont{labelfont}{%
825   \usebeamerfont*{caption name}%
826   \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.)

```

827 \defbeamertemplate{caption}{caption3}{%
828   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
829 \DeclareOption{beamer}{%
830   % \usebeamertemplate**{caption} will set font
831   \DeclareDefaultCaptionFont{font}{}%
832   \setbeamertemplate{caption}[caption3]}
833 {}

```

1.19.4 The KOMA-Script classes

```

834 \@ifundefined{scr@caption}{}{%
835   \caption@CheckCommand\@makecaption{%
836     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
837     \long\def\@makecaption#1#2{%
838       \if@captionabove
839         \vskip\belowcaptionskip
840       \else
841         \vskip\abovecaptionskip
842       \fi
843       \@@makecaption\@firstofone{#1}{#2}%
844       \if@captionabove
845         \vskip\abovecaptionskip
846       \else
847         \vskip\belowcaptionskip
848       \fi}}

```

```

849 \DeclareCaptionFormat{default}[#1#2#3\par]{%
850   \ifdofullc@p
851     \caption@ifin@list\caption@lsep@crlist\caption@lsepname
852     {\caption@Error{%
853       The option 'labelsep=\caption@lsepname' does not work\MessageBreak
854       with \noexpand\setcaphanging (which is set by default)}}%
855     {\caption@fmt@hang{#1}{#2}{#3}}%
856   \else
857     #1#2%
858     \ifdim\cap@indent<\z@
859       \par
860       \noindent\hspace*{-\cap@indent}%
861     \else\if@capbreak
862       \par
863     \fi\fi
864     #3\par
865   \fi}
866 \DeclareCaptionLabelSeparator{default}{\captionformat}
867 \DeclareDefaultCaptionFont{font}{\scr@fnt@caption}
868 \DeclareDefaultCaptionFont{labelfont}{\scr@fnt@captionlabel}
869 }

```

1.19.5 The NTG Dutch classes

```

870 \@ifundefined{CaptionFonts}{}{%
871   \caption@CheckCommand\@makecaption{%
872     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
873     \long\def\@makecaption#1#2{%
874       \vskip\abovecaptionskip
875       \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
876       \ifdim \wd\@tempboxa >\hsize
877         {\CaptionLabelFont#1:} \CaptionTextFont#2\par
878       \else
879         \global \@minipagefalse
880         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
881       \fi
882       \vskip\belowcaptionskip}}
883 \DeclareDefaultCaptionFont{labelfont}{\CaptionLabelFont}
884 \DeclareDefaultCaptionFont{textfont}{\CaptionTextFont}
885 }

```

1.19.6 The frenchb Babel option

```

886 \@ifundefined{FB@makecaption}{}{%
887   \caption@CheckCommand\@makecaption{%
888     % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
889     \long\def\@makecaption#1#2{%
890       \vskip\abovecaptionskip
891       \sbox\@tempboxa{#1\CaptionSeparator #2}%
892       \ifdim \wd\@tempboxa >\hsize
893         #1\CaptionSeparator #2\par
894       \else
895         \global \@minipagefalse
896         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%

```

```

897     \fi
898     \vskip\belowcaptionskip}}

899 \ifx\@makecaption\STD@makecaption
900   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
901   \def\caption@frenchb{% supress frenchb warning
902     \let\STD@makecaption\@makecaption
903     \let\FB@makecaption\@makecaption}
904 \else
905   \ifx\@makecaption\@undefined\else
906     \PackageInfo{caption}{%
907       The definition of \protect\@makecaption\space
908       has been changed,\MessageBreak
909       frenchb will NOT customize it}%
910   \fi
911 \fi

912 }

```

1.19.7 The frenchle/pro package

```

913 \@ifundefined{frenchTeXmods}{}{%

914   \caption@CheckCommand\@makecaption{%
915     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
916     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
917     \def\@makecaption#1#2{%
918       \ifFTY%
919         \def\@secondofmany##1##2\void{##2}%
920         \def\@tempa{\@secondofmany#2\void}%
921         \ifx\@tempa\empty%
922           \let\captionseparator\empty%
923         \fi%
924         \@mcORI{#1}{\relax\captionfont{#2}}%
925       \else
926         \@mcORI{#1}{#2}%
927       \fi}}

928   \caption@CheckCommand\@makecaption{%
929     % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
930     \def\@makecaption#1#2{%
931       \ifFTY%
932         \def\@secondofmany##1##2\void{##2}%
933         \protected@edef\@tempa{\@secondofmany#2\void}%
934         \ifx\@tempa\empty%
935           \let\captionseparator\empty%
936         \fi%
937         \@mcORI{#1}{\relax\captionfont{#2}}%
938       \else
939         \@mcORI{#1}{#2}%
940       \fi}}

941   \DeclareDefaultCaptionFont{textfont}{\itshape}%
942   \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%

943 }

```

1.20 Execution of options

```

944 \captionsetup{style=default,position=default,listof=1,listofformat=default,%
945             twoside=\if@twoside 1\else 0\fi}

946 \ProcessOptions*

947 \caption@ifcheckcommand{%
948   \caption@setbool{documentclass}{1}%
949 }{%
950   \caption@setbool{documentclass}{0}%
951   \PackageInfo{caption}{%
952     Unknown document class (or package), \MessageBreak
953     standard defaults will be used}%
954   \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
955 }

```

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`.

```

956 \newcommand*\caption@addcontentsline[2]{%
957   \caption@iflof%
958   {\def\@tempa{#2}}%
959   {\let\@tempa\@empty}%
960   \ifx\@tempa\@empty \else
961     {\let\\space
962      \addcontentsline{\csname ext@#1\endcsname}{#1}%
963                      {\protect\numberline
964                      {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
965                      {\ignorespaces #2}}}%
966   \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`.)

```

967 \newif\ifcaption@star

```

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

```

968 \newcommand*\caption@fnum[1]{%
969   \caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}{\theContinuedFloat}}

```

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

```

970 \newcommand\caption@make[2]{%
971   \caption@@make{\caption@lfmt{#1}{#2}}{\theContinuedFloat}}

```

`\caption@@make` `\caption@@make{<caption label>}{<caption text>}`

```

972 \newcommand\caption@@make[2]{%
973   \begingroup
974   \caption@stepcounter
975   \caption@beginhook

```

Check margin, if \caption@minmargin or \caption@maxmargin is set

```

976 \ifx\caption@maxmargin\undefined \else
977   \ifdim\captionmargin>\caption@maxmargin\relax
978     \captionmargin\caption@maxmargin\relax
979   \fi
980 \fi
981 \ifx\caption@minmargin\undefined \else
982   \ifdim\captionmargin<\caption@minmargin\relax
983     \captionmargin\caption@minmargin\relax
984   \fi
985 \fi

```

Special single-line treatment (option singlelinecheck=)

```

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

```

Typeset the left margin (option margin=)

```

987 \caption@calcmargin
988 \@tempdima\captionmargin
989 \ifdim\captionmargin@=\z@ \else
990   \caption@ifoddpagel{\advance\@tempdima\captionmargin@}%
991 \fi
992 \caption@ifh{\advance\@tempdima\caption@indent}%
993 \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*.

```

994 \@tempdima\captionwidth
995 \caption@ifh{\advance\@tempdima-\caption@indent}%
996 \captionbox\@tempdima{%

```

Typeset the indentation (option indentation=)

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

```

997 \caption@ifh{%
998   \ifdim\caption@indent=\z@
999     \leavevmode
1000   \else
1001     \hskip-\caption@indent
1002   \fi}%

```

Typeset the caption itself and close the \captionbox

```

1003 \caption@@@make{#1}{#2}}%

```

Typeset the right margin (option margin=)

```

1004 \@tempdima\captionmargin
1005 \ifdim\captionmargin@=\z@ \else
1006   \caption@ifoddpagel{\advance\@tempdima\captionmargin@}{}%
1007 \fi
1008 \hspace\@tempdima
1009 \caption@endhook
1010 \endgroup
1011 \global\caption@starfalse}

```

\caption@calcmargin \caption@calcmargin

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

```

1012 \newcommand*\caption@calcmargin{%
1013   \ifdim\captionwidth=\z@
1014     \captionwidth\linewidth
1015     \advance\captionwidth by -2\captionmargin
1016     \advance\captionwidth by -\captionmargin@
1017   \else
1018     \captionmargin\linewidth
1019     \advance\captionmargin by -\captionwidth
1020     \divide\captionmargin by 2
1021     \captionmargin@\z@
1022   \fi

1023 \caption@Debug{%
1024   \string\hsize=\the\hsize,
1025   \string\linewidth=\the\linewidth,\MessageBreak
1026   \string\leftmargin=\the\leftmargin,
1027   \string\rightmargin=\the\rightmargin,\MessageBreak
1028   \string\margin=\the\captionmargin,
1029   \string\margin@=\the\captionmargin@,
1030   \string\width=\the\captionwidth}%
1031 }

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.

1032 \newcommand\caption@slc[4]{%
1033   \caption@Debug{Begin SLC}%
1034   \begingroup
1035   \caption@singleline
1036   \let\caption@hj\@empty
1037   \caption@calcmargin % calculate #3 if necessary
1038   \caption@prepareslc
1039   \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1040   \ifdim\wd\@tempboxa>#3%
1041     \endgroup
1042   \else
1043     \endgroup
1044     \caption@singleline
1045     #4%
1046   \fi
1047   \caption@Debug{End SLC}}

1048 \newcommand*\caption@singleline{%
1049   \expandafter\caption@setup\expandafter{\caption@opt@singleline}%
1050   \let\caption@fmt\caption@slfmt}

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

1051 \newcommand*\caption@prepareslc{%
1052   \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1053   \def\label{\caption@withoptargs\@gobbletwo}%
1054   \let\stepcounter\caption@l@stepcounter
1055   \let\refstepcounter\stepcounter\let\H\refstepcounter\stepcounter}

1056 \newcommand*\caption@l@stepcounter[1]{\advance\c@#1\endcsname\@ne\relax}

```


`\captionbox` `\captionbox{<width>}{<contents>}`

This macro defines the box which surrounds the caption paragraph.

1057 `\newcommand*\captionbox{\parbox[b]}`

`\caption@@@make` `\caption@@@make{<caption label>}{<caption text>}`

This one finally typesets the caption paragraph, without margin and indentation.

1058 `\newcommand\caption@@@make[2]{%`

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

1059 `\sbox\@tempboxa{#1}%`

1060 `\ifdim\wd\@tempboxa=\z@`

1061 `\let\caption@lsep\relax`

1062 `% \@capbreakfalse`

1063 `\fi`

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

1064 `\caption@ifempty{#2}{%`

1065 `\let\caption@lsep\relax`

1066 `% \@capbreakfalse`

1067 `% \let\caption@ifstrut\@secondoftwo`

1068 `% }`

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

1069 `\@setpar{\@par\caption@@par}\caption@@par`

Finally typeset the caption.

