

The Implementation of the caption package^{*}

Axel Sommerfeldt
caption@sommerfee.de

2007/09/01

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 provides 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, like the caption package and the subfig package.

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 itself or L^AT_EX 2_ε 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

caption-eng.pdf	The English documentation
caption-rus.pdf	The Russian documentation ¹
caption-deu.pdf	The German documentation

^{*}This package has version number v3.1, last revised 2007/09/01.

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	11
1.6	Indentions	13
1.7	Styles	14
1.8	Formats	15
1.9	Label formats	16
1.10	Label separators	16
1.11	Text formats	17
1.12	Fonts	18
1.13	Justifications	19
1.13.1	The ragged2e package	20
1.14	Vertical spaces before and after captions	21
1.15	Positioning	21
1.16	Hooks	23
1.17	Lists	23
1.18	Debug option	24
1.19	Document classes & Babel support	24
1.19.1	The standard L ^A T _E X classes	24
1.19.2	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	24
1.19.3	The beamer class	26
1.19.4	The KOMA-Script classes	26
1.19.5	The NTG Dutch classes	27
1.19.6	The frenchb Babel option	27
1.19.7	The frenchle/pro package	28
1.20	Execution of options	28
1.21	Typesetting the caption	29
1.22	Bonus features	33
1.23	subfig package adaption	33
2	Main package	35
2.1	Identification	35
2.2	Loading the kernel	35
2.3	Check against incompatible packages	35
2.4	Check document class	35
2.5	Adaption to the $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes	36

2.6	Emulation of the KOMA-Script commands	36
2.7	Option for configuration files	38
2.8	Options for figure and table	38
2.9	caption v1.x compatibility options	38
2.10	caption2 v2.x compatibility options	39
2.11	Obsolete caption v3.0 options	39
2.12	ftpage package support options	40
2.13	hyperref package support options	40
2.14	Processing of options	40
2.15	Option ‘type’, \captionof and \ContinuedFloat	40
2.16	Internal helpers	43
2.17	\caption, \@caption, and \@makecaption	45
2.18	Document class & Babel package support	51
2.18.1	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	51
2.18.2	The beamer class	52
2.18.3	The KOMA-Script classes	52
2.18.4	The frenchb Babel option	52
2.18.5	The frenchle/pro package	52
2.19	Package support	53
2.19.1	The float package	55
2.19.2	The floatflt package	58
2.19.3	The ftpage package	59
2.19.4	The hyperref package	61
2.19.5	The hypcap package	64
2.19.6	The listings package	65
2.19.7	The longtable package	65
2.19.8	The picinpar package	68
2.19.9	The picins package	69
2.19.10	The rotating package	70
2.19.11	The sidecap package	71
2.19.12	The subfigure package	72
2.19.13	The supertabular package	73
2.19.14	The threeparttable package	74
2.19.15	The wrapfig package	74

1 Kernel

1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2007/09/01 v3.1 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` his 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\@firstoftwo
58 \else
59 \expandafter\expandafter\expandafter\expandafter
60 \expandafter\expandafter\expandafter\@secondoftwo
61 \fi\fi\fi}

\caption@setbool For setting and testing boolean options we offer these three helper macros:
\caption@ifbool
\caption@undefbool \caption@setbool{\langle name\rangle}{\langle value\rangle}
(with value = false/true/no/yes/off/on/0/1)
\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>}
74 \newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}

\caption@withoptargs \caption@withoptargs{<cmd>}
75 \newcommand*\caption@withoptargs[1]{%
76   \@ifstar
77   {\def\caption@tempa{*}\caption@@withoptargs#1}%
78   {\def\caption@tempa{}\caption@@withoptargs#1}}

79 \def\caption@@withoptargs#1{%
80   \@ifnextchar[%]
81   {\caption@@@withoptargs#1}%
82   {\caption@@@withoptargs#1}}

83 \def\caption@@@withoptargs#1[#2]{%
84   \l@addto@macro\caption@tempa{[#{#2}]}%
85   \caption@@withoptargs#1}

86 \def\caption@@@withoptargs#1{%
87   \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.)

88 \newcommand\caption@DoCheckCommand[2]{%
89   \begingroup
90     \let\@tempa#1%
91     #2%
92     \ifx\@tempa#1%
93       \endgroup
94       \let\caption@CheckCommand\@gobbletwo
95     \else
96       \endgroup
97     \fi}
98 \@onlypreamble\caption@DoCheckCommand

```

```

99 \let\caption@CheckCommand\caption@DoCheckCommand
100 \@onlypreamble\caption@CheckCommand

101 \newcommand*\caption@ifCheckCommand{%
102   \ifx\caption@CheckCommand\@gobbletwo
103     \let\next\@firstoftwo
104   \else
105     \let\next\@secondoftwo
106   \fi
107   \let\caption@CheckCommand\caption@DoCheckCommand
108   \next}
109 \@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.

```

110 \let\caption@begindocumenthook\@empty
111 \let\caption@@begindocumenthook\@empty

112 \def\caption@AtBeginDocument{%
113   \caption@teststar\g@addto@macro
114     \caption@@begindocumenthook\caption@begindocumenthook}
115 \@onlypreamble\caption@AtBeginDocument

116 \AtBeginDocument{%
117   \PackageInfo{caption}{Begin \noexpand\AtBeginDocument code\@gobble}%

118   \def\caption@AtBeginDocument{%
119     \ifstar{\g@addto@macro\caption@@begindocumenthook}\@firstofone}%
120   \caption@begindocumenthook
121   \let\caption@begindocumenthook\@undefined

122   \def\caption@AtBeginDocument{%
123     \ifstar\@firstofone\@firstofone}%
124   \caption@@begindocumenthook
125   \let\caption@@begindocumenthook\@undefined

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

```

1.3 Errors

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

This is mainly identical to `\PackageError{caption}{<message>}{\caption@eh}`.

```

127 \newcommand*\caption@Error[1]{%
128   \PackageError\caption@Package{#1}\caption@eh}

129 \newcommand*\caption@Package{caption}

```

`\caption@eh` At the moment we only offer this simple error message as generic helper for the user.

```

130 \newcommand*\caption@eh{%
131   If you do not understand this error, please take a closer look\MessageBreak
132   at the documentation of the '\caption@Package' package.\MessageBreak
133   \@ehc}

```

1.4 Using the keyval package

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

```
134 \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.

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

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

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

```
137 \providecommand*\@onlypreamble@key[2]{%
138   \expandafter\@onlypreamble\csname KV@#1@#2\endcsname}
139 \@onlypreamble\@onlypreamble@key
```

```
\DeclareCaptionOption \DeclareCaptionOption{<option>}{<code>}
```

```
\DeclareCaptionOption*{<option>}{<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.

```
140 \newcommand*\DeclareCaptionOption{%
141   \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
142 \@onlypreamble\DeclareCaptionOption
143 \newcommand*\caption@declareoption[2]{%
144   #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
145 \@onlypreamble\caption@declareoption
```

```
\captionsetup \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`.

```
146 \newcommand*\captionsetup{%
147   \caption@teststar\@captionsetup\@gobble\@firstofone}
148 \newcommand*\@captionsetup[1]{%
149   \ifnextchar[{\caption@setup@options#1}\caption@setup}
150 \newcommand*\caption@setup{\caption@setkeys{caption}}
151 \def\caption@setup@options#1[#2]{%
152   % #1 is \@gobble in star form, and \@firstofone otherwise
153   #1{\caption@addtooptlist{#2}}%
154   \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 a warning.

```
155 \newcommand*\clearcaptionsetup{%
156   \caption@teststar\@clearcaptionsetup\@gobble\@firstofone}
157 \newcommand*\@clearcaptionsetup[1]{%
158   \let\caption@tempa#1%
159   \@testopt\@clearcaptionsetup{}}
```



```

160 \def\@@clearcaptionsetup[#1]#2{%
161   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
162   {\caption@tempa{\PackageWarning{caption}{Option list `#2' undefined}}}%
163   {\ifx,#1,%
164     \caption@clearsetup{#2}%
165   \else
166     \caption@@removefromsetup{#1}{#2}%
167   \fi}}

168 \newcommand*\caption@clearsetup[1]{%
169   \caption@removefromoptlist{#1}%
170   \@nameundef{caption@opt@#1}}

171 \newcommand*\caption@removefromsetup{%
172   \let\caption@tempa\@gobble
173   \caption@@removefromsetup}

174 \newcommand*\caption@@removefromsetup[2]{%
175   \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
176   \expandafter\let\csname caption@opt@#2\endcsname\@undefined
177   \def\@tempb##1=##2\@nil{##1}%
178   \edef\@tempc{#1}%
179   \@for\@tempa:=\@tempa\do{%
180     \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
181     \ifx\@tempd\@tempc
182       \let\caption@tempa\@gobble
183     \else
184       \expandafter\expandafter\expandafter\caption@l@addto@list
185       \expandafter\csname caption@opt@#2\endcsname\expandafter\endcsname
186       \expandafter{\@tempa}%
187     \fi}%
188   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
189   {\caption@removefromoptlist{#2}}}%
190   \caption@tempa{%
191     \PackageWarning{caption}{%
192       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>*.

```

193 \newcommand*\showcaptionsetup[2][\@firstofone]{%
194   \GenericWarning{}{%
195     #1 Caption Info: KV list on `#2'\MessageBreak
196     #1 Caption Data: \@ifundefined{caption@opt@#2}{%
197       -none-%
198     }{%
199       {\expandafter\expandafter\expandafter\strip@prefix
200         \expandafter\meaning\csname caption@opt@#2\endcsname}%
201     }}

202 \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.

```

203 \newcommand*\caption@setoptions[1]{%
204 \caption@Debug{options=#1}%
205 \expandafter\let\expandafter\@tempa\csname caption@opt@#1\endcsname
206 \ifx\@tempa\relax \else
207 \let\caption@iftypewarning\@firstoftwo % = true
208 \expandafter\caption@setup\expandafter{\@tempa}%
209 \caption@clearsetup{#1}%
210 \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”).

211 \newcommand*\caption@addtooptlist[1]{%
212 \ifundefined{caption@opt@#1@lineno}{%
213 \caption@dooptlist\caption@g@addto@list{#1}%
214 \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
215 }{}}

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

219 \newcommand*\caption@dooptlist[2]{%
220 \begingroup
221 \edef\@tempa{#2}\@onelevel@sanitize\@tempa
222 \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
223 \endgroup}

224 \AtEndDocument{%
225 \caption@for@list\caption@optlist{%
226 \PackageWarningNoLine{caption}{%
227 Unused \string\captionsetup[1]
228 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.

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

230 \long\def\caption@@setkeys[#1]#2#3{%
231 \ifx\caption@keydepth\@empty
232 \let\caption@ORI@KV@errx@KV@errx
233 \let\caption@ORI@KV@err@KV@err
234 \let\KV@errx\caption@Error
235 \let\KV@err\caption@Error
236 \fi
237 \l@addto@macro\caption@keydepth.%

238 \def\caption@Package{#1}%
239 \caption@Debug{\protect\setkeys{#2}{#3}}%
240 \setkeys{#2}{#3}%
241 \def\caption@Package{caption}%

```

```

242 \edef\caption@keydepth{\expandafter\@cdr\caption@keydepth\@nil}%
243 \ifx\caption@keydepth\@empty
244   \let\KV@errx\caption@ORI@KV@errx
245   \let\KV@err\caption@ORI@KV@err
246 \fi}
247 \let\caption@keydepth\@empty

\caption@ExecuteOptions \caption@ExecuteOptions{\family}{\key-values}
We execute our options using the keyval interface, so we use this one instead of
\ExecuteOptions offered by LATEX 2ε.

248 \newcommand*\caption@ExecuteOptions[2]{%
249   \@expandtwoargs\caption@setkeys{#1}{#2}}%
250 \@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.)

251 \newcommand*\caption@ProcessOptions{%
252   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
253 \@onlypreamble\caption@ProcessOptions

254 \newcommand*\caption@@ProcessOptions[2]{%
255   \let\@tempc\relax
256   \let\caption@tempa\@empty
257   #1{% \@firstofone -or- \@gobble
258     \for\CurrentOption:=\@classoptionslist\do{%
259       \ifundefined{KV@#2@\CurrentOption}}{%
260         \ifundefined{KV@#2@\CurrentOption @default}}{%
261           \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
262         }%
263         \PackageInfo{#2}{Global option '\CurrentOption' processed}%
264         \edef\caption@tempa{\caption@tempa,\CurrentOption,%
265           \@expandtwoargs\@removeelement\CurrentOption
266           \@unusedoptionlist\@unusedoptionlist
267         }%
268       }%
269     }%
270     \let\CurrentOption\@empty
271   }%
272   \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
273   \AtEndOfPackage{\let\@unprocessedoptions\relax}
274 \@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@`.

```

275 \newdimen\captionmargin

```

```

276 \newdimen\captionmargin@
277 \newdimen\captionwidth

278 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
279 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
280 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
281 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}

282 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
283 \DeclareCaptionOption{maxmargin}{\caption@setoptcmd\caption@maxmargin{#1}}

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

284 \newcommand*\setcaptionmargin{%
285   \caption@teststar\caption@setmargin\@gobble\@firstofone}

286 \newcommand*\caption@setmargin[2]{%
287   #1{\captionwidth\z@}%
288   \caption@@setmargin#2,#2,\@nil}

289 \def\caption@@setmargin#1,#2,#3\@nil{%
290   \setlength\captionmargin@{#2}%
291   \setlength\captionmargin{#1}%
292   \addtolength\captionmargin@{-\captionmargin}}

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

293 \newcommand*\setcaptionwidth{%
294   \captionmargin\z@
295   \captionmargin@\z@
296   \setlength\captionwidth}

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

297 \newcommand*\caption@count{0}

\caption@newlabel This command is a modified version of \newlabel from LATEX2ε. It will be written
to the .aux file to pass label information from one run to another. (We use it inside
\caption@ifoddpage and \caption@ragged.)