1070 `\caption@hj\captionsize\captionfont\caption@fmt`

1071 `{\ifcaption@star\else\captionlabelfont#1}\fi}%`

1072 `{\ifcaption@star\else\caption@iflf\captionlabelfont\caption@lsep}\fi}%`

1073 `{\captiontextfont`

1074 `\caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}}%`

1075 `\nobreak\hskip\z@skip % enable hyphenation`

1076 `\caption@tfmt{#2}%`

1077 `% \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}}%`

1078 `\caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}}%`

1079 `\par}}`

`\caption@ifempty` `\caption@ifempty{<text>}{<true>}(no <false>)`

This one tests if the *<text>* is actually empty.

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

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

1080 `\newcommand\caption@ifempty[1]{%`

1081 `\def\caption@tempa{#1}%`

1082 `\def\caption@tempb{\ignorespaces}%`

1083 `\ifx\caption@tempa\caption@tempb`

1084 `\let\caption@tempa\@empty`

1085 `\fi`

1086 `\ifx\caption@tempa\@empty`

1087 `\let\caption@ifempty\@secondoftwo`

1088 `\else`

1089 `\let\caption@ifempty\@gobbletwo`

1090 `\fi`

1091 `\caption@ifempty\@unused}`

```

\caption@@par \caption@@par
This command will be executed with every \par inside the caption.
1092 \newcommand*\caption@@par{%
1093   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

1.23 Bonus features

```

eclareFloatingEnvironment \DeclareFloatingEnvironment[⟨file ext⟩]{⟨environment⟩}[⟨name⟩][⟨list name⟩]
1094 \providecommand*\DeclareFloatingEnvironment{%
1095   \@ifnextchar[%]
1096     \caption@@DeclareFloatingEnvironment\caption@DeclareFloatingEnvironment
1097   \@onlypreamble\DeclareFloatingEnvironment

1098 \newcommand*\caption@DeclareFloatingEnvironment[1]{%
1099   \caption@@DeclareFloatingEnvironment[lo#1]{#1}%
1100   \@onlypreamble\caption@DeclareFloatingEnvironment

1101 \def\caption@@DeclareFloatingEnvironment[#1]#2{%
1102   \ifundefined{c@chapter}%
1103     {\newcounter{#2}%
1104       \@namedef{the#2}{\arabic{#2}}%
1105       \@namedef{theH#2}{\arabic{#2}}}%
1106     {\newcounter{#2}[chapter]%
1107       \@namedef{the#2}{\ifnum\c@chapter>\z@ \thechapter.\fi \arabic{#2}}%
1108       \@namedef{theH#2}{\theHchapter.\arabic{#2}}}%
1109   \@namedef{fps@#2}{tbp}%
1110   \ifundefined{c@float@type}%
1111     {\newcounter{float@type}%
1112       \setcounter{float@type}{\ifundefined{c@figure}14}}%
1113     {}%
1114   \caption@Debug{float type '#2'=\the\value{float@type}}%
1115   \expandafter\xdef\csname ftype#2\endcsname{\the\value{float@type}}%
1116   \addtocounter{float@type}{\value{float@type}}%
1117   \@namedef{ext@#2}{#1}%
1118   \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1119   \newenvironment{#2}{\@float{#2}}{\end@float}%
1120   \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1121   \expandafter\newcommand\csname listof#2s\endcsname{\begingroup
1122     \expandafter\let\expandafter\listfigurename\csname list#2name\endcsname
1123     \expandafter\let\expandafter\ext@figure\csname ext@#2\endcsname
1124     \let\caption@ORI@starttoc\@starttoc
1125     \renewcommand*\@starttoc[1]{%
1126       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1127     \listoffigures
1128     \endgroup}%
1129   \ifundefined{l@figure}%
1130     {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1131     {\expandafter\let\csname l@#2\endcsname\l@figure}%
1132   \expandafter\newcommand\csname #2name\endcsname{}%
1133   \edef\@tempa{%
1134     \def\noexpand\@tempa{\expandafter\noexpand\@car#2\@nil}}%
1135   \uppercase\expandafter{\@tempa}%
1136   \edef\@tempb{%
1137     \noexpand\g@addto@macro\noexpand\@tempa{\expandafter\@cdr#2\@nil}}%

```

```

1138 \@tempb
1139 \expandafter\let\csname #2name\endcsname\@tempa
1140 \expandafter\newcommand\csname list#2name\endcsname{ }%
1141 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1142 \@cons\caption@floatlist{{#2}}%
1143 \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#2}}\relax}
1144 \@onlypreamble\caption@@DeclareFloatingEnvironment
1145 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1146 \@namedef{#1name}{#2}%
1147 \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#1}}\relax}
1148 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1149 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1150 \@namedef{list#1name}{#2}}
1151 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1152 \newcommand*\caption@floatlist{}

```

```

\ForFloatingEnvironments \ForFloatingEnvironments{<code with #1>}
1153 \providecommand\ForFloatingEnvironments[1]{%
1154 \caption@AtBeginDocument{%
1155 \def\@elt##1{#1}%
1156 \caption@floatlist
1157 \let\@elt\relax}}%
1158 \@onlypreamble\ForFloatingEnvironments

```

1.24 subfig package adaptations

Unfortunately the author of the subfig package does not seem to maintain his package anymore, so we have to make several adaptations here.

```

1159 \caption@AtBeginDocument{%
1160 \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1161 \ifx\subref\@tempa
1162 \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1163 \let\caption@setfloattype\@gobble
1164 % \let\sf@ifinlist\caption@ifinlist
1165 \let\sf@ifpositiontop\caption@iftop
1166 \def\@dottedxxxline#1#2#3#4#5#6#7{%
1167 \begingroup
1168 \caption@setfloattype{#1}%
1169 \caption@setoptions{subfloat}%
1170 \caption@setoptions{sub#1}%
1171 \ifnum #3>\@nameuse{c@#2depth}\else
1172 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1173 \fi
1174 \endgroup}%
1175 \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.

```

1176 \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%

```

```

1177 \caption@removefromoptlist{subfloat}%
1178 \caption@removefromoptlist{subfigure}%
1179 \caption@removefromoptlist{subtable}%

1180 \fi
1181 \let\sf@NEW@subfloat\@undefined}

1182 \def\sf@NEW@subfloat{%
1183 \beginingroup
1184 \caption@setfloattype\@capttype
1185 \sf@ifpositiontop{%
1186 \maincaptiontoptrue
1187 }{%
1188 \maincaptiontopfalse
1189 }%
1190 \caption@setoptions{subfloat}%
1191 \caption@setoptions{sub\@capttype}%
1192 \let\sf@oldlabel=\label
1193 \let\label=\subfloat@label
1194 \ifmaincaptiontop\else
1195 \advance\@nameuse{c@\@capttype}\@ne
1196 \fi
1197 \refstepcounter{sub\@capttype}%
1198 \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1199 \@ifnextchar [% %] match left bracket
1200 {\sf@@subfloat}%
1201 {\sf@@subfloat[\@empty]}}%

```

2 Main package

2.1 Identification

```
1202 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1203 \ProvidesPackage{caption}[2007/11/04 v3.1e Customising captions (AR)]
1204 %\@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.
1205 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1206 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1207 \RequirePackage{caption3}[2007/11/04] % needs v3.1e or newer
```

2.3 Check against incompatible packages

```
1208 \@ifpackageloaded{caption2}{%
1209   \caption@Error{%
1210     You can't use both, the (obsolete) caption2 *and* \MessageBreak
1211     the (current) caption package}%
1212   \endinput
1213 }{}

1214 \caption@AtBeginDocument{%
1215   \@ifpackageloaded{ftcap}{%
1216     \caption@IncompatiblePackage{ftcap}{}%
1217   }{}%
1218   \@ifpackageloaded{nonfloat}{%
1219     \caption@IncompatiblePackage{nonfloat}{.\MessageBreak
1220       Furthermore you could use \noexpand\captionof instead of \MessageBreak
1221       \noexpand\figcaption or \string\tabcaption}%
1222   }{}%
1223   \@ifpackageloaded{topcapt}{%
1224     \caption@IncompatiblePackage{topcapt}{}%
1225   }{}%
1226 }

\caption@Incompatible Issue an 'incompatible' warning.
1227 \newcommand*\caption@IncompatiblePackage[2]{%
1228   \PackageWarningNoLine{caption}{%
1229     Usage of the #1 package together with the \MessageBreak
1230     caption package is strongly *not* recommended.\MessageBreak
1231     Instead of loading the #1 package you should \MessageBreak
1232     use the caption package option \MessageBreak
1233     \space\space `tableposition=top' #2}}
1234 \@onlypreamble\caption@IncompatiblePackage
```

2.4 Check document class

```
1235 \caption@ifbool{documentclass}{}{}%
1236   \PackageWarningNoLine{caption}{%
1237     Unsupported document class (or package) detected, \MessageBreak
```

```

1238     usage of the caption package is *not* recommended}%
1239 \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1240 }

```

2.5 Adaption to the \mathcal{AMS} & SMF document classes

```

1241 \@ifundefined{@captionheadfont}{}{}%
1242 \caption@Info{AMS or SMF document class}%
1243 \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1244 }

```

2.6 Emulation of the KOMA-Script commands

```

1245 \@ifundefined{scr@caption}{}{}%
1246 \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 1247 \def\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1248 \def\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1249 \if@tablecaptionabove
1250 \caption@AtBeginDocument{%
1251 \def\caption@tempa#1{%
1252 \ifpackageloaded{#1}{\caption@Error{%
1253 You can't use the global option\MessageBreak
1254 \space\space'tablecaptionabove'\MessageBreak
1255 and use the #1 package, too}}}%
1256 \caption@tempa{ftcap}%
1257 \caption@tempa{nonfloat}%
1258 \caption@tempa{topcapt}}%
1259 \@tablecaptionabovetrue
1260 \else
1261 \@tablecaptionabovefalse
1262 \fi

\onlinecaptionstrue
\onlinecaptionfalse 1263 \def\onlinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1264 \def\onlinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1265 \ifonlinecaptions
1266 \onlinecaptionstrue
1267 \else
1268 \onlinecaptionfalse
1269 \fi

\@captionabovetrue
\@captionabovefalse 1270 \def\@captionabovetrue{\let\caption@position\@firstoftwo}
1271 \def\@captionabovefalse{\let\caption@position\@secondoftwo}

\setcapindent
1272 \let\caption@KOMA@setcapindent\@setcapindent
1273 \renewcommand*\@setcapindent[1]{%
1274 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

```

```

1275 \let\caption@KOMA@@setcapindent\@@setcapindent
1276 \renewcommand*\@@setcapindent[1]{%
1277   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1278 \newcommand*\caption@setcapindent{%
1279   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1280 \@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.)
1281 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1282   \csname\string\setcapwidth\endcsname
1283 \@namedef{\string\setcapwidth}[1]#2{%
1284   \caption@KOMA@setcapwidth{#1}{#2}\caption@setcapwidth{#1}}
1285 \newcommand*\caption@setcapwidth[1]{%
1286   \ifx\#1\\\else
1287     \@ifundefined{cap@margin}{}{%
1288       \def\@tempa{captionbeside}%
1289       \ifx\@tempa\@currenvir\else\caption@Warning{%
1290         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1291       \fi}%
1292   \fi
1293   \captionsetup{width=\cap@width}}

1294 \def\caption@tempa{\hsize}%
1295 \ifx\caption@tempa\cap@width \else
1296   \caption@setcapwidth{?}
1297 \fi

\setcapmargin
1298 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1299   \csname\string\@setcapmargin\endcsname
1300 \@namedef{\string\@setcapmargin}[1]#2{%
1301   \caption@KOMA@setcapmargin{#1}{#2}\caption@setcapmargin}
1302 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1303   \csname\string\@@setcapmargin\endcsname
1304 \@namedef{\string\@@setcapmargin}[1]#2{%
1305   \caption@KOMA@@setcapmargin{#1}{#2}\caption@setcapmargin}
1306 \newcommand*\caption@setcapmargin{%
1307   \begingroup
1308     \let\onelinecaptionsfalse\relax
1309     \def\@twoside{0}%
1310     \def\if@twoside{\def\@twoside{1}\iffalse}%
1311     \cap@margin
1312     \def\@tempa{\endgroup}%
1313     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1314       \def\hspace##1##{\@firstofone}%
1315       \edef\@tempa{\endgroup
1316         \noexpand\captionsetup{%
1317           twoside=\@twoside,singlelinecheck=0,%
1318           margin={\cap@left,\cap@right}}}%
1319     \fi\fi
1320   \@tempa}

```

```

1321 \ifx\cap@margin\relax \else
1322 \caption@setcapmargin
1323 \fi
1324 }

```

2.7 Option for configuration files

```

1325 \DeclareCaptionOption*{config}[caption]{%
1326 \InputIfFileExists{#1.cfg}%
1327 {\typeout{*** Local configuration file #1.cfg used ***}}%
1328 {\caption@Warning{Configuration file #1.cfg not found}}

```

2.8 Options for figure and table

```

1329 \DeclareCaptionOption{figureposition}{%
1330 \captionsetup*[figure]{position=#1}}
1331 \@onlypreamble@key{caption}{figureposition}

1332 \DeclareCaptionOption{tableposition}{%
1333 \caption@AtBeginDocument{%
1334 \def\caption@tempa##1{%
1335 \ifpackageloaded{##1}{\caption@Error{%
1336 You can't use the caption package option\MessageBreak
1337 \space\space`tableposition=#1'\MessageBreak
1338 and use the ##1 package, too}}{}}%
1339 \caption@tempa{ftcap}%
1340 \caption@tempa{nonfloat}%
1341 \caption@tempa{topcapt}}}%
1342 \captionsetup*[table]{position=#1}}
1343 \@onlypreamble@key{caption}{tableposition}

1344 \DeclareCaptionOption{figurename}{\caption@SetFloatName{figure}{#1}}
1345 \@onlypreamble@key{caption}{figurename}
1346 \DeclareCaptionOption{tablename}{\caption@SetFloatName{table}{#1}}
1347 \@onlypreamble@key{caption}{tablename}

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

1348 \newcommand*\caption@SetFloatName[2]{%
1349 \begingroup
1350 \def\@tempa{\endgroup\@firstofone}%
1351 \ifundefined{language\@tempa}{%
1352 \ifundefined{captions\language\@tempa}{%
1353 \def\@tempa{\endgroup
1354 \expandafter\addto\csname captions\language\@tempa\endcsname}}}%
1355 \@tempa{\@namedef{#1name}{#2}}}%
1356 \@onlypreamble\caption@SetFloatName

```

2.9 caption v1.x compatibility options

```

1357 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1358 \@onlypreamble@key{caption}{compatibility}

1359 \DeclareCaptionOptionNoValue*{normal}{%
1360 \caption@setformat{plain}%
1361 \caption@setjustification{justified}}
1362 \DeclareCaptionOptionNoValue*{isu}{%

```



```

1363 \caption@setformat{hang}%
1364 \caption@setjustification{justified}}
1365 \DeclareCaptionOptionNoValue*{hang}{%
1366 \caption@setformat{hang}%
1367 \caption@setjustification{justified}}
1368 \DeclareCaptionOptionNoValue*{center}{%
1369 \caption@setformat{plain}%
1370 \caption@setjustification{centering}}
1371 \DeclareCaptionOptionNoValue*{anne}{%
1372 \caption@setformat{plain}%
1373 \caption@setjustification{centerlast}}
1374 \DeclareCaptionOptionNoValue*{centerlast}{%
1375 \caption@setformat{plain}%
1376 \caption@setjustification{centerlast}}

1377 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1378 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1379 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1380 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1381 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1382 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1383 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1384 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1385 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1386 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1387 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1388 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1389 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1390 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1391 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1392 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}

1393 \caption@setbool{ruled}{0}
1394 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}

```

2.10 caption2 v2.x compatibility options

```

1395 \DeclareCaptionOptionNoValue*{flushleft}{%
1396 \caption@setformat{plain}%
1397 \caption@setjustification{raggedright}}
1398 \DeclareCaptionOptionNoValue*{flushright}{%
1399 \caption@setformat{plain}%
1400 \caption@setjustification{raggedleft}}

1401 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}

1402 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
1403 \caption@WarningNoLine{Obsolete option 'ignoreLTcapwidth' ignored}}

```

2.11 Obsolete caption v3.0 options

```

1404 \DeclareCaptionOption*{caption}{%
1405 \caption@setbool{temp}{#1}%
1406 \caption@ifbool{temp}{}{%
1407 \caption@Error{%
1408 The package option 'caption=#1' is obsolete.\MessageBreak
1409 Please pass this option to the subfig package instead\MessageBreak

```

```
1410         and do *not* load the caption package anymore}}}
```

2.12 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.

```
1411 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1412 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1413 \@onlypreamble@key{caption}{FPlist}
1414 \@onlypreamble@key{caption}{FPref}

1415 \newcommand*\caption@setFPoption[2]{%
1416   \edef\@tempa{\@car#2\@nil}%
1417   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1418 \@onlypreamble\caption@setFPoption

1419 \captionsetup{FPlist=caption,FPref=figure}
```

2.13 hyperref package support options

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

```
1420 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1421 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}

1422 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

2.14 Processing of options

```
1423 \caption@ProcessOptions*{caption}
```

2.15 Option ‘type’, `\captionof` and `\ContinuedFloat`

```
1424 \caption@AtBeginDocument{%
1425   \DeclareCaptionOption{type}{\caption@settype{#1}}
1426   \DeclareCaptionOption{type*}{\caption@settype*{#1}}}
```

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@settypehook` (which will be used by our float package support).

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

```
1427 \newcommand*\caption@settype{%
1428   \caption@@settype{}}

1429 \newcommand*\caption@@settype[1]{%
1430   \caption@teststar{\caption@@settype{#1}}{\@firstoftwo\@secondoftwo}
```

```

1431 \newcommand*\caption@@@settype[3]{%
1432 % #1 = "" or "sub"
1433 % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1434 % #3 = <type>, e.g. "figure" or "table"
1435 \caption@Debug{type=#3}%
1436 \caption@checkgrouplevel{#1}{%
1437   \captionsetup{#1type#2*\@empty=...}#2{ or \@backslashchar#1captionof}{}}}%
1438 \@ifundefined{@#1capttype}{}{%
1439   \edef\@tempa{#3}%
1440   \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1441     \caption@iftypewarning{\caption@Warning{%
1442       Caption type was already set to '#3'\MessageBreak}}{}%
1443   \fi}%
1444 \expandafter\edef\csname @#1capttype\endcsname{#3}%
1445 \let\caption@iftypewarning\caption@ifdebug
1446 \@nameuse{#1caption@settypehook}%
1447 \expandafter\caption@setoptions\csname @#1capttype\endcsname
1448 \let\caption@ifrefstepcounter\@secondoftwo
1449 #2{}{%
1450   \let\@currentlabel\caption@undefinedlabel
1451 % \let\@currentHlabel\@undefined
1452   \ifx\caption@ORI@label\@undefined
1453     \let\caption@ORI@label\label
1454     \def\label{\caption@withoptargs\caption@xlabel}%
1455   \fi
1456   \caption@start}}

```

`\caption@settypehook` Hook, will be extended by our float package support.