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

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

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

\caption@label This command is a modified version of \label from LATEX2ε. It will be used inside
\caption@ifoddpage and \FP@helpNote.

300 \newcommand*\caption@label[1]{%
301   \protected@write\@auxout{\let\caption@thepage\relax}%
302     {\string\caption@newlabel{#1}{\caption@thepage}}}

```

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

```

303 \newcommand*\caption@pageref[1]{%
304   \expandafter\ifx\csname caption@r@#1\endcsname\relax
305     \G@refundefinedtrue % => 'There are undefined references.'
306     \PackageWarning{caption}{Reference on page \thepage \space undefined}%
307   \else
308     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
309   \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.

```

310 \newcommand*\caption@ifoddpage{%
311   \caption@iftwoside{%
312     \begingroup
313       \@tempcnta\caption@count
314       \advance\@tempcnta\@ne
315       \xdef\caption@count{\the\@tempcnta}%
316     \endgroup
317     \caption@label\caption@count
318     \caption@pageref\caption@count
319     \ifodd\caption@thepage
320       \let\caption@ifoddpage\@firstoftwo
321     \else
322       \let\caption@ifoddpage\@secondoftwo
323     \fi
324   }{\let\caption@ifoddpage\@firstoftwo}%
325   \caption@ifoddpage}

```

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

```

326 \newcommand*\caption@setoptcmd[2]{%
327   \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 328 \newdimen\caption@indent
\caption@hangindent 329 \newdimen\caption@parindent
330 \newdimen\caption@hangindent

331 \DeclareCaptionOption{indent}[\leftmargini]{%
332   \setlength\caption@indent{#1}}% obsolete!
333 \DeclareCaptionOption{indentation}[\leftmargini]{%
334   \setlength\caption@indent{#1}}
335 \DeclareCaptionOption{parindent}{%
336   \setlength\caption@parindent{#1}}
337 \DeclareCaptionOption{hangindent}{%
338   \setlength\caption@hangindent{#1}}
339 \DeclareCaptionOption{parskip}{%
340   \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

```

341 \ifundefined{scr@caption}{}{%
342   \let\caption@KV@parindent\KV@caption@parindent
343   \DeclareCaptionOption{parindent}[]{%
344     \ifx,#1,%
345       \caption@Debug{Option 'parindent' ignored}%
346     \else
347       \caption@KV@parindent{#1}%
348     \fi}%
349   \let\caption@KV@parskip\KV@caption@parskip
350   \DeclareCaptionOption{parskip}[]{%
351     \ifx,#1,%
352       \caption@Debug{Option 'parskip' ignored}%
353     \else
354       \caption@KV@parskip{#1}%
355     \fi}%
356 }

```

1.7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
357 \newcommand*\DeclareCaptionStyle[1]{%
358   \@testopt{\caption@declarestyle{#1}}{}%
359   \@onlypreamble\DeclareCaptionStyle
360   \def\caption@declarestyle#1[#2]#3{%
361     \global\@namedef{caption@sls@#1}{#2}%
362     \global\@namedef{caption@sty@#1}{#3}%
363     \@onlypreamble\caption@declarestyle
364   }
364   \DeclareCaptionOption{style}{\caption@setstyle{#1}}
365   \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.

```

366 \newcommand*\caption@setstyle{%
367   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

```

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!)

```

368 \newcommand*\caption@@setstyle[2]{%
369   \ifundefined{caption@sty@#2}%
370     {#1{\caption@Error{Undefined style `#2'}}}%
371     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
372       \ifx\caption@setstyle@flag\@undefined
373         \let\caption@setstyle@flag\caption@setstyle
374         \let\caption@iftypewarning\@firstoftwo % = true
375         \caption@resetstyle
376         \expandafter\caption@setup\expandafter{\caption@sty}%
377         \let\caption@setstyle@flag\@undefined
378       \else
379         \expandafter\caption@setup\expandafter{\caption@sty}%
380       \fi
381     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
382     \expandafter\caption@l@addto@list\expandafter\caption@opt@SingleLine
383     \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!*

```

384 \newcommand*\caption@resetstyle{%
385   \caption@setup{%
386     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
387     justification=justified,font=,labelfont=,textfont=,%
388     margin=0pt,minmargin=0,maxmargin=0,%
389     indent=0pt,parindent=0pt,hangindent=0pt,%
390     singlelinecheck=1,strut=1}%
391   \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.

```

392 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
393 \DeclareCaptionStyle{default}[indent=0pt,justification=centering,format=plain]{%
394   format=default,labelsep=default,textformat=default,%
395   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 `indent=` option.

```

396 \newcommand\DeclareCaptionFormat{%
397   \caption@teststar\caption@declareformat\@gobble\@firstofone}
398 \@onlypreamble\DeclareCaptionFormat

399 \newcommand\caption@declareformat[3]{%
400   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
401   \global\long\@namedef{caption@fmt@#2}##1##2##3{#3}}
402 \@onlypreamble\caption@declareformat

403 \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`).

```
404 \newcommand*\caption@setformat[1]{%
405   \ifundefined{caption@fmt@#1}%
406     {\caption@Error{Undefined format `#1'}}%
407     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
408       \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}
```

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

```
409 \DeclareCaptionFormat{plain}{#1#2#3\par}
410 \DeclareCaptionFormat{hang}{%
411   \@hangfrom{#1#2}%
412   \advance\caption@parindent\hangindent
413   \advance\caption@hangindent\hangindent
414   \caption@par
415   #3\par}
```

‘default’ usually maps to ‘plain’.

```
416 \def\caption@fmt@default{\caption@fmt@plain}
417 \def\caption@ifh@default{\caption@ifh@plain}
```

1.9 Label formats

`\DeclareCaptionLabelFormat` `\DeclareCaptionLabelFormat{<name>}{<code with #1, #2, and #3>}`

```
418 \newcommand*\DeclareCaptionLabelFormat[2]{%
419   \global\@namedef{caption@lfmt@#1}##1##2##3{#2}}
420 \onlypreamble\DeclareCaptionLabelFormat

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

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

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

```
422 \newcommand*\caption@setlabelformat[1]{%
423   \ifundefined{caption@lfmt@#1}%
424     {\caption@Error{Undefined label format `#1'}}%
425     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}
```

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

```
426 \DeclareCaptionLabelFormat{empty}{}
427 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
428 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace} (#2)}
```

‘default’ usually maps to ‘simple’.

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

1.10 Label separators

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

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

```
430 \newcommand\DeclareCaptionLabelSeparator{%
431   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
432 \onlypreamble\DeclareCaptionLabelSeparator
```



```

433 \newcommand\caption@declarelableseparator[3]{%
434   \global\expandafter\let\csname caption@ifl@#2\endcsname#1%
435   \global\long\@namedef{caption@lsep@#2}{#3}}
436 \@onlypreamble\caption@declarelableseparator

437 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
438 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}

```

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

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

```

439 \newcommand*\caption@setlabelseparator[1]{%
440   \@ifundefined{caption@lsep@#1}%
441   {\caption@Error{Undefined label separator `#1'}}%
442   {\expandafter\let\expandafter\caption@ifl@#1\endcsname
443    \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’.

```

444 \DeclareCaptionLabelSeparator{none}{}
445 \DeclareCaptionLabelSeparator{colon}{:}
446 \DeclareCaptionLabelSeparator{period}{.}
447 \DeclareCaptionLabelSeparator{space}{ }
448 \DeclareCaptionLabelSeparator*{quad}{\quad}
449 \DeclareCaptionLabelSeparator*{newline}{\\}
450 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

```

‘default’ usually maps to ‘colon’.

```

451 \def\caption@lsep@default{\caption@lsep@colon}
452 \def\caption@ifl@default{\caption@ifl@colon}

```

1.11 Text formats

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

```

453 \newcommand*\DeclareCaptionTextFormat[2]{%
454   \global\long\@namedef{caption@tfmt@#1}##1{#2}}
455 \@onlypreamble\DeclareCaptionTextFormat

456 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
457 \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`).

```

458 \newcommand*\caption@settextformat[1]{%
459   \@ifundefined{caption@tfmt@#1}%
460   {\caption@Error{Undefined text format `#1'}}%
461   {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

462 \DeclareCaptionTextFormat{simple}{#1}
463 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

1.12 Fonts

```

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

DeclareDefaultCaptionFont \DeclareDefaultCaptionFont{<name>}{<code>}
468 \newcommand*\DeclareDefaultCaptionFont[2]{%
469   \global\@namedef{caption#1@default}{#2}}
470 \@onlypreamble\DeclareDefaultCaptionFont

471 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
472 \DeclareDefaultCaptionFont{font}{}

473 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
474 \DeclareDefaultCaptionFont{labelfont}{}

475 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
476 \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.
477 \newcommand*\caption@setfont[2]{%
478   \begingroup
479     \let\caption@fnt\@empty
480     \define@key{caption@fnt}{default}[]{%
481       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
482     \caption@setkeys[caption]{caption@fnt}{#2}%
483     \global\let\caption@tempa\caption@fnt
484   \endgroup
485   \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.
486 \newcommand*\caption@font{%
487   \caption@teststar\caption@@font\@firstofone
488     {\caption@setkeys[caption]{caption@fnt}}
489 \newcommand*\caption@@font[2]{%
490   \begingroup
491     \def\caption@fnt{\endgroup}%
492     #1{#2}%
493   \caption@fnt}

```

These are the pre-defined font code snippets.

```

494 \DeclareCaptionFont{normalcolor}{\normalcolor}
495 \DeclareCaptionFont{color}{\color{#1}}

496 \DeclareCaptionFont{normalfont}{\normalfont}
497 \DeclareCaptionFont{up}{\upshape}
498 \DeclareCaptionFont{it}{\itshape}
499 \DeclareCaptionFont{sl}{\slshape}

```

```

500 \DeclareCaptionFont{sc}{\scshape}
501 \DeclareCaptionFont{md}{\mdseries}
502 \DeclareCaptionFont{bf}{\bfseries}
503 \DeclareCaptionFont{rm}{\rmfamily}
504 \DeclareCaptionFont{sf}{\sffamily}
505 \DeclareCaptionFont{tt}{\ttfamily}

506 \DeclareCaptionFont{scriptsize}{\scriptsize}
507 \DeclareCaptionFont{footnotesize}{\footnotesize}
508 \DeclareCaptionFont{small}{\small}
509 \DeclareCaptionFont{normalsize}{\normalsize}
510 \DeclareCaptionFont{large}{\large}
511 \DeclareCaptionFont{Large}{\Large}

512 \DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespace}% normally 1
513 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
514 \DeclareCaptionFont{doublespacing}{\doublespacing}
515 \DeclareCaptionFont{stretch}{\setstretch{#1}}
516 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }

517 %\DeclareCaptionFont{normal}{%
518 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
519 \DeclareCaptionFont{normal}{%
520   \caption@font*{%
521     \KV@caption@fnt@normalcolor\@unused
522     \KV@caption@fnt@normalfont\@unused
523     \KV@caption@fnt@normalsize\@unused
524     \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.

```

525 \providecommand\captionsize{}

526 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
527 \DeclareDefaultCaptionFont{size}{}

```

1.13 Justifications

```

\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
528 \newcommand*\DeclareCaptionJustification[2]{%
529   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
530   \DeclareCaptionFont{#1}{#2}}
531 \@onlypreamble\DeclareCaptionJustification

\DeclareDefaultCaptionJustification{<code>}
532 \newcommand*\DeclareDefaultCaptionJustification[1]{%
533   \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
534   \DeclareDefaultCaptionFont{@hj}{#1}}
535 \@onlypreamble\DeclareDefaultCaptionJustification

536 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
537 \DeclareDefaultCaptionJustification{}

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

```

These are the pre-defined justification code snippets.

```
539 \DeclareCaptionJustification{justified}{}
540 \DeclareCaptionJustification{centering}{\centering}
541 \DeclareCaptionJustification{centerfirst}{\centerfirst}
542 \DeclareCaptionJustification{centerlast}{\centerlast}
543 \DeclareCaptionJustification{raggedleft}{\raggedleft}
544 \DeclareCaptionJustification{raggedright}{\raggedright}
```

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

```
545 \providecommand\centerfirst{%
546   \let\\\@centercr
547   \edef\caption@normaladjust{%
548     \leftskip\the\leftskip
549     \rightskip\the\rightskip
550     \parfillskip\the\parfillskip\relax}%
551   \leftskip\z@\@plus -1fil%
552   \rightskip\z@\@plus 1fil%
553   \parfillskip\z@skip
554   \noindent\hskip\z@\@plus 2fil%
555   \@setpar{\@@par\@restorepar\caption@normaladjust}}
```

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

```
556 \providecommand\centerlast{%
557   \let\\\@centercr
558   \leftskip\z@\@plus 1fil%
559   \rightskip\z@\@plus -1fil%
560   \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.

```
561 \DeclareCaptionJustification{Centering}{%
562   \caption@ragged\Centering\centering}
563 \DeclareCaptionJustification{RaggedLeft}{%
564   \caption@ragged\RaggedLeft\raggedleft}
565 \DeclareCaptionJustification{RaggedRight}{%
566   \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.

```
567 \newcommand*\caption@ragged{%
568   \caption@Debug{We need ragged2e}%
569   \protected@write\@auxout{}{\string\caption@newlabel{ragged2e}{}%
570     \global\let\caption@ragged\caption@ragged}
571   \caption@ragged}
```

```

572 \caption@AtBeginDocument{%
573   \@ifundefined{caption@r@ragged2e}{%
574     \newcommand*\caption@@ragged{%
575       \PackageWarning{caption}{%
576         'ragged2e' support has been changed.\MessageBreak
577         Rerun to get captions right}%
578     \global\let\caption@ragged\@secondoftwo % suppress further warnings
579     \caption@ragged}%
580   }{%
581     \caption@Debug{We load ragged2e}%
582     \IfFileExists{ragged2e.sty}{%
583       \RequirePackage{ragged2e}%
584       \let\caption@@ragged\@firstoftwo
585     }{%
586       \newcommand*\caption@@ragged[2]{%
587         \@ifundefined{caption\string#1}{%
588           \PackageWarning{caption}{%
589             'ragged2e' package not loaded, therefore\MessageBreak
590             substituting \string#2 for \string#1\MessageBreak}%
591           \global\@namedef{caption\string#1}}{}%
592         #2}%
593     }%
594   }}

```

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.

```

595 \@ifundefined{abovecaptionskip}{%
596   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
597 \@ifundefined{belowcaptionskip}{%
598   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

599 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
600 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
601 \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.

```

602 \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.*

```

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

```

```

605 \let\caption@position\caption@defaultpos
606 }\caption@ifinlist{#1}{t,top,above}{%
607 \let\caption@position\@firstoftwo
608 }\caption@ifinlist{#1}{b,bottom,below}{%
609 \let\caption@position\@secondoftwo
610 }\caption@ifinlist{#1}{a,auto}{%
611 \let\caption@position\@undefined
612 }{%
613 \caption@Error{Undefined position `#1'}%
614 }}}}

```

`\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.

```

615 %\caption@setdefaultpos{a}% default = auto
616 \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.)

```

617 \newcommand*\caption@iftop{%
618 \ifx\caption@position\@undefined
619 \let\caption@position\@secondoftwo
620 % = \caption@setposition b%
621 \fi
622 \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.

```

623 \newcommand*\caption@fixposition{%
624 \ifx\caption@position\@undefined
625 \caption@autoposition
626 \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.

```

627 \newcommand*\caption@autoposition{%
628 \ifvmode
629 \edef\caption@tempa{\the\prevdepth}%
630 \caption@Debug{\protect\prevdepth=\caption@tempa}%
631 \ifdim\prevdepth>-\p@
632 \let\caption@position\@secondoftwo
633 \else
634 \let\caption@position\@firstoftwo

```

```

635     \fi
636 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
637 \else
638     \caption@Debug{no \protect\prevdepth}%
639     \let\caption@position\@secondoftwo
640 %   = \caption@setposition b%
641 \fi}

```

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

```

642 \newcommand*\caption@setautoposition[1]{%
643   \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`.

```

644 \newcommand*\caption@beginhook{}
645 \newcommand*\caption@endhook{}
646 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
647 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

1.17 Lists

```

648 \DeclareCaptionOption{listof}{\caption@set@bool\caption@iflof{#1}}
\DeclareCaptionListFormat \DeclareCaptionListFormat {<name>}{<code with #1 and #2>}
649 \newcommand*\DeclareCaptionListFormat [2]{%
650   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
651 \@onlypreamble\DeclareCaptionListFormat

```

```

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

```

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

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

```

653 \newcommand*\caption@setlistofformat[1]{%
654   \@ifundefined{caption@lstfmt@#1}%
655     {\caption@Error{Undefined list-of format `#1'}}%
656     {\expandafter\let\expandafter\caption@lstfmt
657       \csname caption@lstfmt@#1\endcsname}}

```

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

```

658 \DeclareCaptionListFormat{empty}{}
659 \DeclareCaptionListFormat{simple}{#1#2}
660 \DeclareCaptionListFormat{parens}{#1(#2)}
661 \DeclareCaptionListFormat{subsimple}{#2}
662 \DeclareCaptionListFormat{subparens}{(#2)}
663 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}

```

1.18 Debug option

```
664 \DeclareCaptionOption{debug}[1]{%
665   \caption@set@bool\caption@ifdebug{#1}%
666   \let\caption@iftypewarning\caption@ifdebug
667   \caption@ifdebug{%
668     \def\caption@Debug{\PackageInfo{caption}}%
669   }{%
670     \let\caption@Debug\@gobble
671   }}

672 \DeclareOption{debug}{\setkeys{caption}{debug}}
673 \setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

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

1.19.1 The standard L^AT_EX classes

```
675 \caption@CheckCommand\@makecaption{%
676   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
677   \long\def\@makecaption#1#2{%
678     \vskip\abovecaptionskip
679     \sbox\@tempboxa{#1: #2}%
680     \ifdim \wd\@tempboxa >\hsize
681       #1: #2\par
682     \else
683       \global \@minipagefalse
684       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
685     \fi
686     \vskip\belowcaptionskip}}
```

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

```
687 \ifundefined{@captionheadfont}{}{%
688   \caption@CheckCommand\@makecaption{%
689     % amsart|amsproc|amsbook [2004/08/06 v2.20]
690     \long\def\@makecaption#1#2{%
691       \setbox\@tempboxa\vbox{\color@setgroup
692         \advance\hsize-2\captionindent\noindent
693         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
694           {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
695         \unskip\kern-2\captionindent\par
696         \global\setbox\@ne\lastbox\color@endgroup}%
697       \ifhbox\@ne % the normal case
698         \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
699       \fi
700       \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
701         \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
702       \else % tempboxa contained more than one line
703         \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
704           \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
705       \fi
706       \ifnum\@tempcnta<64 % if the float IS a figure...
707         \addvspace\abovecaptionskip
708         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%

```



```

709     \else % if the float IS NOT a figure...
710         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
711         \nobreak
712         \vskip\belowcaptionskip
713     \fi
714 \relax
715 }}

716 \caption@CheckCommand\@makecaption{%
717 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
718 \long\def\@makecaption#1#2{%
719     \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
720     \setbox\@tempboxa\vbox{\color@setgroup
721         \advance\hsize-2\captionindent\noindent
722         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
723             {\@cdr#2\@nil}{\@addpunct{.}}\@captionfont\upshape\enspace#2}%
724     \unskip\kern-2\captionindent\par
725     \global\setbox\@ne\lastbox\color@endgroup}%
726 \ifhbox\@ne % the normal case
727     \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
728 \fi
729 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
730     \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
731     \@tempdima\wd\@ne\advance\@tempdima-\captionindent
732     \wd\@ne\@tempdima
733 \else % tempboxa contained more than one line
734     \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
735         \unvbox\@tempboxa\parskip\z@skip
736         \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
737 \fi
738 \ifnum\@tempcnta<64 % if the float IS a figure...
739     \addvspace\abovecaptionskip
740     \noindent\kern\captionindent\box\@ne
741 \else % if the float IS NOT a figure...
742     \noindent\kern\captionindent\box\@ne
743     \nobreak
744     \vskip\belowcaptionskip
745 \fi
746 \relax
747 }}

748 \let\captionmargin\captionindent
749 \begingroup\edef\@tempa{\endgroup
750     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
751     {margin=\the\captionmargin
752     \@ifundefined{smf@makecaption}{}{,maxmargin=.1\linewidth}}}
753 \@tempa
754 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
755 \DeclareCaptionLabelSeparator{default}{.\enspace}
756 \DeclareDefaultCaptionFont{font}{\@captionfont}
757 \DeclareDefaultCaptionFont{labelfont}{\@captionheadfont}
758 \DeclareDefaultCaptionFont{textfont}{\@captionfont\upshape}
759 \captionsetup*[figure]{position=b}
760 \captionsetup*[table]{position=t}

761 }

```

1.19.3 The beamer class

```

762 \@ifclassloaded{beamer}{%
763   \caption@CheckCommand\beamer@makecaption{%
764     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
765     \long\def\beamer@makecaption#1#2{%
766       \def\insertcaptionname{\csname#1name\endcsname}%
767       \def\insertcaptionnumber{\csname the#1\endcsname}%
768       \def\insertcaption{#2}%
769       \nobreak\vskip\abovecaptionskip\nobreak
770       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
771       \ifdim \wd\@tempboxa >\hsize
772         \usebeamertemplate**{caption}\par
773       \else
774         \global \@minipagefalse
775         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
776       \fi
777       \nobreak\vskip\belowcaptionskip\nobreak}%
778   \DeclareCaptionLabelFormat{default}{#1}
779   \DeclareDefaultCaptionJustification{\raggedright}
780   \DeclareDefaultCaptionFont{font}{%
781     \usebeamerfont*{caption}%
782     \usebeamercolor[fg]{caption}}%
783   \DeclareDefaultCaptionFont{labelfont}{%
784     \usebeamercolor[fg]{caption name}%
785     \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.)

```

786   \defbeamertemplate{caption}{caption3}{%
787     \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
788   \DeclareOption{beamer}{%
789     % \usebeamertemplate**{caption} will set font
790     \DeclareDefaultCaptionFont{font}{}%
791     \setbeamertemplate{caption}[caption3]}
792 {}

```

1.19.4 The KOMA-Script classes

```

793 \@ifundefined{scr@caption}{}{%
794   \caption@CheckCommand\@makecaption{%
795     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
796     \long\def\@makecaption#1#2{%
797       \if@captionabove
798         \vskip\belowcaptionskip
799       \else
800         \vskip\abovecaptionskip
801       \fi
802       \@@makecaption\@firstofone{#1}{#2}%
803       \if@captionabove
804         \vskip\abovecaptionskip
805       \else
806         \vskip\belowcaptionskip

```

```

807     \fi}}
808 \DeclareCaptionFormat{default}{%
809   \ifdofullc@p
810     \caption@fmt@hang{#1}{#2}{#3}%
811   \else
812     #1#2%
813     \ifdim\cap@indent<\z@
814       \par
815       \noindent\hspace*{-\cap@indent}%
816     \else\if@capbreak
817       \par
818     \fi\fi
819     #3\par
820   \fi}
821 \DeclareCaptionLabelSeparator{default}{\captionformat}
822 \DeclareDefaultCaptionFont{font}{\scr@fnt@caption}
823 \DeclareDefaultCaptionFont{labelfont}{\scr@fnt@captionlabel}
824 }

```

1.19.5 The NTG Dutch classes

```

825 \@ifundefined{CaptionFonts}{}{%
826   \caption@CheckCommand\@makecaption{%
827     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
828     \long\def\@makecaption#1#2{%
829       \vskip\abovecaptionskip
830       \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
831       \ifdim \wd\@tempboxa >\hsize
832         {\CaptionLabelFont#1:} \CaptionTextFont#2\par
833       \else
834         \global \@minipagefalse
835         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
836       \fi
837       \vskip\belowcaptionskip}}
838 \DeclareDefaultCaptionFont{labelfont}{\CaptionLabelFont}
839 \DeclareDefaultCaptionFont{textfont}{\CaptionTextFont}
840 }

```

1.19.6 The frenchb Babel option

```

841 \@ifundefined{FB@makecaption}{}{%
842   \caption@CheckCommand\@makecaption{%
843     % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
844     \long\def\@makecaption#1#2{%
845       \vskip\abovecaptionskip
846       \sbox\@tempboxa{#1\CaptionSeparator #2}%
847       \ifdim \wd\@tempboxa >\hsize
848         #1\CaptionSeparator #2\par
849       \else
850         \global \@minipagefalse
851         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
852       \fi
853       \vskip\belowcaptionskip}}

```

```

854 \ifx\@makecaption\STD@makecaption
855   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
856   \def\caption@frenchb{% supress frenchb warning
857     \let\STD@makecaption\@makecaption
858     \let\FB@makecaption\@makecaption}
859 \else
860   \ifx\@makecaption\@undefined\else
861     \PackageInfo{caption}{%
862       The definition of \protect\@makecaption\space
863       has been changed,\MessageBreak
864       frenchb will NOT customize it}%
865   \fi
866 \fi
867 }

```

1.19.7 The frenchle/pro package

```

868 \@ifundefined{frenchTeXmods}{}{%
869   \caption@CheckCommand\@makecaption{%
870     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
871     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
872     \def\@makecaption#1#2{%
873       \ifFTY%
874         \def\@secondofmany##1##2\void{##2}%
875         \def\@tempa{\@secondofmany#2\void}%
876         \ifx\@tempa\empty%
877           \let\captionseparator\empty%
878         \fi%
879         \@mcORI{#1}{\relax\captionfont{#2}}%
880       \else
881         \@mcORI{#1}{#2}%
882       \fi}}
883   \caption@CheckCommand\@makecaption{%
884     % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
885     \def\@makecaption#1#2{%
886       \ifFTY%
887         \def\@secondofmany##1##2\void{##2}%
888         \protected@edef\@tempa{\@secondofmany#2\void}%
889         \ifx\@tempa\empty%
890           \let\captionseparator\empty%
891         \fi%
892         \@mcORI{#1}{\relax\captionfont{#2}}%
893       \else
894         \@mcORI{#1}{#2}%
895       \fi}}
896   \DeclareDefaultCaptionFont{textfont}{\itshape}%
897   \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
898 }

```

1.20 Execution of options

```

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

```

```

901 \ProcessOptions*
902 \caption@ifcheckcommand{%
903   \caption@setbool{documentclass}{1}%
904 }{%
905   \caption@setbool{documentclass}{0}%
906   \PackageInfo{caption}{%
907     Unknown document class (or package), \MessageBreak
908     standard defaults will be used}%
909   \caption@debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
910 }

```

1.21 Typesetting the caption

`\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.