```
1457 \newcommand*\caption@settypehook{}
```

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

```

1458 \newcommand*\caption@iftype{%
1459   \@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 an error will be issued. (needs ϵ -TeX)

```

1460 \begingroup\expandafter\expandafter\expandafter\endgroup
1461 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1462   \caption@Debug{TeX engine: TeX}
1463   \let\caption@checkgrouplevel\@gobbletwo
1464 \else
1465   \caption@Debug{TeX engine: e-TeX}
1466   \newcommand*\caption@checkgrouplevel[2]{%
1467     \@ifundefined{#1caption@grouplevel}{%
1468       \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}}{%
1469       \ifnum\currentgrouplevel>\caption@grouplevel\relax
1470         \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1471           \the\currentgrouplevel}%
1472       \else
1473         \caption@Error{\string#2\MessageBreak outside box or environment}%
1474       \fi

```

```

1475     } {} }
1476 \fi

\caption@undefinedlabel This label will be used for \currentlabel inside (floating) environments as default.
                        (see above)
1477 \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.
1478 \newcommand*\caption@xlabel[2]{%
1479   \ifx\@currentlabel\caption@undefinedlabel
1480     \caption@Warning{\noexpand\label before \noexpand\caption ignored}%
1481   \else
1482     \caption@ORI@label#1{#2}%
1483   \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)
1484 \caption@AtBeginDocument{%
1485   \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}
1486 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

\captionlistentry \captionlistentry[<float type>]{<list entry>}
1487 \newcommand*\captionlistentry{%
1488   \@testopt\caption@listentry\@capttype}

1489 \def\caption@listentry[#1]#2{%
1490   \caption@refstepcounter{#1}%
1491   \caption@makecurrent{#1}{#2}%
1492   \caption@addcontentsline{#1}{#2}}

\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 pack-
age, except for the optional argument.
1493 \def\ContinuedFloat{%
1494   \@ifnextchar[\@ContinuedFloat{%
1495     \caption@iftype
1496       {\addtocounter\@capttype\m@ne
1497        \caption@ContinuedFloat\@capttype}%
1498     {\caption@Error{\noexpand\ContinuedFloat outside float}}}}
1499 \def\caption@ContinuedFloat#1{%
1500   \edef\@tempa{#1}%
1501   \ifx\@tempa\caption@CFtype \else
1502     \caption@Error{Continued `#1' after '\caption@CFtype'}%

```

```

1503 \fi
1504 \stepcounter{ContinuedFloat}%
1505 \caption@@ContinuedFloat{#1}%
1506 \caption@setoptions{ContinuedFloat}%
1507 \caption@setoptions{continued#1}%
1508 \let\caption@resetContinuedFloat\@gobble}

1509 \def\@ContinuedFloat[#1]{\addtocounter{#1}\m@ne}
1510 \newcounter{ContinuedFloat}
1511 \newcommand*\caption@CFtype{??}

\caption@@ContinuedFloat \caption@@ContinuedFloat{<type>}
Usually this do nothing but this changes if the hyperref package is loaded. (See hyperref
package support for details.)
1512 \newcommand*\caption@@ContinuedFloat[1]{}

\caption@resetContinuedFloat \caption@resetContinuedFloat{<type>}
If a continuation counter is defined, we reset it. (This one will be called inside
\@caption.)
1513 \newcommand*\caption@resetContinuedFloat[1]{%
1514 \st@pelt{ContinuedFloat}\xdef\caption@CFtype{#1}}

```

2.16 Internal helpers

```

\caption@refstepcounter Resets the continuation counter, increments the float (i.e. figure or table) counter,
and sets the refstepcounter flag.
1515 \newcommand*\caption@refstepcounter[1]{%
1516 \caption@resetContinuedFloat{#1}%
1517 \caption@@refstepcounter{#1}%
1518 \let\caption@ifrefstepcounter\@firstoftwo}

1519 \newcommand*\caption@@refstepcounter{\refstepcounter}
1520 \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[]{}).
1521 \newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1522 \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)

1523 \newcommand*\caption@begin[1]{%
1524 \begingroup
1525 \caption@setfnum{#1}%
1526 \caption@fixposition
1527 \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>`.

```
1528 \newcommand\caption@beginex[3]{%
1529   \caption@begin{#1}%
1530   \caption@addcontentsline{#1}{#2}%
1531   \caption@ifempty{#3}{} }
```

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

```
1532 \newcommand*\caption@end{%
1533   \endgroup
1534   \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.

```
1535 \newcommand*\caption@setfnum[1]{%
1536   \ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1537   \namedef{fnum@#1}{\caption@fnum{#1}}%
1538   \fi }
```

`\caption@addcontentsline` If `\sf@ifpositiontop` is defined, we extend `\caption@addcontentsline` so
it handles sub-captions, too.

```
1539 \caption@AtBeginDocument*{%
1540   \ifundefined{sf@ifpositiontop}{}{%
1541     \let\caption@kernel@addcontentsline\caption@addcontentsline
1542     \renewcommand*\caption@addcontentsline[2]{%
1543       \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
1544       \caption@kernel@addcontentsline{#1}{#2}%
1545       \sf@ifpositiontop{}{\@listsubcaptions{#1}}}%
1546   }}
```

`\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`.

```

1547 \newcommand*\caption@boxrestore{%
1548   \let\if@nobreak\iffalse
1549   \let\if@noskipsec\iffalse
1550   \let\par\@par
1551   % \let\-\@dischyph
1552   \let'\@acci\let'\@accii\let\=\@acciii
1553   \parindent\z@ \parskip\z@skip
1554   \everypar{}%
1555   % \linewidth\hsize
1556   % \@totalleftmargin\z@
1557   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1558   \parfillskip\@flushglue \lineskip\normallineskip
1559   \baselineskip\normalbaselineskip
1560   \sloppy
1561   \let\\ \@normalcr
1562 }
```

`\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).

```

1563 \newcommand*\caption@normalsize{%
1564   \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.

```
1565 \let\caption@setfloatcapt\@firstofone
```

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

```

\caption@@anchor 1566 \newcommand*\caption@makecurrent[2]{}
\caption@start    1567 \let\caption@@anchor\@firstofone
\caption@@start   1568 \let\caption@start\relax
\caption@freezeHref 1569 \let\caption@@start\relax
\caption@defrostHref 1570 \let\caption@freezeHref\relax
                  1571 \let\caption@defrostHref\relax
```

2.17 `\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 its 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.

```

1572 \caption@AtBeginDocument{%
1573   \caption@setbool{incompatible}{0}%
```

```

1574 \caption@CheckCommand\caption{%
1575   % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1576   \def\caption{%
1577     \ifx\@capttype\@undefined
1578       \@latex@error{\noexpand\caption outside float}\@ehd
1579       \expandafter\@gobble
1580     \else
1581       \refstepcounter\@capttype
1582       \expandafter\@firstofone
1583     \fi
1584     {\@dblarg{\@caption\@capttype}}%
1585   }%
1586 \caption@CheckCommand\caption{%
1587   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1588   \def\caption{
1589     \ifx\@capttype\@undefined
1590       \@latex@error{\noexpand\caption outside figure or table}\@ehd
1591       \expandafter\@gobble
1592     \else
1593       \refstepcounter\@capttype
1594       \expandafter\@firstofone
1595     \fi
1596     {\@dblarg{\@caption\@capttype}}%
1597   }%
1598 \caption@CheckCommand\caption{%
1599   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1600   \renewcommand\caption{%
1601     \ifx\@capttype\@undefined
1602       \@latex@error{\noexpand\caption outside float}\@ehd
1603       \expandafter\@gobble
1604     \else
1605       \refstepcounter\@capttype
1606       \let\@tempf\@caption
1607       \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1608         \expandafter\expandafter\let
1609           \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1610       \fi
1611       \fi
1612       \@dblarg{\@tempf\@capttype}}%
1613 \caption@CheckCommand\caption{%
1614   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1615   % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1616   % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1617   \def\caption{%
1618     \ifx\@capttype\@undefined
1619       \@latex@error{\noexpand\caption outside float}\@ehd
1620       \expandafter\@gobble
1621     \else
1622       \H@refstepcounter\@capttype
1623       \@ifundefined{fst@\@capttype}{%
1624         \let\Hy@tempa\@caption
1625       }{%
1626         \let\Hy@tempa\Hy@float@caption

```



```

1627         }%
1628         \expandafter\@firstofone
1629         \fi
1630         {\@dblarg{\Hy@tempa\@capttype}}%
1631     } }%
1632 \caption@CheckCommand\caption{%
1633 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1634 \def\caption{%
1635     \ifx\@capttype\@undefined
1636     \@latex@error{\noexpand\caption outside float}\@ehd
1637     \expandafter\@gobble
1638     \else
1639     \H@refstepcounter\@capttype
1640     \let\Hy@tempa\@caption
1641     \@ifundefined{float@caption}{%
1642     }{%
1643     \expandafter\ifx\csname @float@c\@capttype\endcsname\float@caption
1644     \let\Hy@tempa\Hy@float@caption
1645     \fi
1646     }%
1647     \expandafter\@firstofone
1648     \fi
1649     {\@dblarg{\Hy@tempa\@capttype}}%
1650 } }%
1651 \caption@IfCheckCommand{}{%
1652 \caption@Info{%
1653     Incompatible package detected (regarding \string\caption).\MessageBreak
1654     \string\caption\space=\space\meaning\caption}%
1655 \caption@setbool{incompatible}{1}}%
1656 \caption@CheckCommand\caption{%
1657 % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1658 \long\def\caption#1[#2]#3{%
1659     \par
1660     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1661     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1662     \begingroup
1663     \@parboxrestore
1664     \if@minipage
1665     \@setminipage
1666     \fi
1667     \normalsize
1668     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1669     \endgroup}%
1670 \caption@CheckCommand\caption{%
1671 % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1672 \long\def\caption#1[#2]#3{% second argument ignored
1673     \par\nobreak
1674     \begingroup
1675     \@parboxrestore
1676     \if@minipage
1677     \@setminipage
1678     \fi
1679     \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak

```

```

1680         \endgroup}}%
1681 % \caption@CheckCommand\float@caption{%
1682 % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1683 % \long\def\float@caption#1[#2]#3{%
1684 % \addcontentsline{\@nameuse{ext@#1}}{#1}%
1685 % {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
1686 % \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
1687 % \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
1688 % \@ifnextchar[{\float@ccon}{\egroup}}%
1689 % \long\def\float@ccon[#1]{#1\par\egroup}}%
1690 \caption@CheckCommand\@caption{%
1691 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1692 % \long\def\@caption#1[#2]#3{%
1693 % \hyper@makecurrent{\@capytype}%
1694 % \def\@currentlabelname{#2}%
1695 % \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1696 % \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1697 % }%
1698 % \begingroup
1699 % \@parboxrestore
1700 % \if@minipage
1701 % \setminipage
1702 % \fi
1703 % \normalsize
1704 % \@makecaption{\csname fnum@#1\endcsname}{%
1705 % \ignorespaces
1706 % \ifHy@nesting
1707 % \hyper@@anchor{\@currentHref}{#3}%
1708 % \else
1709 % \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1710 % \fi
1711 % }%
1712 % \par
1713 % \endgroup
1714 % }}%
1715 \caption@CheckCommand\@caption{%
1716 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1717 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1718 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1719 % \long\def\@caption#1[#2]#3{%
1720 % \expandafter\ifx\csname if@capstart\expandafter\endcsname
1721 % \csname iftrue\endcsname
1722 % \global\let\@currentHref\hc@currentHref
1723 % \else
1724 % \hyper@makecurrent{\@capytype}%
1725 % \fi
1726 % \def\@currentlabelname{#2}%
1727 % \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1728 % \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1729 % }%
1730 % \begingroup
1731 % \@parboxrestore
1732 % \if@minipage

```

```

1733         \@setminipage
1734     \fi
1735     \normalsize
1736     \expandafter\ifx\csname if@capstart\expandafter\endcsname
1737         \csname iftrue\endcsname
1738         \global\@capstartfalse
1739         \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1740     \else
1741         \@makecaption{\csname fnum@#1\endcsname}{%
1742             \ignorespaces
1743             \ifHy@nesting
1744                 \hyper@@anchor{\@currentHref}{#3}%
1745             \else
1746                 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1747             \fi
1748         }%
1749     \fi
1750     \par
1751 \endgroup
1752 }}%