```

911 \newcommand*\caption@addcontentsline[2]{%
912   \caption@iflof%
913     {\def\@tempa{#2}}%
914     {\let\@tempa\@empty}%
915   \ifx\@tempa\@empty \else
916     {\let\\\space
917       \addcontentsline{\csname ext@#1\endcsname}{#1}%
918       {\protect\numberline
919         {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
920         {\ignorespaces #2}}}%
921   \fi}

```

`\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`.)

```
922 \newif\ifcaption@star
```

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

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

```

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

```

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

Typesets the caption.

```

925 \newcommand\caption@make[2]{%
926   \caption@@make{\caption@lfmt{#1}{#2}}{\theContinuedFloat}}

```

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

```

927 \newcommand\caption@@make[2]{%
928   \begingroup
929   \caption@beginhook

```

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

```

930   \ifx\caption@maxmargin\undefined \else
931     \ifdim\captionmargin>\caption@maxmargin\relax
932       \captionmargin\caption@maxmargin\relax
933   \fi
934   \fi

```

```

935 \ifx\caption@minmargin\@undefined \else
936 \ifdim\captionmargin<\caption@minmargin\relax
937 \captionmargin\caption@minmargin\relax
938 \fi
939 \fi

```

Special single-line treatment (option singlelinecheck=)

```

940 \caption@ifslc{\caption@ifempty@list\caption@opt@SingleLine{}}{%
941 \caption@slc{#1}{#2}\captionwidth\relax}}}%

```

Typeset the left margin (option margin=)

```

942 \caption@calcmargin
943 \@tempdima\captionmargin
944 \ifdim\captionmargin@=\z@ \else
945 \caption@ifoddpage{\advance\@tempdima\captionmargin@}%
946 \fi
947 \caption@ifh{\advance\@tempdima\caption@indent}%
948 \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*`.

```

949 \@tempdima\captionwidth
950 \caption@ifh{\advance\@tempdima-\caption@indent}%
951 \captionbox\@tempdima{%

```

Typeset the indentation (option indentation=)

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

```

952 \caption@ifh{%
953 \ifdim\caption@indent=\z@
954 \leavevmode
955 \else
956 \hskip-\caption@indent
957 \fi}%

```

Typeset the caption itself and close the `\captionbox`

```

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

```

Typeset the right margin (option margin=)

```

959 \@tempdima\captionmargin
960 \ifdim\captionmargin@=\z@ \else
961 \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
962 \fi
963 \hspace\@tempdima
964 \caption@endhook
965 \endgroup
966 \global\caption@starfalse}

```

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

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

```

967 \newcommand*\caption@calcmargin{%
968 \ifdim\captionwidth=\z@
969 \captionwidth\linewidth
970 \advance\captionwidth by -2\captionmargin
971 \advance\captionwidth by -\captionmargin@

```

```

972 \else
973   \captionmargin\linewidth
974   \advance\captionmargin by -\captionwidth
975   \divide\captionmargin by 2
976   \captionmargin@z@
977 \fi

978 \caption@Debug{%
979   \string\hsize=\the\hsize,
980   \string\linewidth=\the\linewidth,\MessageBreak
981   \string\leftmargin=\the\leftmargin,
982   \string\rightmargin=\the\rightmargin,\MessageBreak
983   \string\margin=\the\captionmargin,
984   \string\margin@=\the\captionmargin@,
985   \string\width=\the\captionwidth}%
986 }

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.
987 \newcommand\caption@slc[4]{%
988   \caption@Debug{Begin SLC}%
989   \begingroup
990   \expandafter\caption@setup\expandafter{\caption@opt@SingleLine}%
991   \let\caption@hj\empty
992   \caption@calcmargin % calculate #3 if necessary
993   \caption@prepareslc
994   \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
995   \ifdim\wd\@tempboxa>#3%
996     \endgroup
997   \else
998     \endgroup
999     \expandafter\caption@setup\expandafter{\caption@opt@SingleLine}%
1000    #4%
1001  \fi
1002  \caption@Debug{End SLC}}

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1003 \newcommand*\caption@prepareslc{%
1004   \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1005   \def\label{\caption@withoptargs\@gobbletwo}%
1006   \def\stepcounter##1{\advance\cname c@##1\endcname\@ne\relax}%
1007   \let\refstepcounter\stepcounter\let\H@refstepcounter\stepcounter}

\captionbox \captionbox{<width>}{<contents>}
This macro defines the box which surrounds the caption paragraph.
1008 \newcommand*\captionbox{\parbox[b]}

\caption@@@make \caption@@@make{<caption label>}{<caption text>}
This one finally typesets the caption paragraph, without margin and indentation.
1009 \newcommand\caption@@@make[2]{%

```

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

```
1010 \sbox\@tempboxa{#1}%
1011 \ifdim\wd\@tempboxa=\z@
1012 \let\caption@lsep\relax
1013 % \@capbreakfalse
1014 \fi
```

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

```
1015 \caption@ifempty{#2}{%
1016 \let\caption@lsep\relax
1017 % \@capbreakfalse
1018 % \let\caption@ifstrut\@secondoftwo
1019 }%
```

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

```
1020 \setpar{\@par\caption@@par}\caption@@par
```

Finally typeset the caption.

```
1021 \caption@hj\captionsize\captionfont\caption@fmt
1022 {\ifcaption@star\else\captionlabelfont#1}\fi}%
1023 {\ifcaption@star\else\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1024 {\captiontextfont
1025 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1026 \nobreak\hskip\z@skip % enable hyphenation
1027 \caption@tfmt{#2}%
1028 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1029 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1030 \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.

```
1031 \newcommand\caption@ifempty[1]{%
1032 \def\caption@tempa{#1}%
1033 \def\caption@tempb{\ignorespaces}%
1034 \ifx\caption@tempa\caption@tempb
1035 \let\caption@tempa\@empty
1036 \fi
1037 \ifx\caption@tempa\@empty
1038 \let\caption@ifempty\@secondoftwo
1039 \else
1040 \let\caption@ifempty\@gobbletwo
1041 \fi
1042 \caption@ifempty\@unused}
```

```
\caption@@par \caption@@par
```

This command will be executed with every `\par` inside the caption.

```
1043 \newcommand*\caption@@par{%
1044 \parindent\caption@parindent\hangindent\caption@hangindent}%

```


1.22 Bonus features

```

\DeclareFloatingEnvironment{environment name}
1045 \providecommand*\DeclareFloatingEnvironment[1]{%
1046   \@ifundefined{c@chapter}%
1047     {\newcounter{#1}%
1048      \@namedef{the#1}{\arabic{#1}}%
1049      \@namedef{theH#1}{\arabic{#1}}}%
1050     {\newcounter{#1}[chapter]%
1051      \@namedef{the#1}{\ifnum\c@chapter>\z@ \thechapter.\fi \arabic{#1}}%
1052      \@namedef{theH#1}{\theHchapter.\arabic{#1}}}%
1053   \@namedef{fps@#1}{tbp}%
1054   \@ifundefined{c@float@type}{%
1055     \newcounter{float@type}%
1056     \setcounter{float@type}{\@ifundefined{c@figure}14}{}}%
1057   \expandafter\xdef\csname ftype@#1\endcsname{\the\value{float@type}}%
1058   \addtocounter{float@type}{\value{float@type}}%
1059   \@namedef{ext@#1}{lo#1}%
1060   \@namedef{fnum@#1}{\@nameuse{#1name}\nobreakspace\@nameuse{the#1}}%
1061   \newenvironment{#1}{\@float{#1}}{\end@float}%
1062   \newenvironment{#1*}{\@dblfloat{#1}}{\end@dblfloat}%
1063   \expandafter\newcommand\csname listof#1s\endcsname{\beginngroup
1064     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1065     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1066     \let\caption@ORI@starttoc\starttoc
1067     \renewcommand*\@starttoc[1]{%
1068       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1069     \listoffigures
1070     \endgroup}%
1071   \@ifundefined{l@figure}%
1072     {\@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1073     {\expandafter\let\csname l@#1\endcsname\l@figure}%
1074   \expandafter\newcommand\csname #1name\endcsname{}%
1075   \edef\@tempa{%
1076     \def\noexpand\@tempa{\expandafter\noexpand\@car#1\@nil}}%
1077   \uppercase\expandafter{\@tempa}%
1078   \edef\@tempb{%
1079     \noexpand\g@addto@macro\noexpand\@tempa{\expandafter\@cdr#1\@nil}}%
1080   \@tempb
1081   \expandafter\let\csname #1name\endcsname\@tempa
1082   \expandafter\newcommand\csname list#1name\endcsname{}%
1083   \expandafter\xdef\csname list#1name\endcsname{List of \@tempa s}%
1084   \@cons\caption@floatlist{{#1}}%
1085 \onlypreamble\DeclareFloatingEnvironment
1086 \newcommand*\caption@floatlist{}

```

1.23 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.

```

1087 \caption@AtBeginDocument{%
1088   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1089   \ifx\subref\@tempa

```

```

1090 \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1091 \let\caption@setfloattype\@gobble
1092 % \let\sf@ifinlist\caption@ifinlist
1093 \let\sf@ifpositiontop\caption@iftop
1094 \def\@dottedxxxline#1#2#3#4#5#6#7{%
1095 \begingroup
1096 \caption@setfloattype{#1}%
1097 \caption@setoptions{subfloat}%
1098 \caption@setoptions{sub#1}%
1099 \ifnum #3>\@nameuse{c@#2depth}\else
1100 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1101 \fi
1102 \endgroup}%
1103 \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.

```

1104 \DeclareRobustCommand*\subref{\@ifstar\sf@subref\sf@subref}%
1105 \caption@removefromoptlist{subfloat}%
1106 \caption@removefromoptlist{subfigure}%
1107 \caption@removefromoptlist{subtable}%
1108 \fi
1109 \let\sf@NEW@subfloat\@undefined}
1110 \def\sf@NEW@subfloat{%
1111 \begingroup
1112 \caption@setfloattype\@capttype
1113 \sf@ifpositiontop{%
1114 \maincaptiontoptrue
1115 }{%
1116 \maincaptiontopfalse
1117 }%
1118 \caption@setoptions{subfloat}%
1119 \caption@setoptions{sub\@capttype}%
1120 \let\sf@oldlabel=\label
1121 \let\label=\subfloat@label
1122 \ifmaincaptiontop\else
1123 \advance\@nameuse{c@\@capttype}\@ne
1124 \fi
1125 \refstepcounter{sub\@capttype}%
1126 \setcounter{sub\@capttype}{\value{sub\@capttype}}%
1127 \ifnextchar [% %] match left bracket
1128 {\sf@subfloat}%
1129 {\sf@subfloat[\@empty]}}%

```

2 Main package

2.1 Identification

```
1130 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1131 \ProvidesPackage{caption}[2007/09/01 v3.1 Customising captions (AR)]

\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.

1132 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1133 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1134 \RequirePackage{caption3}[2007/09/01] % needs v3.1 or newer
```

2.3 Check against incompatible packages

```
1135 \@ifpackageloaded{caption2}{%
1136   \caption@Error{%
1137     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1138     the (current) caption package}%
1139   \endinput
1140 }{}

1141 \caption@AtBeginDocument{%
1142   \@ifpackageloaded{ftcap}{%
1143     \caption@IncompatiblePackage{ftcap}{}%
1144   }{}%
1145   \@ifpackageloaded{nonfloat}{%
1146     \caption@IncompatiblePackage{nonfloat}{. %
1147       Furthermore you could use \string\captionof\MessageBreak
1148       instead of \noexpand\figcaption or \string\tabcaption}%
1149   }{}%
1150   \@ifpackageloaded{topcapt}{%
1151     \caption@IncompatiblePackage{topcapt}{}%
1152   }{}%
1153 }

\caption@Incompatible Issue an ‘incompatible’ warning.

1154 \newcommand*\caption@IncompatiblePackage[2]{%
1155   \PackageWarningNoLine{caption}{%
1156     Usage of the #1 package together with the\MessageBreak
1157     caption package is strongly *not* recommended.\MessageBreak
1158     Instead of loading the #1 package you should\MessageBreak
1159     use the caption package option\MessageBreak
1160     \space\space`tableposition=top'\MessageBreak
1161     #2}}

1162 \@onlypreamble\caption@IncompatiblePackage
```

2.4 Check document class

```
1163 \caption@ifbool{documentclass}{}{}%
1164   \PackageWarningNoLine{caption}{%
1165     Unsupported document class (or package) detected,\MessageBreak
```

```

1166     usage of the caption package is *not* recommended}%
1167   \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1168 }

```

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

```

1169 \@ifundefined{@captionheadfont}{}{}%
1170   \caption@Info{AMS or SMF document class}%
1171   \setlength\belowcaptionskip{0pt}% set to 3pc by AMS class
1172 }

```

2.6 Emulation of the KOMA-Script commands

```

1173 \@ifundefined{scr@caption}{}{}%
1174   \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 1175   \def\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1176   \def\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1177   \if@tablecaptionabove
1178     \caption@AtBeginDocument{%
1179       \def\caption@tempa#1{%
1180         \ifpackageloaded{#1}{\caption@Error{%
1181           You can't use the global option\MessageBreak
1182           \space\space`tablecaptionabove'\MessageBreak
1183           and use the #1 package, too}}{}}}%
1184       \caption@tempa{ftcap}%
1185       \caption@tempa{nonfloat}%
1186       \caption@tempa{topcapt}}}%
1187   \@tablecaptionabovetrue
1188   \else
1189     \@tablecaptionabovefalse
1190   \fi

\onelinecaptionstrue
\onelinecaptionfalse 1191   \def\onelinecaptionstrue{\captionsetup{singlelinecheck=1}}
1192   \def\onelinecaptionfalse{\captionsetup{singlelinecheck=0}}

1193   \ifonelinecaptions
1194     \onelinecaptionstrue
1195   \else
1196     \onelinecaptionfalse
1197   \fi

\@captionabovetrue
\@captionabovefalse 1198   \def\@captionabovetrue{\captionsetup{position=t}}
1199   \def\@captionabovefalse{\captionsetup{position=b}}

\setcapindent
1200   \let\caption@KOMA@setcapindent\@setcapindent
1201   \renewcommand*\@setcapindent[1]{%
1202     \caption@KOMA@setcapindent{#1}\caption@setcapindent}