1753 \caption@CheckCommand\@caption{%
1754     % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1755     \long\def\@caption#1[#2]{%
1756         \def\@currentlabelname{#2}%
1757         \NR@@caption{#1}[{#2}]%
1758     }%

1759 \caption@CheckCommand\@caption{%
1760     % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1761     \long\def\@caption#1[#2]#3{%
1762         \@ifundefined{if#1topcap}%
1763             {\subfig@oldcaption{#1}[{#2}]{#3}}%
1764             {\@nameuse{if#1topcap}%
1765                 \@listsubcaptions{#1}%
1766                 \subfig@oldcaption{#1}[{#2}]{#3}%
1767             \else
1768                 \subfig@oldcaption{#1}[{#2}]{#3}%
1769                 \@listsubcaptions{#1}%
1770             \fi}}%

1771 \caption@CheckCommand\@caption{%
1772     % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1773     \def\@caption{\caption@}%
1774 % \long\def\caption@#1[#2]#3{%
1775 %     \@ifundefined{caption@setfloattype}%
1776 %         \caption@settype
1777 %         \caption@setfloattype
1778 %         \capttype
1779 %     \sf@ifpositiontop{%
1780 %         \@listsubcaptions{#1}%
1781 %         \sf@old@caption{#1}[{#2}]{#3}%
1782 %     }{%
1783 %         \sf@old@caption{#1}[{#2}]{#3}%
1784 %         \@listsubcaptions{#1}%
1785 %     }%

```

```

1786 }%
1787 \caption@ifcheckcommand{}{%
1788 \caption@info{%
1789 Incompatible package detected (regarding \string\@caption).\MessageBreak
1790 \string\@caption\space=\space\meaning\@caption}%
1791 \caption@setbool{incompatible}{1}}%

```

The option `compatibility=` will override the compatibility mode.

```

1792 \@ifundefined{caption@ifcompatibility}%
1793 {\let\caption@ifcompatibility\caption@ifincompatible
1794 \let\@tempa\@firstofone}%
1795 {\let\@tempa\@gobble}% suppress warning
1796 \caption@ifcompatibility{%
1797 \@tempa{%
1798 \caption@warningnoline{%
1799 \noexpand\caption will not be redefined since it's already\MessageBreak
1800 redefined by a document class or package which is\MessageBreak
1801 unknown to the caption package.\MessageBreak
1802 As a result some features will not work as expected}%
1803 }%

```

`\ContinuedFloat` is not supported in compatibility mode.

```

1804 \renewcommand*\caption@ContinuedFloat[1]{%
1805 \caption@error{Not available in compatibility mode}}%

```

`\caption@start` is not supported in compatibility mode.

```

1806 \caption@atbegindocument*{%
1807 \let\caption@start\relax
1808 \@ifundefined{caption@ORI@capstart}}{%
1809 \caption@debug{%
1810 Restore hycap definition of \string\capstart\@gobble}%
1811 \let\capstart\caption@ORI@capstart}%
1812 \@ifundefined{caption@ORI@float@makebox}}{%
1813 \caption@debug{%
1814 Restore hyperref redefinition of \string\float@makebox\@gobble}%
1815 \let\float@makebox\caption@ORI@float@makebox}%
1816 }%

```

`\caption@star` We define `\caption@star` here, too, so it's defined but does not make any harm.

```

1817 \newcommand*\caption@star[2]{#1#2}%
1818 }{%
1819 \caption@ifincompatible{%
1820 \caption@warningnoline{%
1821 Forced redefinition of \noexpand\caption since the\MessageBreak
1822 unsupported(!) package option 'compatibility=false'\MessageBreak
1823 was given. Please fasten your seat belts, as we may\MessageBreak
1824 experience turbulence}%
1825 }{}%

```

`\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`.

```
1826 \def\caption{%
1827   \caption@iftyp
1828   {\caption@checkgrouplevel\@empty\caption
1829   \caption@star
1830   {\caption@refstepcounter\@capttype}%
1831   {\caption@dblarg{\@caption\@capttype}}}%
1832   {\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.

```
1833 \newcommand*\caption@star[2]{%
1834   \ifstar{\global\caption@startrue#2[]}{#1#2}}%
```

`\@caption` As above, our version has been adapted to the packages we support. Additionally our code is nested by `\caption@beginex` & `\caption@end` instead of `\begingroup` & `\endgroup`. Furthermore we use `\caption@boxrestore` instead of `\@parboxrestore` so this code also works correctly inside list-based environments like `wide` & `addmargin`. (This, and the fact that we use `\linewidth` instead of `\hsize` inside `\@makecaption`, solves [L^AT_EX PR latex/2472](#).)

```
1835 \long\def\@caption#1[#2]#3{%
1836   \ifcaption@star \else
1837   \caption@makecurrent{#1}{#2}%
1838   \caption@ifhycap{\caption@@start}{}%
1839   \fi
1840   \par
1841   \caption@beginex{#1}{#2}{#3}%
1842   \caption@setfloatcapt{%
1843     \caption@boxrestore
1844     \caption@normalsize
1845     \ifcaption@star
1846     \let\caption@@anchor\@firstofone
1847     \fi
1848     \@makecaption{\csname fnum@#1\endcsname}%
1849     {\ignorespaces\caption@@anchor{#3}}\par}%
1850   \caption@end}%
1851 }%
1852 \caption@AtBeginDocument*{%
1853   \let\caption@ORI@capstart\@undefined
1854   \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.

```
1855 \let\caption@ORI@xfloat\@xfloat
1856 \def\@xfloat#1[#2]{%
1857   \caption@ORI@xfloat{#1}[#2}%
1858   \caption@settype{#1}}%
1859 }
```

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

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

```
1860 \long\def\@makecaption#1#2{%
1861   \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
1862   \caption@@make{#1}{#2}%
1863   \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}}
```

2.18 Document class & Babel package support

2.18.1 The $\mathcal{M}\mathcal{S}$ & SMF classes

```
1864 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}
```

2.18.2 The beamer class

```
1865 \@ifclassloaded{beamer}{%
1866   \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.

```
1867   \captionsetup{listof=false}
1868   \undefine@key{caption}{listof}
1869   \undefine@key{caption}{listofformat}
1870 }{ }
```

2.18.3 The KOMA-Script classes

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

```
1871 \@ifundefined{scr@caption}{}{%
1872   \caption@AtBeginDocument{\let\scr@caption\caption}}
```

2.18.4 The frenchb Babel option

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

```
1873 \@ifundefined{caption@frenchb}{}{\caption@frenchb}
```

2.18.5 The frenchle/pro package

```
1874 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
1875   \caption@Info{frenchle/pro package is loaded}%
1876   \let\captionfont@ORI\captionfont
1877   \let\captionlabelfont@ORI\captionlabelfont
1878   \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.

```
1879   \@ifundefined{GOfrench}%
1880     {\let\@tempa\@firstofone}%
```

```

1881      {\def\@tempa{\g@addto@macro\GOfrench}}}%
1882      \@tempa{%
1883          \let\captionfont\captionfont@ORI
1884          \let\captionfont@ORI\@undefined
1885          \let\captionlabelfont\captionlabelfont@ORI
1886          \let\captionlabelfont@ORI\@undefined
1887          \let\@makecaption\@makecaption@ORI
1888          \let\@makecaption@ORI\@undefined
1889
\@cnORI We update the definition of \@cnORI so it actually reflects our definition of \caption.
1889      \let\@cnORI\caption
1890
\@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.
1890      \let\caption@tcORI\@tablescaption
1891      \def\@tablescaption{\caption@star\relax\caption@tcORI}%
1892
\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.
1892      \let\@eatDP\@undefined
1893      \let\@tempa\@empty
1894      \ifx\ffrench\figure
1895          \l@addto@macro\@tempa{\let\fnum@figure\ffrench}%
1896      \fi
1897      \ifx\tfrench\table
1898          \l@addto@macro\@tempa{\let\fnum@table\tfrench}%
1899      \fi
1900      \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
1901      \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
1902      \@tempa
1903  }%
1904 }}

```

2.19 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.
1905 \newcommand\caption@ifpackageloaded[1]{%
1906     \testopt{\caption@ifpackageloaded{#1}}{}%
1907     \@onlypreamble\caption@ifpackageloaded
1908     \long\def\caption@ifpackageloaded#1[#2]#3#4{%
1909         \ifpackageloaded{#1}\@firstofone{%
1910             \caption@debug{#1 package is not loaded (yet)\@gobble}%
1911             \caption@atbegindocument}%
1912         \caption@ifpackageloaded{#1}[#2]{#3}{#4}}%
1913     \@onlypreamble\caption@ifpackageloaded

```

```

1914 \newcommand\caption@ifpackageloaded[1]{%
1915   \@testopt{\caption@@ifpackageloaded{#1}}{}}
1916 \@onlypreamble\caption@ifpackageloaded

1917 \long\def\caption@@ifpackageloaded#1[#2]{%
1918   \@ifpackageloaded{#1}{%
1919     \caption@Info{#1 package is loaded}%
1920     \@ifpackagelater{#1}{#2}\@firstoftwo{%
1921       \caption@Error{%
1922         For a successful cooperation we need at least version\MessageBreak
1923         `#2' of package #1,\MessageBreak
1924         but only version\MessageBreak
1925         '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
1926         is available}%
1927       \@secondoftwo}%
1928     }\@secondoftwo}}
1929 \@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.

```

1930 \newcommand*\caption@clearmargin{%
1931   \setcaptionmargin\z@
1932   \let\caption@minmargin\@undefined}

1933 \caption@setbool{needfreeze}{0}
1934 \caption@AtBeginDocument*{%
1935   \caption@ifneedfreeze{%

```

`\caption@freeze` `\caption@freeze*`
Used by the `fltpage` & `sidecap` package support.