```

```

1203 \let\caption@KOMA@@setcapindent\@@setcapindent
1204 \renewcommand*\@@setcapindent[1]{%
1205   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1206 \newcommand*\caption@setcapindent{%
1207   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1208 \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.)
1209 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1210   \csname\string\setcapwidth\endcsname
1211 \@namedef{\string\setcapwidth}[1]#2{%
1212   \caption@KOMA@setcapwidth[#1]{#2}\caption@setcapwidth{#1}}
1213 \newcommand*\caption@setcapwidth[1]{%
1214   \ifx\#1\\\else
1215     \ifundefined{cap@margin}{}{%
1216       \def\@tempa{captionbeside}%
1217       \ifx\@tempa\@currentenv\else\PackageWarning{caption}{%
1218         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1219       \fi}%
1220   \fi
1221   \captionsetup{width=\cap@width}}

1222 \def\caption@tempa{\hsize}%
1223 \ifx\caption@tempa\cap@width \else
1224   \caption@setcapwidth{?}
1225 \fi

\setcapmargin
1226 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1227   \csname\string\@setcapmargin\endcsname
1228 \@namedef{\string\@setcapmargin}[1]#2{%
1229   \caption@KOMA@setcapmargin[#1]{#2}\caption@setcapmargin}
1230 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1231   \csname\string\@@setcapmargin\endcsname
1232 \@namedef{\string\@@setcapmargin}[1]#2{%
1233   \caption@KOMA@@setcapmargin[#1]{#2}\caption@setcapmargin}
1234 \newcommand*\caption@setcapmargin{%
1235   \begingroup
1236     \let\onelinecaptionsfalse\relax
1237     \def\@twoside{0}%
1238     \def\if@twoside{\def\@twoside{1}\iffalse}%
1239     \cap@margin
1240     \def\@tempa{\endgroup}%
1241     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1242       \def\hspace##1##{\@firstofone}%
1243       \edef\@tempa{\endgroup
1244         \noexpand\captionsetup{%
1245           twoside=\@twoside,singlelinecheck=0,%
1246           margin={\cap@left,\cap@right}}}%
1247     \fi\fi
1248   \@tempa}

```

```

1249 \ifx\cap@margin\relax \else
1250 \caption@setcapmargin
1251 \fi
1252 }

```

2.7 Option for configuration files

```

1253 \DeclareCaptionOption*{config}[caption]{%
1254 \InputIfFileExists{#1.cfg}%
1255 {\typeout{*** Local configuration file #1.cfg used ***}}%
1256 {\PackageWarning{caption}{Configuration file #1.cfg not found}}}

```

2.8 Options for figure and table

```

1257 \DeclareCaptionOption{figureposition}{%
1258 \captionsetup*[figure]{position=#1}}
1259 \@onlypreamble@key{caption}{figureposition}

1260 \DeclareCaptionOption{tableposition}{%
1261 \caption@AtBeginDocument{%
1262 \def\caption@tempa##1{%
1263 \ifpackageloaded{##1}{\caption@Error{%
1264 You can't use the caption package option\MessageBreak
1265 \space\space`tableposition=#1'\MessageBreak
1266 and use the ##1 package, too}}{}}}%
1267 \caption@tempa{ftcap}%
1268 \caption@tempa{nonfloat}%
1269 \caption@tempa{topcapt}}}%
1270 \captionsetup*[table]{position=#1}}
1271 \@onlypreamble@key{caption}{tableposition}

1272 \DeclareCaptionOption{figurename}{\caption@SetFloatName{figure}{#1}}
1273 \@onlypreamble@key{caption}{figurename}
1274 \DeclareCaptionOption{tablename}{\caption@SetFloatName{table}{#1}}
1275 \@onlypreamble@key{caption}{tablename}

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

1276 \newcommand*\caption@SetFloatName[2]{%
1277 \begingroup
1278 \def\@tempa{\endgroup\@firstofone}%
1279 \ifundefined{language\@tempa}{\@tempa}%
1280 \ifundefined{captions\language\@tempa}{\@tempa}%
1281 \def\@tempa{\endgroup
1282 \expandafter\addto\csname captions\language\@tempa\endcsname}}}%
1283 \@tempa{\@namedef{#1name}{#2}}}%
1284 \@onlypreamble\caption@SetFloatName

```

2.9 caption v1.x compatibility options

```

1285 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1286 \@onlypreamble@key{caption}{compatibility}

1287 \DeclareCaptionOption*{normal}[]{%
1288 \caption@setformat{normal}%
1289 \caption@setjustification{justified}}
1290 \DeclareCaptionOption*{isu}[]{%

```

```

1291 \caption@setformat{hang}%
1292 \caption@setjustification{justified}}
1293 \DeclareCaptionOption*{hang}[]{%
1294 \caption@setformat{hang}%
1295 \caption@setjustification{justified}}
1296 \DeclareCaptionOption*{center}[]{%
1297 \caption@setformat{normal}%
1298 \caption@setjustification{centering}}
1299 \DeclareCaptionOption*{anne}[]{%
1300 \caption@setformat{normal}%
1301 \caption@setjustification{centerlast}}
1302 \DeclareCaptionOption*{centerlast}[]{%
1303 \caption@setformat{normal}%
1304 \caption@setjustification{centerlast}}

1305 \DeclareCaptionOption*{scriptsize}[]{\def\captionfont{\scriptsize}}
1306 \DeclareCaptionOption*{footnotesize}[]{\def\captionfont{\footnotesize}}
1307 \DeclareCaptionOption*{small}[]{\def\captionfont{\small}}
1308 \DeclareCaptionOption*{normalsize}[]{\def\captionfont{\normalsize}}
1309 \DeclareCaptionOption*{large}[]{\def\captionfont{\large}}
1310 \DeclareCaptionOption*{Large}[]{\def\captionfont{\Large}}

1311 \DeclareCaptionOption*{up}[]{\l@addto@macro\captionlabelfont\upshape}
1312 \DeclareCaptionOption*{it}[]{\l@addto@macro\captionlabelfont\itshape}
1313 \DeclareCaptionOption*{sl}[]{\l@addto@macro\captionlabelfont\slshape}
1314 \DeclareCaptionOption*{sc}[]{\l@addto@macro\captionlabelfont\scshape}
1315 \DeclareCaptionOption*{md}[]{\l@addto@macro\captionlabelfont\mdseries}
1316 \DeclareCaptionOption*{bf}[]{\l@addto@macro\captionlabelfont\bfseries}
1317 \DeclareCaptionOption*{rm}[]{\l@addto@macro\captionlabelfont\rmfamily}
1318 \DeclareCaptionOption*{sf}[]{\l@addto@macro\captionlabelfont\sffamily}
1319 \DeclareCaptionOption*{tt}[]{\l@addto@macro\captionlabelfont\ttfamily}

1320 \DeclareCaptionOption*{nooneline}[]{\caption@setbool{slc}{0}}

1321 \caption@setbool{ruled}{0}
1322 \DeclareCaptionOption*{ruled}[]{\caption@setbool{ruled}{1}}

```

2.10 caption2 v2.x compatibility options

```

1323 \DeclareCaptionOption*{flushleft}[]{%
1324 \caption@setformat{normal}%
1325 \caption@setjustification{raggedright}}
1326 \DeclareCaptionOption*{flushright}[]{%
1327 \caption@setformat{normal}%
1328 \caption@setjustification{raggedleft}}

1329 \DeclareCaptionOption*{oneline}[]{\caption@setbool{slc}{1}}

1330 \DeclareCaptionOption*{ignoreLTcapwidth}[]{%
1331 \PackageWarning{caption}{Obsolete option 'ignoreLTcapwidth' ignored}}

```

2.11 Obsolete caption v3.0 options

```

1332 \DeclareCaptionOption*{caption}{%
1333 \caption@setbool{temp}{#1}%
1334 \caption@ifbool{temp}{}{%
1335 \caption@Error{%
1336 The package option 'caption=#1' is obsolete.\MessageBreak
1337 Please pass this option to the subfig package instead\MessageBreak

```

```
1338         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.

```
1339 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1340 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1341 \@onlypreamble@key{caption}{FPlist}
1342 \@onlypreamble@key{caption}{FPref}

1343 \newcommand*\caption@setFPoption[2]{%
1344   \edef\@tempa{\@car#2\@nil}%
1345   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1346 \@onlypreamble\caption@setFPoption

1347 \captionsetup{FPlist=caption,FPref=figure}
```

2.13 hyperref package support options

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

```
1348 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1349 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}

1350 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

2.14 Processing of options

```
1351 \caption@ProcessOptions*{caption}
```

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

```
1352 \caption@AtBeginDocument{%
1353   \DeclareCaptionOption{type}{\caption@settype{#1}}
1354   \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).

```
1355 \newcommand*\caption@settype{%
1356   \caption@@settype{}}

1357 \newcommand*\caption@@settype[1]{%
1358   \caption@teststar{\caption@@@settype{#1}}{\@gobble\@firstofone}}
```



```

1359 \newcommand*\caption@@@settype[3]{%
1360 % #1 = "" or "sub"
1361 % #2 = \@gobble in star form, \@firstofone otherwise
1362 % #3 = <type>, e.g. "figure" or "table"
1363 \caption@Debug{type=#3}%
1364 \caption@checkgrouplevel{#1}{\captionsetup{#1type#2\@gobble*=#3}}%
1365 \@ifundefined{@#1capttype}{}{%
1366 \edef\@tempa{#3}%
1367 \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1368 \caption@iftypewarning{%
1369 \PackageWarning{caption}{%
1370 Caption type was already set to `#3'\MessageBreak}}{}%
1371 \fi}%
1372 \expandafter\edef\csname @#1capttype\endcsname{#3}%
1373 \let\caption@iftypewarning\caption@ifdebug
1374 \@nameuse{#1caption@settypehook}%
1375 \expandafter\caption@setoptions\csname @#1capttype\endcsname
1376 \let\caption@ifrefstepcounter\@secondoftwo
1377 #2{% \@gobble or \@firstofone
1378 \let\@currentlabel\caption@undefinedlabel
1379 \let\@currentHlabel\@undefined
1380 \caption@start}}

```

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

```
1381 \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.

```

1382 \newcommand*\caption@iftype{%
1383 \@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)

```

1384 \begingroup\expandafter\expandafter\expandafter\endgroup
1385 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1386 \caption@Debug{TeX engine: TeX}
1387 \let\caption@checkgrouplevel\@gobbletwo
1388 \else
1389 \caption@Debug{TeX engine: e-TeX}
1390 \newcommand*\caption@checkgrouplevel[2]{%
1391 \@ifundefined{#1caption@grouplevel}{%
1392 \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}}{}%
1393 \ifnum\currentgrouplevel>\caption@grouplevel\relax
1394 \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1395 \the\currentgrouplevel}%
1396 \else
1397 \caption@Error{\string#2 outside group or environment}%
1398 \fi
1399 }{}%
1400 \fi

```

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

Note: All commands used have to be robust here.

```
1401 \newcommand*\caption@undefinedlabel{%
1402   \emph{Please place \caption@cs{label} \textbf{after} \caption@cs{caption}}}%
1403 \DeclareRobustCommand\caption@cs[1]{\texttt{\char'\#\1}}
```

```
\captionof \captionof{<type>}[<lst.entry>]{<heading>}
\captionof* [<lst.entry>]{<heading>}
1404 \def\captionof{\caption@teststar\caption@of{\caption*}\caption}
1405 \newcommand*\caption@of[2]{%
1406   \let\caption@setfloatcapt\@firstofone
1407   \caption@settype*{#2}#1}
```

```
\captionlistentry \captionlistentry[<float type>]{<list entry>}
1408 \newcommand*\captionlistentry{%
1409   \@testopt\caption@listentry\@capttype}
1410 \def\caption@listentry[#1]#2{%
1411   \caption@refstepcounter{#1}%
1412   \caption@makecurrent{#1}{#2}%
1413   \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.
Note: The definition of \ContinuedFloat itself is compatible to the one inside the subfig package.
```

```
1414 \def\ContinuedFloat{%
1415   \caption@iftype
1416     {\addtocounter\@capttype\m@ne
1417     \caption@ContinuedFloat\@capttype}%
1418     {\caption@Error{\noexpand\ContinuedFloat outside float}}}%
1419 \def\caption@ContinuedFloat#1{%
1420   \edef\@tempa{#1}%
1421   \ifx\@tempa\caption@CFtype \else
1422     \caption@Error{Continued `#1' after '\caption@CFtype'}%
1423   \fi
1424   \stepcounter{ContinuedFloat}%
1425   \caption@@ContinuedFloat{#1}%
1426   \caption@setoptions{ContinuedFloat}%
1427   \caption@setoptions{continued#1}%
1428   \let\caption@resetContinuedFloat\@gobble}
1429 \newcounter{ContinuedFloat}
1430 \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.)
1431 \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`.)

```

1432 \newcommand*\caption@resetContinuedFloat[1]{%
1433   \@stpelt{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.