```

1936 \newcommand*\caption@freeze{%
1937   \caption@teststar\caption@@freeze\@gobble\@firstofone}%

1938 \newcommand*\caption@@freeze[1]{%
1939   \global\let\caption@SCcontinued\relax
1940   \global\let\caption@SCsetup\@undefined
1941   \global\let\caption@SClentry\@undefined
1942   \global\let\caption@SCtext\@undefined
1943   \global\let\caption@SClabel\@undefined

1944   \let\caption@ORI@ContinuedFloat\ContinuedFloat
1945   \def\ContinuedFloat{%
1946     \let\caption@ORI@setcounter\setcounter
1947     \let\caption@ORI@addtocounter\addtocounter
1948     \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
1949     \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
1950     \caption@ORI@ContinuedFloat
1951     \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
1952     \let\setcounter\caption@ORI@setcounter
1953     \let\addtocounter\caption@ORI@addtocounter}%
1954   \def\captionsetup##1{%
1955     \caption@g@addto@list\caption@SCsetup{##1}%
1956     \caption@setup{##1}}%
1957   \let\caption@ORI\caption
1958   \def\caption{%
1959     \def\caption{\caption@Error{%

```



```

1960         Only one \noexpand\caption can be placed in this environment}}%
1961         \let\captionsetup\caption@setup
1962         \let\caption@@refstepcounter\caption@l@stepcounter
1963         \caption@ORI}%
1964     \long\def\@caption##1[##2]##3{%
1965         \gdef\caption@SClentry{##2}\gdef\caption@SCtext{##3}%
1966         \ignorespaces}%
1967     #1{% is \@gobble in star form, and \@firstofone otherwise
1968         \def\label##1{\gdef\caption@SCLabel{##1}\ignorespaces}}%
1969     }%

\caption@defrost \caption@defrost
1970 \newcommand*\caption@defrost{%
1971     \ifx\caption@ORI@ContinuedFloat\@undefined
1972         \caption@SCcontinued
1973     \ifx\caption@SCsetup\@undefined \else
1974         \expandafter\captionsetup\expandafter{\caption@SCsetup}%
1975     \fi
1976     \ifx\caption@SCtext\@undefined \else
1977         \expandafter\expandafter\expandafter\caption
1978         \expandafter\expandafter\expandafter[%
1979         \expandafter\expandafter\expandafter{%
1980         \expandafter\caption@SClentry\expandafter}\expandafter]%
1981         \expandafter{\caption@SCtext}%
1982     \fi
1983     \ifx\caption@SCLabel\@undefined \else
1984         \expandafter\label\expandafter{\caption@SCLabel}%
1985     \fi
1986     \else
1987         \caption@Error{Internal Error:\MessageBreak
1988         \noexpand\caption@defrost in same group as \string\caption@freeze}%
1989     \fi}%

1990 }{}%
1991 \caption@undefbool{needfreeze}}

```

2.19.1 The float package

The float package usually do not use the L^AT_EX 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.)

```

1992 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
1993     \@ifpackageloaded{floatrow}{%
1994         \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
1995     }{}%

```

`\@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>]`.

```
1996 \let\caption@ORI@float@setevery\@float@setevery
1997 \def\@float@setevery#1{%
1998   \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.

```
1999   \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.

```
2000   \renewcommand\caption@setfloatcapt{%
2001     \let\@makecaption\caption@make
2002     \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.

```
2003   \float@getstyle\float@style{#1}%
2004   \caption@setstyle*\float@style
2005   \caption@setoptions\float@style
2006   }{}%
2007   \caption@freezeHref % will be defrosted in \float@makebox
2008   \caption@ORI@float@setevery{#1}}%
```

`\caption@settypehook` 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.

```
2009 \g@addto@macro\caption@settypehook{%
2010   \expandafter\ifx\csname #1name\endcsname\relax
2011     \expandafter\let\csname #1name\expandafter\endcsname
2012                           \csname fname@#1\endcsname
2013   \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.

```
2014 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2015 \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`).

```
2016 \providecommand*\float@ifstyle[1]{%
2017   \expandafter\ifx\csname fst@#1\endcsname\relax
2018     \expandafter\@secondoftwo
2019   \else
2020     \expandafter\@firstoftwo
2021   \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.

```
2022 \providecommand*\float@getstyle[2]{%
2023   \edef#1{%
2024     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2025     \expandafter\expandafter\expandafter\noexpand
2026     \csname fst@#2\endcsname}%
2027   \edef#1{#1}%
2028   \caption@Debug{floatstyle{#2} = '#1'}%}
```

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

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

```
2029 \providecommand*\float@setstyle[2]{%
2030   \expandafter\edef\csname fst@#1\endcsname{%
2031     \expandafter\noexpand\csname fs@#2\endcsname}%}
```

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

```
2032 \providecommand*\float@dostyle[1]{%
2033   \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.)

```

2034 \providecommand*\float@ifcaption[1]{%
2035   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2036   \expandafter\@firstoftwo
2037   \else
2038     \expandafter\@secondoftwo
2039   \fi}%

2040 } } {%
2041   \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2042   \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2043 % \clearcaptionsetup{boxed}% used by the floatrow package?
2044 }

```

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

```

2045 \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’.

```

2046 \caption@ifbool{ruled}{%
2047   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
2048 }{%
2049   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2050 }
2051 \caption@undefbool{ruled}

```

2.19.2 The floatflt package

```

2052 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%

```

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

```

2053 \let\caption@ORI@floatingfigure\floatingfigure
2054 \def\floatingfigure{%
2055   \caption@floatflt{figure}%
2056   \caption@ORI@floatingfigure}%

```

`\floatingtable` Same with `\floatingtable`...

```

2057 \let\caption@ORI@floatingtable\floatingtable
2058 \def\floatingtable{%
2059   \caption@floatflt{table}%
2060 % \caption@setautoposition b%
2061   \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.

```

2062 \newcommand*\caption@floatflt[1]{%
2063   \caption@settype{#1}%
2064   \caption@clearmargin
2065   \caption@setoptions{floating#1}%
2066   \let\caption@boxrestore\@parboxrestore}%

2067 }{}

```

2.19.3 The fltpage package

```

2068 \caption@ifPackageLoaded{fltpage}[1998/10/29 v.0.3]{%
2069   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

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

2070 \renewcommand\FP@helpNote[2]{%
2071   \begingroup % save \caption@thepage
2072     \caption@pageref{#2}%
2073     \typeout{FP#1 is inserted on page \caption@thepage!}%
2074   \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}%
}

2075 \renewcommand*\FP@floatBegin[1]{%
2076   \def\@captype{#1}%
2077   \let\FP@LabelText\@empty
2078   \begin{lrbox}{\FP@floatCorpusBOX}%
2079     \caption@ifFPrefcap
2080       {\caption@freeze\relax}%
2081       {\def\label##1{\gdef\FP@LabelText{##1}\ignorespaces}%
2082       \caption@freeze*}}%

```

\FP@floatEnd **Original code:**

```

\newcommand{\FP@floatEnd}{%
  \end{lrbox}%
  \global\setbox\FP@floatCorpusBOX=\box\FP@floatCorpusBOX
  \stepcounter{FP@\@capytype C}%
  \FP@savedLabelCommand{\FP@positionLabel}%
  \FP@helpNote{\@capytype}{\FP@positionLabel}%
  \FP@float
    {\FP@positionLabel}% location label test
    {\begin{\@capytype}[p!]}
      \usebox{\FP@floatCorpusBOX}%
      \refstepcounter{\@capytype}%
      \ifthenelse{\equal{\FP@LabelText}{\@empty}}{
        {\FP@savedLabelCommand{\expandafter\protect\FP@LabelText}}}{%
      }
    \end{\@capytype}}
    {\addtocounter{\@capytype}{-1}}
    {\begin{\@capytype}[b!]}
      \ifthenelse{\equal{\FP@guide}{\@empty}}{
        {\ifthenelse{\equal{\@capytype}{figure}}{
          {\renewcommand{\fnum@figure}{\old@Fnum\ {\FP@guide}}}%
          {\renewcommand{\fnum@table}{\old@Fnum\ {\FP@guide}}}}}{%
        }
      \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{\@capytype}}}%
}%

2083 \renewcommand*\FP@floatEnd{%
2084   \end{lrbox}%
2085   \stepcounter{FP@\@capytype C}%
2086   \caption@label\FP@positionLabel
2087   \FP@helpNote\@capytype\FP@positionLabel
2088   \let\FP@ContinuedFloat\relax
2089   \FP@float
2090     {\FP@positionLabel}% location label test
2091     {\begin{\@capytype}[p!]}%
2092       \FP@ContinuedFloat
2093       \usebox{\FP@floatCorpusBOX}
2094       \caption@ifFPlistcap
2095         {\caption@refstepcounter\@capytype
2096           \expandafter\caption@makecurrent\expandafter\@capytype
2097             \expandafter{\caption@SClentry}}}%
2098         {\expandafter\captionlistentry\expandafter{\caption@SClentry}}}%
2099       \caption@@anchor\relax
2100       \ifx\FP@LabelText\@empty \else
2101         \expandafter\label\expandafter{\FP@LabelText}%
2102       \fi
2103     \end{\@capytype}%

```

```

2104     {\let\FP@ContinuedFloat\ContinuedFloat
2105     \caption@ifcompatibility
2106     {\let\caption@ContinuedFloat\@gobble}%
2107     {\caption@clearsetup{ContinuedFloat}%
2108     \caption@clearsetup{continued\@capttype}}}%
2109     {\begin\@capttype[b!}%
2110     \FP@ContinuedFloat
2111     \let\FP@savedSetfnumCommand\caption@setfnum
2112     \def\caption@setfnum##1{%
2113     \FP@savedSetfnumCommand{##1}%
2114     \ifx\FP@guide\@empty \else
2115     \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}}%
2116     \fi}%
2117     \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2118     \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2119     \caption@setoptions{FP\@capttype}%
2120     \FP@separatorCaption
2121     \caption@ifFPlistcap{}{\caption@set@bool\caption@iflof0}%
2122     \caption@defrost
2123     \end\@capttype}%
2124 }%

2125 \ForFloatingEnvironments{%
2126   \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2127 }{%
2128   \let\caption@ifFPlistcap\@undefined
2129   \let\caption@ifFPrefcap\@undefined
2130 }

```

2.19.4 The hyperref package

```

2131 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2132   \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2133     \caption@WarningNoLine{%
2134       Hyperref support is turned off\MessageBreak
2135       because hyperref has stopped early}%
2136   }{%
2137     \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`.**

```

2138   \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`.

```

2139   \renewcommand*\caption@makecurrent[2]{%
2140     \caption@makecurrentHref{#1}%
2141     \caption@Debug{hyperref current=\@currentHref}%
2142     \def\@currentlabelname{#2}}}%
2143   \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

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

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

```

2144 \renewcommand\caption@@anchor[1]{%
2145 \caption@Debug{hyperref anchor: \@currentHref}%
2146 % If we cannot have nesting, the anchor is empty.
2147 \ifHy@nesting
2148 \hyper@@anchor{\@currentHref}{#1}%
2149 \else
2150 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2151 \fi}%
2152 \g@addto@macro\caption@prepareslc{\let\caption@@anchor\@firstofone}%

```

`\caption@@ContinuedFloat`

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.