```

1434 \newcommand*\caption@refstepcounter[1]{%
1435   \caption@resetContinuedFloat{#1}%
1436   \caption@@refstepcounter{#1}%
1437   \let\caption@ifrefstepcounter\@firstoftwo}

1438 \newcommand*\caption@@refstepcounter{\refstepcounter}
1439 \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[]{}`).

```

1440 \newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1441 \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`)

```

1442 \newcommand*\caption@begin[1]{%
1443   \begingroup
1444     \caption@setfnum{#1}%
1445     \caption@fixposition
1446     \global\let\caption@fixedposition\caption@position}

```

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

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

```

1447 \newcommand\caption@beginex[3]{%
1448   \caption@begin{#1}%
1449   \caption@addcontentsline{#1}{#2}%
1450   \caption@ifempty{#3}{} }

```

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

```

1451 \newcommand*\caption@end{%
1452   \endgroup
1453   \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.
1454 \newcommand*\caption@setfnum[1]{%
1455   \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1456   \@namedef{fnum@#1}{\caption@fnum{#1}}%
1457   \fi}

```

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

```

1458 \caption@AtBeginDocument*{%
1459   \@ifundefined{sf@ifpositiontop}{}{%
1460     \let\caption@kernel@addcontentsline\caption@addcontentsline
1461     \renewcommand*\caption@addcontentsline[2]{%
1462       \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
1463       \caption@kernel@addcontentsline{#1}{#2}%
1464       \sf@ifpositiontop{}{\@listsubcaptions{#1}}}%
1465   }}

```

`\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`.

```

1466 \newcommand*\caption@boxrestore{%
1467   \let\if@nobreak\iffalse
1468   \let\if@noskipsec\iffalse
1469   \let\par\@par
1470   \let\-\@dischyph
1471   \let'\@acci\let'\@accii\let\=\@acciii
1472   \parindent\z@ \parskip\z@skip
1473   \everypar{}%
1474   \linewidth\hsize
1475   \@totalleftmargin\z@
1476   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1477   \parfillskip\@flushglue \lineskip\normallineskip
1478   \baselineskip\normalbaselineskip
1479   \sloppy
1480   \let\\\@normalcr
1481 }

```

`\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).
1482 \newcommand*\caption@normalsize{%
1483   \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.
1484 \let\caption@setfloatcapt\@firstofone

\caption@makecurrent All these are needed for support of the hyperref package.
\caption@makestart 1485 \newcommand*\caption@makecurrent[2]{}
\caption@anchor 1486 \let\caption@makestart\@gobble
\caption@start 1487 \let\caption@anchor\@firstofone
\caption@@start 1488 \let\caption@start\relax
\caption@freezeHref 1489 \let\caption@@start\relax
\caption@defrostHref 1490 \let\caption@freezeHref\relax
1491 \let\caption@defrostHref\relax
1492 \g@addto@macro\caption@prepareslc{\let\caption@anchor\@firstofone}
```

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 it's own definition of `\caption*`, was quite common these days.

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

```

1493 \caption@AtBeginDocument{%
1494   \caption@setbool{incompatible}{0}%
1495   \caption@CheckCommand\caption{%
1496     % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1497     \def\caption{%
1498       \ifx\@capttype\@undefined
1499         \@latex@error{\noexpand\caption outside float}\@ehd
1500         \expandafter\@gobble
1501       \else
1502         \refstepcounter\@capttype
1503         \expandafter\@firstofone
1504       \fi
1505       {\@dblarg{\@caption\@capttype}}%
1506     }%
1507   \caption@CheckCommand\caption{%
1508     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1509     \def\caption{
```

```

1510     \ifx\@capttype\@undefined
1511         \@latex@error{\noexpand\caption outside figure or table}\@ehd
1512         \expandafter\@gobble
1513     \else
1514         \refstepcounter\@capttype
1515         \expandafter\@firstofone
1516     \fi
1517     {\@dblarg{\@caption\@capttype}}%
1518 } }%

1519 \caption@CheckCommand\caption{%
1520 % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1521 \renewcommand\caption{%
1522     \ifx\@capttype\@undefined
1523         \@latex@error{\noexpand\caption outside float}\@ehd
1524         \expandafter\@gobble
1525     \else
1526         \refstepcounter\@capttype
1527         \let\@tempf\@caption
1528         \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1529             \expandafter\expandafter\let
1530                 \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1531         \fi
1532     \fi
1533     \@dblarg{\@tempf\@capttype}}}%

1534 \caption@CheckCommand\caption{%
1535 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1536 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1537 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1538 \def\caption{%
1539     \ifx\@capttype\@undefined
1540         \@latex@error{\noexpand\caption outside float}\@ehd
1541         \expandafter\@gobble
1542     \else
1543         \H@refstepcounter\@capttype
1544         \@ifundefined{fst@\@capttype}{%
1545             \let\Hy@tempa\@caption
1546         }{%
1547             \let\Hy@tempa\Hy@float@caption
1548         }%
1549         \expandafter\@firstofone
1550     \fi
1551     {\@dblarg{\Hy@tempa\@capttype}}%
1552 } }%

1553 \caption@CheckCommand\caption{%
1554 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1555 \def\caption{%
1556     \ifx\@capttype\@undefined
1557         \@latex@error{\noexpand\caption outside float}\@ehd
1558         \expandafter\@gobble
1559     \else
1560         \H@refstepcounter\@capttype
1561         \let\Hy@tempa\@caption
1562         \@ifundefined{float@caption}{%

```

```

1563         }{%
1564         \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
1565         \let\Hy@tempa\Hy@float@caption
1566         \fi
1567     }%
1568     \expandafter\@firstofone
1569     \fi
1570     {\@dblarg{\Hy@tempa\@capttype}}%
1571 }}%

1572 \caption@ifcheckcommand{}{%
1573     \caption@info{%
1574         Incompatible package detected (regarding \string\caption).\MessageBreak
1575         \string\caption\space=\space\meaning\caption}%
1576     \caption@setbool{incompatible}{1}}%

1577 \caption@checkcommand\@caption{%
1578     % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1579     \long\def\@caption#1[#2]#3{%
1580         \par
1581         \addcontentsline{\csname ext@#1\endcsname}{#1}%
1582         {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1583         \begingroup
1584         \@parboxrestore
1585         \if@minipage
1586             \setminipage
1587         \fi
1588         \normalsize
1589         \makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1590     \endgroup}}%

1591 \caption@checkcommand\@caption{%
1592     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1593     \long\def\@caption#1[#2]#3{% second argument ignored
1594         \par\nobreak
1595         \begingroup
1596         \@parboxrestore
1597         \if@minipage
1598             \setminipage
1599         \fi
1600         \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
1601     \endgroup}}%

1602 % \caption@checkcommand\float@caption{%
1603 %     % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1604 %     \long\def\float@caption#1[#2]#3{%
1605 %         \addcontentsline{\@nameuse{ext@#1}}{#1}%
1606 %         {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
1607 %         \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
1608 %         \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
1609 %         \@ifnextchar[{\float@ccon}{\egroup}}%
1610 %     \long\def\float@ccon[#1]{#1\par\egroup}}%

1611 \caption@checkcommand\@caption{%
1612     % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1613     \long\def\@caption#1[#2]#3{%
1614         \hyper@makecurrent{\@capttype}%

```

```

1615 \def\@currentlabelname{#2}%
1616 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1617 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1618 }%
1619 \begingroup
1620 \parboxrestore
1621 \if@minipage
1622 \setminipage
1623 \fi
1624 \normalsize
1625 \@makecaption{\csname fnum@#1\endcsname}{%
1626 \ignorespaces
1627 \ifHy@nesting
1628 \hyper@@anchor{\@currentHref}{#3}%
1629 \else
1630 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1631 \fi
1632 }%
1633 \par
1634 \endgroup
1635 }%

1636 \caption@CheckCommand\@caption{%
1637 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1638 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1639 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1640 \long\def\@caption#1[#2]#3{%
1641 \expandafter\ifx\csname if@capstart\expandafter\endcsname
1642 \csname iftrue\endcsname
1643 \global\let\@currentHref\hc@currentHref
1644 \else
1645 \hyper@makecurrent{\@cuptype}%
1646 \fi
1647 \def\@currentlabelname{#2}%
1648 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1649 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1650 }%
1651 \begingroup
1652 \parboxrestore
1653 \if@minipage
1654 \setminipage
1655 \fi
1656 \normalsize
1657 \expandafter\ifx\csname if@capstart\expandafter\endcsname
1658 \csname iftrue\endcsname
1659 \global\@capstartfalse
1660 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1661 \else
1662 \@makecaption{\csname fnum@#1\endcsname}{%
1663 \ignorespaces
1664 \ifHy@nesting
1665 \hyper@@anchor{\@currentHref}{#3}%
1666 \else
1667 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1668 \fi

```



```

1669         }%
1670         \fi
1671         \par
1672     \endgroup
1673 } }%

1674 \caption@CheckCommand\@caption{%
1675     % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1676     \long\def\@caption#1[#2]{%
1677         \def\@currentlabelname{#2}%
1678         \NR@@caption{#1}[{#2}]%
1679     } }%

1680 \caption@CheckCommand\@caption{%
1681     % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1682     \long\def\@caption#1[#2]#3{%
1683         \ifundefined{if#1topcap}%
1684             {\subfig@oldcaption{#1}[{#2}]{#3}}%
1685             {\@nameuse{if#1topcap}%
1686                 \@listsubcaptions{#1}%
1687                 \subfig@oldcaption{#1}[{#2}]{#3}%
1688             \else
1689                 \subfig@oldcaption{#1}[{#2}]{#3}%
1690                 \@listsubcaptions{#1}%
1691             \fi}}}%

1692 \caption@CheckCommand\@caption{%
1693     % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1694     \def\@caption{\caption@}%
1695     \long\def\@caption#1[#2]#3{%
1696         \ifundefined{caption@setfloattype}%
1697             \caption@settype
1698             \caption@setfloattype
1699             \@capttype
1700         \sf@ifpositiontop{%
1701             \@listsubcaptions{#1}%
1702             \sf@old@caption{#1}[{#2}]{#3}%
1703         }{%
1704             \sf@old@caption{#1}[{#2}]{#3}%
1705             \@listsubcaptions{#1}%
1706         } }%
1707     }%

1708 \caption@ifCheckCommand{}{%
1709     \caption@Info{%
1710         Incompatible package detected (regarding \string\@caption).\MessageBreak
1711         \string\@caption\space=\space\meaning\@caption}%
1712     \caption@setbool{incompatible}{1}}%

```

The option compatibility= will override the compatibility mode.

```

1713 \ifundefined{caption@ifcompatibility}%
1714     {\let\caption@ifcompatibility\caption@ifincompatible
1715     \let\@tempa\@firstofone}%
1716     {\let\@tempa\@gobble}% suppress warning
1717 \caption@ifcompatibility{%
1718     \@tempa{%

```

```

1719     \PackageWarning{caption}{%
1720         *****\MessageBreak
1721         \noexpand\caption will not be redefined since it's already\MessageBreak
1722         redefined by a document class or package which is\MessageBreak
1723         unknown to the caption package.\MessageBreak
1724         As a result some new features of the caption\MessageBreak
1725         package will not work as expected.\MessageBreak
1726         *****\@gobbletwo}%
1727     }%

\ContinuedFloat is not supported in compatibility mode.
1728     \renewcommand*\caption@ContinuedFloat[1]{%
1729         \caption@Error{Not available in compatibility mode}}%

\caption@start is not supported in compatibility mode.
1730     \caption@AtBeginDocument*{%
1731         \let\caption@start\relax
1732         \@ifundefined{caption@ORI@capstart}{}{%
1733             \caption@Debug{%
1734                 Undo redefinition of \string\capstart\@gobble}%
1735             \let\capstart\caption@ORI@capstart
1736             \let\caption@ORI@capstart\@undefined}%
1737         \@ifundefined{Hy@float@makebox}{}{%
1738             \caption@Debug{%
1739                 Restore hyperref redefinition of \string\float@makebox\@gobble}%
1740             \let\float@makebox\Hy@float@makebox
1741             \let\Hy@float@makebox\@undefined}%
1742         }%

\caption@star We define \caption@star here, too, so it's defined but does not make any harm.
1743     \newcommand*\caption@star[2]{#1#2}%

1744 }{%

1745     \caption@ifincompatible{%
1746         \PackageWarning{caption}{%
1747             *****\MessageBreak
1748             Forced redefinition of \noexpand\caption since the\MessageBreak
1749             unsupported(!) package option 'compatibility=false'\MessageBreak
1750             was given. Please fasten your seat belts, as we may\MessageBreak
1751             experience turbulence!\MessageBreak
1752             *****\@gobbletwo}%
1753     }{}%

\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.
1754     \def\caption{%
1755         \caption@iftype
1756         {\caption@checkgrouplevel\@empty\caption
1757         \caption@star
1758         {\caption@refstepcounter\@captype}%
1759         {\caption@dblarg{\@caption\@captype}}}%
1760     {\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.

```

1761 \newcommand*\caption@star[2]{%
1762   \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](#).)

```

1763 \long\def\@caption#1[#2]#3{%
1764   \ifcaption@star \else
1765     \caption@makecurrent{#1}{#2}%
1766     \caption@ifhypcap{\caption@@start}{}%
1767   \fi
1768   \par
1769   \caption@beginex{#1}{#2}{#3}%
1770   \caption@setfloatcapt{%
1771     \caption@boxrestore
1772     \caption@normalsize
1773     \ifcaption@star
1774       \let\caption@anchor\@firstofone
1775     \fi
1776     \@makecaption{\csname fnum@#1\endcsname}%
1777                   {\ignorespaces\caption@anchor{#3}}\par}%
1778   \caption@end}%

```

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

```

1779 \let\caption@ORI@xfloat\@xfloat
1780 \def\@xfloat#1[#2]{%
1781   \caption@ORI@xfloat{#1}[#2]%
1782   \caption@settype{#1}}%
1783 }%
1784 }

```

`\@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.

```

1785 \long\def\@makecaption#1#2{%
1786   \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
1787   \caption@@make{#1}{#2}%
1788   \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}}

```

2.18 Document class & Babel package support

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

```
1789 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}
```

2.18.2 The beamer class

```
1790 \@ifclassloaded{beamer}{%
1791   \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.

```
1792   \captionsetup{listof=false}
1793   \undefine@key{caption}{listof}
1794   \undefine@key{caption}{listofformat}
1795 }{}
```

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.

```
1796 \@ifundefined{scr@caption}{}{%
1797   \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)

```
1798 \@ifundefined{caption@frenchb}{}{\caption@frenchb}
```

2.18.5 The frenchle/pro package

```
1799 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
1800   \caption@Info{frenchle/pro package is loaded}%

1801   \let\captionfont@ORI\captionfont
1802   \let\captionlabelfont@ORI\captionlabelfont
1803   \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`.

```
1804   \@ifundefined{GOfrench}%
1805     {\let\@tempa\@firstofone}%
1806     {\def\@tempa{\g@addto@macro\GOfrench}}}%
1807   \@tempa{%

1808     \let\captionfont\captionfont@ORI
1809     \let\captionfont@ORI\undefined
1810     \let\captionlabelfont\captionlabelfont@ORI
1811     \let\captionlabelfont@ORI\undefined
1812     \let\@makecaption\@makecaption@ORI
1813     \let\@makecaption@ORI\undefined
```

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

```
1814     \let\@cnORI\caption
```

```

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

1815 \let\caption@tcORI\@tablescaption
1816 \def\@tablescaption{\caption@star\relax\caption@tcORI}%

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

1817 \let\@eatDP\@undefined
1818 \let\@tempa\@empty
1819 \ifx\ffrench\figure
1820 \l@addto@macro\@tempa{\let\fnum@figure\ffrench}%
1821 \fi
1822 \ifx\tfrench\table
1823 \l@addto@macro\@tempa{\let\fnum@table\tfrench}%
1824 \fi
1825 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
1826 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
1827 \@tempa

1828 }%
1829 }

```

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.

1830 \newcommand\caption@ifpackageloaded[1]{%
1831 \testopt{\caption@ifpackageloaded{#1}}{}}
1832 \@onlypreamble\caption@ifpackageloaded

1833 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
1834 \ifpackageloaded{#1}\@firstofone{%
1835 \caption@Debug{#1 package is not loaded (yet)\@gobble}%
1836 \caption@AtBeginDocument}{%
1837 \caption@ifpackageloaded{#1}[#2]{#3}{#4}}
1838 \@onlypreamble\caption@ifpackageloaded

1839 \newcommand\caption@ifpackageloaded[1]{%
1840 \testopt{\caption@ifpackageloaded{#1}}{}}
1841 \@onlypreamble\caption@ifpackageloaded

1842 \long\def\caption@ifpackageloaded#1[#2]{%
1843 \ifpackageloaded{#1}{%
1844 \caption@Info{#1 package is loaded}%
1845 \ifpackagelater{#1}{#2}\@firstoftwo{%
1846 \caption@Error{%
1847 For a successful cooperation we need at least version\MessageBreak
1848 `#2' of package #1,\MessageBreak
1849 but only version\MessageBreak

```

```

1850      '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
1851      is available}%
1852      \@secondoftwo}%
1853    }\@secondoftwo}}
1854 \@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.

```

1855 \newcommand*\caption@clearmargin{%
1856   \setcaptionmargin\z@
1857   \let\caption@minmargin\@undefined}

1858 \caption@setbool{needfreeze}{0}
1859 \caption@AtBeginDocument*{%
1860   \caption@ifneedfreeze{%

```

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

```

1861 \newcommand*\caption@freeze{%
1862   \caption@teststar\caption@@freeze\@gobble\@firstofone}%

1863 \newcommand*\caption@@freeze[1]{%
1864   \global\let\caption@SCcontinued\relax
1865   \global\let\caption@SCsetup\@undefined
1866   \global\let\caption@SClentry\@undefined
1867   \global\let\caption@SCtext\@undefined
1868   \global\let\caption@SClabel\@undefined

1869   \let\caption@ORI@ContinuedFloat\ContinuedFloat
1870   \def\ContinuedFloat{%
1871     \let\caption@ORI@setcounter\setcounter
1872     \let\caption@ORI@addtocounter\addtocounter
1873     \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
1874     \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
1875     \caption@ORI@ContinuedFloat
1876     \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
1877     \let\setcounter\caption@ORI@setcounter
1878     \let\addtocounter\caption@ORI@addtocounter}%
1879   \def\captionsetup##1{%
1880     \caption@g@addto@list\caption@SCsetup{##1}%
1881     \caption@setup{##1}}%
1882   \def\caption{%
1883     \let\captionsetup\@gobble
1884     \caption@star
1885     {\let\caption@ORI@refstepcounter\caption@@refstepcounter
1886      \def\caption@@refstepcounter#####1{%
1887        \advance\csname c@#####1\endcsname\@ne\relax}%
1888      \caption@refstepcounter\@captype
1889      \let\caption@@refstepcounter\caption@ORI@refstepcounter}%
1890     {\caption@dblarg\caption@SC}}%
1891   \def\@caption@SC[##1]##2{%
1892     \gdef\caption@SClentry{##1}\gdef\caption@SCtext{##2}%
1893     \ignorespaces}%
1894   #1{% is \@gobble in star form, and \@firstofone otherwise
1895     \def\label##1{\gdef\caption@SClabel{##1}\ignorespaces}}%

```

```

1896 }%

\caption@defrost \caption@defrost
1897 \newcommand*\caption@defrost{%
1898   \ifx\caption@ORI@ContinuedFloat\@undefined
1899     \caption@SCcontinued
1900     \ifx\caption@SCsetup\@undefined \else
1901       \expandafter\captionsetup\expandafter{\caption@SCsetup}%
1902     \fi
1903     \ifx\caption@SCtext\@undefined \else
1904       \expandafter\expandafter\expandafter\caption
1905         \expandafter\expandafter\expandafter[%
1906         \expandafter\expandafter\expandafter{%
1907         \expandafter\caption@SClentry\expandafter}\expandafter]%
1908       \expandafter{\caption@SCtext}%
1909     \fi
1910     \ifx\caption@SClabel\@undefined \else
1911       \expandafter\label\expandafter{\caption@SClabel}%
1912     \fi
1913   \else
1914     \caption@Error{Internal error:\MessageBreak
1915       \noexpand\caption@defrost in same group as \string\caption@freeze}%
1916   \fi}%

1917 }{}%
1918 \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.)

```

1919 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
1920   \@ifpackageloaded{floatrow}{%
1921     \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
1922   }{}%

```

`\@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>]`.

```

1923 \let\caption@ORI@float@setevery\@float@setevery
1924 \def\@float@setevery#1{%
1925   \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.

```
1926 \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.

```
1927 \renewcommand\caption@setfloatcapt{%
1928 \let\@makecaption\caption@@make
1929 \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.

```
1930 \float@getstyle\float@style{#1}%
1931 \caption@setstyle*\float@style
1932 \caption@setoptions\float@style
1933 }{}%
1934 \caption@freezeHref % will be defrosted in \float@makebox
1935 \caption@ORI@float@setevery{#1}}%
```

`\caption@setfloatcapt` Here we do a consistency check: If the caption to handle is a one from the float package (`\newfloat` or `\restylefloat`), `\setfloatcapt` should be redefined, so actually this error should never occur.

```
1936 \renewcommand*\caption@setfloatcapt{%
1937 \float@ifcaption\@capttype{%
1938 \caption@Error{Something is going wrong here...\MessageBreak
1939 For a removal of this problem please consult the\MessageBreak
1940 caption package documentation, especially the sections\MessageBreak
1941 about the 'float' and the 'wrapfig' package}%
1942 \@gobble
1943 }{%
1944 \@firstofone
1945 }}%
```

`\caption@settypehook` L^AT_EX and almost every other packages use `\langle type \rangle`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.

```
1946 \g@addto@macro\caption@settypehook{%
```



```

1947     \expandafter\ifx\csname #1name\endcsname\relax
1948     \expandafter\let\csname #1name\expandafter\endcsname
1949         \csname fname@#1\endcsname
1950     \fi}%

\fs@plaintop \fs@boxed Since the float styles plaintop and boxed don't use \abovecaptionskip
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.

1951 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
1952 \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).

1953 \providecommand*\float@ifstyle[1]{%
1954     \expandafter\ifx\csname fst@#1\endcsname\relax
1955     \expandafter\@secondoftwo
1956     \else
1957     \expandafter\@firstoftwo
1958     \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.

1959 \providecommand*\float@getstyle[2]{%
1960     \edef#1{%
1961         \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
1962         \expandafter\expandafter\expandafter\noexpand
1963         \csname fst@#2\endcsname}%
1964     \edef#1{#1}%
1965     \caption@Debug{floatstyle{#2} = '#1'}}%

\float@setstyle \float@setstyle{<type>}{<style>}
Sets or changes the float style associated with <type>.

1966 \providecommand*\float@setstyle[2]{%
1967     \expandafter\edef\csname fst@#1\endcsname{%
1968         \expandafter\noexpand\csname fs@#2\endcsname}}%

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

1969 \providecommand*\float@dostyle[1]{%
1970     \@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.)
1971 \providecommand*\float@ifcaption[1]{%
1972   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
1973   \expandafter\@firstoftwo
1974   \else
1975   \expandafter\@secondoftwo
1976   \fi}%
1977 }{%
1978 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
1979 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
1980 % \clearcaptionsetup{boxed}% used by the floatrow package?
1981 }

```

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

```
1982 \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’.

```

1983 \caption@ifbool{ruled}{%
1984   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
1985 }{%
1986   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
1987 }
1988 \caption@undefbool{ruled}

```

2.19.2 The floatflt package

```
1989 \caption@IfPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

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

```

1990 \let\caption@ORI@floatingfigure\floatingfigure
1991 \def\floatingfigure{%
1992   \caption@floatflt{figure}%
1993   \caption@ORI@floatingfigure}%

```

\floatingtable Same with \floatingtable...

```

1994 \let\caption@ORI@floatingtable\floatingtable
1995 \def\floatingtable{%
1996   \caption@floatflt{table}%
1997 % \caption@setautoposition b%
1998   \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.

```

1999 \newcommand*\caption@floatflt[1]{%
2000   \caption@settype{#1}%
2001   \caption@setoptions{floating#1}%
2002   \let\caption@boxrestore\@parboxrestore}%

2003 }{}

```

2.19.3 The fltpage package

```

2004 \caption@ifpackageloaded{fltpage}[1998/10/29 v.0.3]{%
2005   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

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

```

```

2006 \renewcommand\FP@helpNote[2]{%
2007   \begingroup % save \caption@thepage
2008     \caption@pageref{#2}%
2009     \typeout{FP#1 is inserted on page \caption@thepage!}%
2010   \endgroup}%

```

`\FP@floatBegin` Original code:

```

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

```

```

2011 \renewcommand*\FP@floatBegin[1]{%
2012   \def\@captive{#1}%
2013   \let\FP@LabelText\@empty
2014   \begin{lrbox}{\FP@floatCorpusBOX}%
2015     \caption@ifFPrefcap\caption@freeze{%
2016       \def\label##1{\gdef\FP@LabelText{##1}\ignorespaces}%
2017     \caption@freeze*}%

```

`\FP@floatEnd` Original code:

```

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

2018 \renewcommand*\FP@floatEnd{%
2019   \end{lrbox}%
2020   \stepcounter{FP@{\@cptype} C}%
2021   \caption@label\FP@positionLabel
2022   \FP@helpNote{\@cptype}\FP@positionLabel
2023   \let\FP@ContinuedFloat\relax
2024   \FP@float
2025     {\FP@positionLabel}% location label test
2026     {\begin{\@cptype}[p!]}%
2027       \FP@ContinuedFloat
2028       \usebox\FP@floatCorpusBOX
2029       \caption@ifFlistcap
2030         {\caption@refstepcounter{\@cptype}
2031           \expandafter\caption@makecurrent\expandafter{\@cptype}
2032             \expandafter{\caption@SClentry}}}%
2033         {\expandafter\caption@listentry\expandafter{\caption@SClentry}}}%
2034       \caption@anchor\relax
2035       \ifx\FP@LabelText\@empty \else
2036         \expandafter\label\expandafter{\FP@LabelText}%
2037       \fi
2038     \end{\@cptype}%
2039   {\let\FP@ContinuedFloat\ContinuedFloat
2040     \caption@ifcompatibility

```

```

2041         {\let\caption@ContinuedFloat\@gobble}%
2042         {\caption@clearsetup{ContinuedFloat}}%
2043         \caption@clearsetup{continued\@capttype}}}%
2044     {\begin\@capttype[b!]}%
2045         \FP@ContinuedFloat
2046         \let\FP@savedSetfnumCommand\caption@setfnum
2047         \def\caption@setfnum##1{%
2048             \FP@savedSetfnumCommand{##1}%
2049             \ifx\FP@guide\@empty \else
2050                 \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}}%
2051             \fi}%
2052         \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2053         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2054         \caption@setoptions{FP\@capttype}%
2055         \FP@separatorCaption
2056         \caption@ifFPlistcap{}{\caption@set@bool\caption@iflof0}%
2057         \caption@defrost
2058     \end\@capttype}%
2059 }%

2060 }{%
2061     \let\caption@ifFPlistcap\@undefined
2062     \let\caption@ifFPrefcap\@undefined
2063 }