```

2153 \renewcommand*\caption@@ContinuedFloat[1]{%
2154 \ifundefined{theH#1}{}{}%
2155 \expandafter\l@addto@macro\csname theH#1\endcsname{%
2156 \alpha\c@ContinuedFloat}}}%

```

The `hypcap` option

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

```

2157 \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`.

```

2158 \def\caption@start{\caption@ifhypcap{%

```

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

```

2159 \caption@makestart\@captype
2160 \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.)

```

2161 \global\@capstarttrue
2162 \let\hc@currentHref\@currentHref
2163 \def\caption@makecurrentHref##1{%
2164 \global\@capstartfalse
2165 \global\let\@currentHref\hc@currentHref}%

```

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

```

2166 \let\caption@@anchor\@firstofone

```



```
2167 }{} }%
```

`\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 `hypertextnames=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 `hypertextnames=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.

```
2168 \newcommand*\caption@makestart[1]{%
2169   \begingroup
2170   \Hy@hypertextnamesfalse
2171 %   \gdef\@currentHlabel{}%
2172   \hyper@makecurrent{#1.caption}%
2173   \endgroup
2174   \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.

```
2175 \newcommand*\caption@startanchor[1]{%
2176   \caption@Debug{hycap anchor: #1}%
2177   \ifvmode\begingroup
2178     \@tempdima\prevdepth
2179     \nointerlineskip
2180     \vspace*{-\caption@hycapSPACE}%
2181     \caption@anchor{#1}%
2182     \vspace*{\caption@hycapSPACE}%
2183     \prevdepth\@tempdima
2184   \endgroup\else
2185     \caption@Warning{Internal Warning: Vertical TeX mode expected}%
2186     \caption@anchor{#1}%
2187   \fi}%%
```

`\caption@anchor` `\caption@anchor{⟨Href⟩}` sets a `hyperref` anchor.

```
2188 \newcommand*\caption@anchor[1]{%
2189   \ifmeasuring@ \else
2190     \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2191   \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 will not be necessary with future versions of `hyperref`.)

```
2192 \ifx\HyperRaiseLinkLength\@tempdima
2193 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2194 \else
2195 \let\caption@raisedlink\Hy@raisedlink
2196 \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.

```
2197 \def\caption@@start{%
2198 \ifundefined{hc@currentHref}{%
2199 \caption@Warning{%
2200 The option ‘hycap=true’ will be ignored for this\MessageBreak
2201 particular \string\caption}}}%
```

`\caption@freezeHref` Suppress `\caption@start` from generating a `hyperref` label and setting a `hyperref` 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`.

```
2202 \def\caption@freezeHref{%
2203 \let\caption@ORI@start\caption@start
2204 \def\caption@start{\let\caption@start\caption@ORI@start}%
2205 %
2206 % \l@addto@macro\subcaption@settypehook{%
2207 % \let\caption@@start\caption@ORI@@start}%
2208 %
2209 \global\let\caption@currentHref\@undefined
2210 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2211 %
2212 \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2213 \renewcommand*\caption@setfloatcapt{%
2214 \ifx\caption@currentHref\@undefined \else
2215 \let\caption@@anchor\@firstofone
2216 \fi
2217 \caption@ORI@setfloatcapt}}%
```

`\caption@defrostHref` If there is a frozen `\@currentHref`, we set the `hyperref` anchor here.

```
2216 \def\caption@defrostHref{%
2217 \ifx\caption@currentHref\@undefined \else
2218 \caption@startanchor\caption@currentHref
2219 \global\let\caption@currentHref\@undefined
2220 \fi}%
```

`\float@makebox` Do our own redefinition of `\float@makebox`, if it was redefined by the `hyperref` package.

```
2221 \ifundefined{HyOrg@float@makebox}{}{%
2222 \caption@Debug{%
2223 Redefining \noexpand\float@makebox (again)\@gobble}%
2224 \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
2225 \renewcommand*\float@makebox[1]{%
2226 \HyOrg@float@makebox{#1}\relax \caption@defrostHref}}%
2227 }
```

```
2228 } {} }
```

2.19.5 The hypcap package

```
2229 \caption@IfPackageLoaded{hypcap}{% v1.0
2230 \ifx\caption@start\relax \else % hyperref hasn't stopped early
```

If the `hypcap` package was loaded, we give up our own hyperlink placement algorithm and give the control over the placement to the `hypcap` package instead.

`\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 environment, at least unless a new `\capstart` will be placed.

```
2231 \let\caption@ORI@capstart\capstart % save for compatibility mode
2232 \let\capstart\caption@start
2233 \let\caption@start\relax
2234 \let\caption@@start\relax
```

`\caption@hypercapspace` Furthermore we map our `\caption@hypercapspace` to `\hypercapspace` offered by the `hypcap` package.

```
2235 \caption@set@bool\caption@ifhypcap1%
2236 \renewcommand*\caption@hypercapspace{\hypercapspace}%
2237 \fi {} }
```

2.19.6 The listings package

```
2238 \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`.

```
2239 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2240 \def\lst@MakeCaption#1{%
2241 \caption@setoptions{lstlisting}%
```

If the `position=` is set to `auto`, we take over the `captionpos=` setting from the `listings` package. Note that we won't do this otherwise, so `listings` settings like `abovecaptionskip=0pt`, `belowcaptionskip=10pt`, `captionpos=t` will *not* cause different outputs with or without the `caption` package loaded.

```
2242 \caption@setautoposition{#1}%
2243 \caption@begin{lstlisting}%
2244 \caption@ORI@lst@MakeCaption{#1}%
2245 \caption@end}%
```

`\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.

```
2246 \providecommand*\ext@lstlisting{lol}%
2247 } {} }
```

2.19.7 The longtable package

```

\LTcapttype \LTcapttype is preset to table.
2248 \providecommand*\LTcapttype{table}

2249 \caption@ifPackageLoaded{longtable}[1995/05/24 v3.14]{%
2250   \RequirePackage{ltcaption}[2007/09/01]%
2251   \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.
2252 \caption@AtBeginDocument{%
2253   \let\caption@ORI@LT@array\LT@array
2254   \renewcommand*\LT@array{%

\captionsetup for longtable:
2255   \global\let\caption@opt@@longtable\@undefined
2256   \def\captionsetup##1{\noalign{\LT@captionsetup{##1}}}%
2257   \def\LT@captionsetup##1{%
2258     \@captionsetup\@gobble[@longtable]{##1}%
2259     \global\let\caption@opt@@longtable\caption@opt@@longtable}%

\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2260   \def\@captionabovetrue{\LT@captionsetup{position=t}}%
2261   \def\@captionabovefalse{\LT@captionsetup{position=b}}%

\captionlistentry for longtable:
2262   \def\captionlistentry##1{\noalign{%
2263     \caption@listentry[\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
2264 % \caption@ifhyperc{}%
2265 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2266 % \def\hyper@makecurrent##1{%
2267 %   \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2268 %   \caption@makestart{##1}%
2269 % \let\Hy@LT@currentHlabel\@currentHlabel
2270 % \let\Hy@LT@currentHref\@currentHref
2271 % \def\hyper@makecurrent####1{%
2272 %   \let\@currentHlabel\Hy@LT@currentHlabel
2273 %   \let\@currentHref\Hy@LT@currentHref}}%
2274 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
2275 % \def\ContinuedFloat{\noalign{%
2276 %   \gdef\caption@setContinuedFloat{%
2277 %     \let\caption@resetContinuedFloat\@gobble}%
2278 %   \def\caption@setoptions####1{%
2279 %     \g@addto@macro\caption@setContinuedFloat{%
2280 %       \caption@setoptions{####1}}}%
2281 % \let\@capttype\LTcapttype
2282 % \caption@ORI@ContinuedFloat}}%

```

```

2283 %      }{%
2284 %      \def\ContinuedFloat{\noalign{%
2285 %      \caption@Error{%
2286 %      \noexpand\ContinuedFloat inside longtables\MessageBreak
2287 %      is only available with 'hypcap=true'}}}%
2288 %      }%
2289 %      \global\let\caption@setContinuedFloat\@empty
2290 \def\ContinuedFloat{\noalign{%
2291 \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2292 \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}`:

```

2293 \def\LT@c@ption#1[#2]#3{%
2294 \LT@makecaption#1{\csname fnun@\LTcapttype\endcsname}{#3}%
2295 \def\@currentlabelname{#2}%
2296 \caption@addcontentsline\LTcapttype{#2}}%

```

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

The original definition:

```

\def\LT@makecaption#1#2#3{%
  \LT@mc@l\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:

```

2297 \renewcommand\LT@makecaption[3]{%
2298 \caption@LT@make{%

```

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

```

2299 \caption@startrue#1\caption@starfalse

```

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`.)

```

2300 \caption@settype*\LTcaptiontype
2301 \ifdim\LTcapwidth=4in \else
2302 \setcaptionwidth\LTcapwidth
2303 \fi
2304 \caption@setoptions{longtable}%
2305 % \caption@setContinuedFloat
2306 \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!

```

2307 \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
2308 \caption@resetContinuedFloat\LTcaptiontype
2309 \caption@begin\LTcaptiontype
2310 \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...)

```

2311 \vskip-\ht\strutbox

```

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

```

2312 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2313 \caption@@make{#2}{#3}\endgraf
2314 \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2315 \caption@end}}%
2316 {}{}

```

2.19.8 The picinpar package

```

2317 \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.

```

2318 \long\def\figwindow[#1,#2,#3,#4] {%
2319 \caption@window{figure}%
2320 \caption@setoptions{figwindow}%
2321 \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
2322 \long\def\tabwindow[#1,#2,#3,#4] {%
2323 \caption@window{table}%
2324 \caption@setoptions{tabwindow}%
2325 \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.

```

2326 \newcommand*\caption@window[1]{%
2327   \let\caption@boxrestore\@parboxrestore
2328   \let\@makecaption\caption@make
2329   \caption@setautoposition b%
2330   \caption@settype{#1}%
2331   \caption@clearmargin}%

```

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

```

2332 \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.
2333   \ifdim\picwd=\z@
2334     \let\caption@makecurrent\@gobbletwo
2335     \let\caption@start\relax
2336     \caption@prepareslc
2337   \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 LOF]{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.

```

2338   \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2339   \if\@tempa*%
2340     \let\@tempa\@firstofone
2341   \else\if\@tempa[%
2342     \let\@tempa\@firstofone
2343   \else
2344     \let\@tempa\@empty
2345   \fi\fi
2346   \expandafter\caption\@tempa{#1}%
2347 }{}

```

2.19.9 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 `\@capytype`, 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.

```

2348 \newcommand*\piccaptiontype[1]{\def\@piccaptiontype{#1}}
2349 \caption@IfPackageLoaded{picins}{%

```

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

```

2350   \@ifundefined{@piccaptiontype}{%
2351     \caption@iftype{%
2352       \let\@piccaptiontype\@capytype
2353     }{%
2354       \def\@piccaptiontype{figure}%
2355     }%
2356   }{}%
2357   \let\@capytype\@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.

```
2358 \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.

```
2359 \let\caption@ORI\make@piccaption\make@piccaption  
2360 \def\make@piccaption{%  
2361   \let\caption@ORI\caption  
2362   \long\def\caption[##1]##2{%  
2363     \caption@freezeHref % will be defrosted in \ivparpic  
2364     \caption@settype\@piccapttype  
2365     \ifnum\c@piccaptionpos>2\relax  
2366       \caption@clearmargin  
2367     \else  
2368       \captionwidth\z@ % do not use "width=" setting  
2369     \fi  
2370     \caption@setoptions{parpic}%  
2371     \let\caption@boxrestore\@parboxrestore  
2372     \caption@setautoposition b%  
2373     \expandafter\expandafter\expandafter\caption@ORI  
2374       \expandafter\expandafter\expandafter[%  
2375       \expandafter\expandafter\expandafter{%  
2376       \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
2377   \caption@ORI@make@piccaption
2378   \let\caption\caption@ORI}%

```

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

```

2379   \let\caption@ORI@ivparpic\ivparpic
2380   \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
2381     \let\caption@ORI@noindent\noindent
2382     \def\noindent{%
2383       \caption@defrostHref
2384       \let\noindent\caption@ORI@noindent
2385       \noindent}%
2386     \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2387     \let\noindent\caption@ORI@noindent}%

2388 }{%
2389   \let\piccaptiontype\@undefined
2390 }

```

2.19.10 The rotating package

```

2391 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%

```

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

```

2392   \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
2393 % \let\@rotcaption\@undefined

```

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

```

2394   \def\rotcaptionof{%
2395     \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.)

```

2396   \long\def\@makerotcaption#1#2{%

```

```

2397 \ifdim\captionwidth=\z@
2398 \setcaptionwidth{.8\textheight}%
2399 \caption@slc{#1}{#2}{.8\vsizel}{%
2400 \let\caption@makerot\caption@@make
2401 \caption@clearmargin
2402 % \long\def\captionbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2403 % (not needed because \rotatebox uses an \hbox anyway)
2404 \let\captionbox\@secondoftwo}%
2405 \caption@set@bool\caption@ifslc0% been there, done that
2406 \fi

2407 \rotatebox{90}{\caption@makerot{#1}{#2}}%
2408 \nobreak\hspace{12pt}}%

2409 \newcommand\caption@makerot[2]{%
2410 \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%

2411 \ForFloatingEnvironments{%
2412 \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
2413 \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
2414 {}{}

```

2.19.11 The sidecap package

```

2415 \caption@ifPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
2416 \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.)

```

2417 \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`.

```

2418 \let\caption@ORI@SC@zfloat\SC@zfloat
2419 \def\SC@zfloat#1#2#3[#4]{%

```

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

```

2420 \let\caption@ORI\caption
2421 \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
2422 \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.

```

2423 \caption@settype*{#2}%
2424 \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.

```

2425 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
2426 \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.

```

2427 \let\caption@ORI@settype\caption@settype
2428 \def\caption@settype##1{% will be done in \xfloat
2429 \caption@ORI@settype*{##1}% do not change \@currentlabel
2430 \caption@setSC@justify
2431 %%% \caption@setoptions{SCfloat}%
2432 \caption@setoptions{SC\@capttype}%
2433 \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 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

```

2434 \def\caption@setSC@justify{%
2435 \caption@clearmargin
2436 \@ifundefined{SC@justify}{}{%
2437 \ifx\SC@justify\@empty \else
2438 \let\caption@hj\SC@justify
2439 \let\SC@justify\@empty
2440 \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!

```

2441 \let\caption\SC@orig@caption
2442 \def\SC@orig@caption[##1]##2{\caption@defrost}%

```

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

```

2443 \caption@setSC@justify % for compatibility mode
2444 \caption@ORI@endSC@FLOAT}%

2445 \ForFloatingEnvironments{%
2446 \newenvironment{SC#1}%
2447 {\SC@float[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@float}%
2448 \newenvironment{SC#1*}%
2449 {\SC@dblfloat[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@dblfloat}}%

2450 {}

```

2.19.12 The subfigure package

```

2451 \caption@ifPackageLoaded{subfigure}[2002/01/23 v2.1]{%

```

\sf@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.

```

2452 \def\sf@ifpositiontop{%
2453 \ifx\@capttype\@undefined
2454 \expandafter\@gobbletwo
2455 \else\ifx\@capttype\relax

```

```

2456     \expandafter\expandafter\expandafter\@gobbletwo
2457   \else
2458     \expandafter\expandafter\expandafter\sfi@if@position@top
2459   \fi\fi}

2460 \def\sfi@if@position@top{%
2461   \ifundefined{if\@captype topcap}%
2462     {\@gobbletwo}%
2463     {\@nameuse{if\@captype topcap}%
2464       \expandafter\@firstoftwo
2465     \else
2466       \expandafter\@secondoftwo
2467     \fi}}

2468 }{}

```

2.19.13 The supertabular package

```

2469 \caption@ifPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.
2470   \let\caption@ORI@tablecaption\tablecaption
2471   \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}

2472 \long\def\ST@caption#1[#2]#3{\par%
2473   \caption@settype*{#1}%
2474   \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.

2475   \def\caption@fixposition{%
2476     \caption@setposition{\if@topcaption t\else b\fi}}%

2477   \caption@beginex{#1}{#2}{#3}%
2478   \caption@boxrestore
2479   \caption@normalsize
2480   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2481   \caption@end}%

2482 }{%
2483   \let\caption@setSTposition\@undefined
2484 }

```

2.19.14 The threeparttable package

```

2485 \caption@ifpackageloaded{threeparttable}[2003/06/13 v3.0]{%
\threeparttable Unfortunately \@capttype is not set when \TPT@common will be used, so we have to
redefine \threeparttable and \measuredfigure instead.
2486 \let\caption@ORI@threeparttable\threeparttable
2487 \renewcommand*\threeparttable{%
2488 \caption@settype{table}%
2489 \caption@setposition a% ?
2490 \caption@clearmargin
2491 \caption@setoptions{threeparttable}%
2492 \caption@ORI@threeparttable}%

```

```

\measuredfigure Same here...
2493 \let\caption@ORI@measuredfigure\measuredfigure
2494 \renewcommand*\measuredfigure{%
2495 \caption@settype{figure}%
2496 \caption@setposition a% ?
2497 \caption@clearmargin
2498 \caption@setoptions{measuredfigure}%
2499 \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}

2500 \def\TPT@caption#1[#2]#3{%
2501 \gdef\TPT@docapt{%
2502 \global\let\TPT@docapt\@undefined
2503 \caption@setautoposition\caption@TPT@position
2504 \TPT@LA@caption{#1}[#2][#3]}%
2505 \ifx\TPT@hsize\@empty
2506 \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
2507 \gdef\caption@TPT@position{t}%
2508 \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2509 \else
2510 \def\caption@TPT@position{b}%
2511 \TPT@docapt
2512 \fi
2513 \ignorespaces}%
2514 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2515 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima}%

2516 {}

```

2.19.15 The wrapfig package

```

2517 \caption@ifpackageloaded{wrapfig}{% ver 3.3 (Oct 12, 1999)
\float@ifstyle \float@ifstyle{<type>}{<if-clause>}{<else-clause>}}

```

(see float package support for details)

```

2518 \providecommand*\float@ifstyle[1]{%
2519   \expandafter\ifx\csname fst@#1\endcsname\relax
2520   \expandafter\@secondoftwo
2521   \else
2522   \expandafter\@firstoftwo
2523   \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.

```

2524 \newcommand*\caption@restylewrapfloat[1]{%
2525   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2526   \csname wrap#1\endcsname
2527   \caption@Error{%
2528     For a successful cooperation of the 'wrapfig' package\MessageBreak
2529     with the 'float' package you should load the 'wrapfig'\MessageBreak
2530     package *after* the 'float' package}%
2531   \else
2532     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2533     \csname wrap#1\endcsname
2534     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
2535     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2536     \csname wrap#1\endcsname
2537   \fi}%

```

`\caption@wrapfloat`

```

2538 \newcommand*\caption@wrapfloat[1]{%
2539   \caption@settype*{#1}%
2540   \float@ifstyle{#1}{%
2541     \ifx\WF@floatstyhook\@undefined
2542     \caption@Error{%
2543       For a successful cooperation of the 'wrapfig' package\MessageBreak
2544       with the 'float' package you should use at least\MessageBreak
2545       'wrapfig' version 3.6}%
2546     \else
2547     \float@dostyle{#1}%
2548     \fi}{}%
2549   \caption@clearmargin
2550   %%% \caption@setoptions{wrapfloat}%
2551   \caption@setoptions{wrap#1}%
2552   \@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`.)

```

2553 \caption@restylewrapfloat{figure}%
2554 \caption@restylewrapfloat{table}%

2555 \ForFloatingEnvironments{%
2556   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
2557   \caption@restylewrapfloat{#1}}%
2558 \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.
2559 \ifundefined{float@restyle}{}{%
2560   \toks@=\expandafter{\float@restyle{#1}}% (env may or may not be defined)
2561   \caption@restylewrapfloat{#1}}%
2562   \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2563   \@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.
2564 \let\caption@ORI@wrapfloat\wrapfloat
2565 \def\wrapfloat#1{%
2566   \float@ifstyle{#1}{%
2567     \caption@Error{%
2568       For a successful cooperation of the 'wrapfig' package\MessageBreak
2569       with the 'float' package you should load the 'wrapfig'\MessageBreak
2570       package *right after* the 'float' package}}{%
2571     \caption@ORI@wrapfloat{#1}}%

2572 \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\bgrou \setlength\hsize{#2}%
\ifdim\hsize>\z@ \@parboxrestore \else
\setbox\z@\hbox\bgrou \let\wf@caption\caption \let\caption\wf@caption
\ignorespaces \fi}

Our code:
2573 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2574   \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2575   \global\setbox\WF@box\top\bgrou \setlength\hsize{#2}%
2576   \caption@start
2577   \ifdim\hsize>\z@ \@parboxrestore \else
2578   \setbox\z@\hbox\bgrou \let\wf@caption\caption \let\caption\wf@caption
2579   \ignorespaces \fi}%

2580 {}

```

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,
June 14, 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 \LaTeX ,
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