```

2.19.4 The hyperref package

```

2064 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2065     \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2066         \PackageWarningNoLine{caption}{%
2067             Hyperref support is turned off\MessageBreak
2068             because hyperref has stopped early}%
2069     }{%

```

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

```

2070     \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

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

```

2071     \renewcommand*\caption@makecurrent[2]{%
2072         \caption@makecurrentHref{#1}%
2073         \caption@Debug{hyper current=\@currentHref}%
2074         \def\@currentlabelname{#2}}}%
2075     \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

`\caption@makestart` We redefine `\caption@makestart` so a hyperref label will be defined inside `\caption@start`. Since we offer `\ContinuedFloat` the float counter can be 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.xxx.<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.

```

2076 \renewcommand*\caption@makestart[1]{%
2077 \begingroup
2078 \Hy@hypertextnamesfalse
2079 \gdef\@currentHlabel{}%
2080 \hyper@makecurrent{#1.xxx}%
2081 \endgroup
2082 \caption@Debug{hyper start=\@currentHref}}%
```

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

```

2083 \renewcommand*\caption@anchor[1]{%
2084 \caption@Debug{hyper anchor \@currentHref}%
2085 % If we cannot have nesting, the anchor is empty.
2086 \ifHy@nesting
2087 \hyper@@anchor{\@currentHref}{#1}%
2088 \else
2089 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2090 \fi}%%
```

`\caption@@anchor` We define `\caption@@anchor` so a `hyperref` anchor will be set inside `\caption@start`.

```

2091 \newcommand*\caption@@anchor[1]{%
2092 \caption@Debug{hyper anchor #1}%
2093 \begingroup
2094 \@tempdima\prevdepth
2095 \nointerlineskip
2096 \vspace*{-\caption@hycapspace}%
2097 \let\leavevmode\relax
2098 \hyper@@anchor{#1}\relax
2099 \vspace*{\caption@hycapspace}%
2100 \prevdepth\@tempdima
2101 \endgroup}%%
```

`\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

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

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

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

```

2102 \renewcommand*\caption@@ContinuedFloat[1]{%
2103 \ifundefined{theH#1}}{}%
2104 \expandafter\l@addto@macro\csname theH#1\endcsname{%
2105 \@alph\c@ContinuedFloat}}}%

2106 \newif\if@capstart

\caption@start This one emulates the hypcap package (if hypcap=true).
2107 \def\caption@start{%
2108 \caption@ifhypcap{%
2109 \ifvmode

Generate the hyperref label and set the hyperref anchor, usually both is done inside
\@caption.

2110 \caption@makestart\@captype
2111 \caption@@anchor\@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.)

2112 \global\@capstarttrue
2113 \let\hc@currentHref\@currentHref
2114 \def\caption@makecurrentHref##1{%
2115 \global\@capstartfalse
2116 \global\let\@currentHref\hc@currentHref}%

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

2117 \let\caption@anchor\@firstofone
2118 \let\caption@@anchor\@gobble

2119 \else
2120 \PackageWarning{caption}{Wrong TeX mode, no hyperlink anchor was set}%
2121 \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.

2122 \def\caption@@start{%
2123 \ifundefined{hc@currentHref}}{%
2124 \PackageWarning{caption}{%
2125 The option 'hypcap=true' will be ignored for this\MessageBreak
2126 particular \string\caption}}}%

\caption@freezeHref Suppress \caption@start from generating a hyperref label and setting a hyper-
ref anchor. Instead if \@caption generates a hyperref label, it will be stored in
\caption@currentHref. Furthermore we need to redefine \caption@setfloatcapt
so no hyperref anchor will be set in \@caption.

2127 \def\caption@freezeHref{%
2128 \let\caption@ORI@start\caption@start
2129 \def\caption@start{\let\caption@start\caption@ORI@start}%

2130 \let\caption@ORI@@start\caption@@start
2131 % \l@addto@macro\subcaption@settypehook{%
2132 % \let\caption@@start\caption@ORI@@start}%
2133 \global\let\caption@currentHref\@undefined
2134 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%

```

```

2135     \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2136     \renewcommand*\caption@setfloatcapt{%
2137         \ifx\caption@currentHref\@undefined \else
2138             \let\caption@anchor\@firstofone
2139         \fi
2140     \caption@ORI@setfloatcapt}}%

\caption@defrostHref If there is a freezed \@currentHref, we set the hyperref anchor here.
2141     \def\caption@defrostHref{%
2142         \ifx\caption@currentHref\@undefined \else
2143             \caption@@anchor\caption@currentHref
2144             \global\let\caption@currentHref\@undefined
2145         \fi}%

\float@makebox Do our own redefinition of \float@makebox, if it was redefined by the hyperref package.
2146     \@ifundefined{HyOrg@float@makebox}{}{%
2147         \caption@Debug{%
2148             Redefining \noexpand\float@makebox (again)\@gobble}%
2149         \let\Hy@float@makebox\float@makebox
2150         \renewcommand\float@makebox[1]{%
2151             \HyOrg@float@makebox{#1}\relax \caption@defrostHref}}%
2152     }%

2153 }{}

2.19.5 The hypcap package

2154 \caption@IfPackageLoaded{hypcap}{% v1.0
2155     \ifx\caption@start\relax \else % hyperref hasn't stopped early

If the hypcap package was loaded, we give up our own hypcap emulation and give the control 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.
2156     \let\caption@ORI@capstart\capstart
2157     \let\capstart\caption@start
2158     \let\caption@start\relax
2159     \let\caption@@start\relax

\caption@hypcapspace Furthermore we map our \caption@hypcapspace to \hypcapspace offered by the hypcap package.
2160     \renewcommand*\caption@hypcapspace{\hypcapspace}%

\float@makebox Redo the redefinition of \float@makebox, if necessary.
2161     \@ifundefined{Hy@float@makebox}{}{%
2162         \caption@Debug{%
2163             Restore hyperref redefinition of \string\float@makebox\@gobble}%
2164         \let\float@makebox\Hy@float@makebox
2165         \let\Hy@float@makebox\@undefined}%
2166     \fi}{}

```


2.19.6 The listings package

```

2167 \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.
2168 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2169 \def\lst@MakeCaption#1{%
2170 \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.
2171 \caption@setautoposition{#1}%
2172 \caption@begin{lstlisting}%
2173 \caption@ORI@lst@MakeCaption{#1}%
2174 \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.
2175 \providecommand*\ext@lstlisting{lol}%

\lst@@caption We define \lst@@caption to non-@empty, so \fnum@lstlisting will pro-
duce a numbered listing caption, which makes \captionof{lstlisting} work
correctly.
2176 \g@addto@macro\caption@settypehook{\let\lst@@caption\relax}%
2177 }{}

```

2.19.7 The longtable package

```

\LTcapttype \LTcapttype is preset to table.
2178 \providecommand*\LTcapttype{table}

2179 \caption@ifpackageloaded{longtable}[1995/05/24 v3.14]{%
2180 \RequirePackage{ltcaption}[2007/09/01]%
2181 \let\LT@makecaption\@undefined

\LT@array We redefine \LT@array here to get \captionsetup{<options>} working inside
longtables.
2182 \let\caption@ORI@LT@array\LT@array
2183 \renewcommand*\LT@array{%
\captionsetup for longtable:
2184 \global\let\caption@opt@@longtable\@undefined
2185 \def\captionsetup##1{\noalign{%
2186 \captionsetup@gobble[@longtable]{##1}%
2187 \global\let\caption@opt@@longtable\caption@opt@@longtable}}%
\captionlistentry for longtable:
2188 \def\captionlistentry##1{\noalign{%
2189 \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`

```

2190 % \caption@ifhypcap{%
2191 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2192 % \def\hyper@makecurrent##1{%
2193 % \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2194 % \caption@makestart{##1}%
2195 % \let\Hy@LT@currentHlabel\@currentHlabel
2196 % \let\Hy@LT@currentHref\@currentHref
2197 % \def\hyper@makecurrent####1{%
2198 % \let\@currentHlabel\Hy@LT@currentHlabel
2199 % \let\@currentHref\Hy@LT@currentHref}}%
2200 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
2201 % \def\ContinuedFloat{\noalign{%
2202 % \gdef\caption@setContinuedFloat{%
2203 % \let\caption@resetContinuedFloat\@gobble}%
2204 % \def\caption@setoptions####1{%
2205 % \g@addto@macro\caption@setContinuedFloat{%
2206 % \caption@setoptions{####1}}}%
2207 % \let\@capttype\LTcapttype
2208 % \caption@ORI@ContinuedFloat}}%
2209 % }{%
2210 % \def\ContinuedFloat{\noalign{%
2211 % \caption@Error{\noexpand\ContinuedFloat inside longtables\MessageBreak
2212 % is only available with 'hypcap=true'}}}%
2213 % }%
2214 % \global\let\caption@setContinuedFloat\@empty
2215 % \def\ContinuedFloat{\noalign{%
2216 % \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2217 % \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}`:

```

2218 \def\LT@c@ption#1[#2]#3{%
2219 \LT@makecaption#1{\csname fnun@\LTcapttype\endcsname}{#3}%
2220 \def\@currentlabelname{#2}%
2221 \caption@addcontentsline\LTcapttype{#2}}%

```

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

The original definition:

```

\def\LT@makecaption#1#2#3{%
  \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\LTcapwidth{%

```

```

% Based on article class "\@makecaption", "#1" is "\@gobble" in star
% form, and "\@firstofone" otherwise.
\sbox\@tempboxa{#1{#2: }#3}%
\ifdim\wd\@tempboxa>\hsize
  #1{#2: }#3%
\else
  \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
\fi
\endgraf\vskip\baselineskip}%
\hss}}

```

Our definition:

```

2222 \renewcommand\LT@makecaption[3]{%
2223   \caption@LT@make{%

```

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

```

2224   \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`.)

```

2225   \caption@settype*\LTcaptype
2226   \ifdim\LTcapwidth=4in \else
2227     \setcaptionwidth\LTcapwidth
2228   \fi
2229   \caption@setoptions{longtable}%
2230 %   \caption@setContinuedFloat
2231   \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!

```

2232   \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
2233   \caption@resetContinuedFloat\LTcaptype
2234   \caption@begin\LTcaptype
2235   \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...)

```

2236   \vskip-\ht\strutbox

```

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

```

2237   \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2238   \caption@@make{#2}{#3}\endgraf
2239   \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2240   \caption@end}%
2241 }{}

```

2.19.8 The picinpar package

2242 \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.

```
2243 \long\def\figwindow[#1,#2,#3,#4] {%
2244   \caption@window{figure}%
2245   \caption@setoptions{figwindow}%
2246   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
2247 \long\def\tabwindow[#1,#2,#3,#4] {%
2248   \caption@window{table}%
2249   \caption@setoptions{tabwindow}%
2250   \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.

```
2251 \newcommand*\caption@window{%
2252   \let\caption@boxrestore\@parboxrestore
2253   \let\@makecaption\caption@@make
2254   \caption@setautoposition b%
2255   \caption@settype}%

```

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

2256 \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.

```
2257 \ifdim\picwd=\z@
2258   \let\caption@makecurrent\@gobbletwo
2259   \let\caption@@start\relax
2260   \caption@prepareslc
2261 \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.

```
2262 \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2263 \if\@tempa*
2264   \let\@tempa\@firstofone
2265 \else\if\@tempa[%
2266   \let\@tempa\@firstofone
2267 \else
2268   \let\@tempa\@empty
2269 \fi\fi
2270 \expandafter\caption\@tempa{#1}}%
```

2271 }{ }

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 `\@capttype`, as proposed in the picins documentation.

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

```
2272 \newcommand*\piccaptiontype[1]{\def\@piccapttype{#1}}
```

```
2273 \caption@ifpackageloaded{picins}{%
```

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

```
2274 \ifundefined{\@piccapttype}{%
```

```
2275 \caption@iftype{%
```

```
2276 \let\@piccapttype\@capttype
```

```
2277 }{%
```

```
2278 \def\@piccapttype{figure}%
```

```
2279 }%
```

```
2280 }{}%
```

```
2281 \let\@capttype\@undefined
```

`\piccaption` The original code:

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

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

```
2282 \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.

```
2283 \let\caption@ORI\make@piccaption\make@piccaption
```

```
2284 \def\make@piccaption{%
```

```
2285 \let\caption@ORI\caption
```

```

2286 \long\def\caption[##1]##2{%
2287 \caption@freezeHref % will be defrosted in \ivparpic
2288 \caption@settype\@piccaptiontype
2289 \ifnum\c@piccaptionpos>2\relax
2290 \caption@clearmargin
2291 \else
2292 \captionwidth\z@ % do not use "width=" setting
2293 \fi
2294 \caption@setoptions{parpic}%
2295 \let\caption@boxrestore\@parboxrestore
2296 \caption@setautoposition b%
2297 \expandafter\expandafter\expandafter\caption@ORI
2298 \expandafter\expandafter\expandafter[%
2299 \expandafter\expandafter\expandafter{%
2300 \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
2301 \caption@ORI@make@piccaption
2302 \let\caption\caption@ORI%

```

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

```

2303 \let\caption@ORI@ivparpic\ivparpic
2304 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
2305 \let\caption@ORI@noindent\noindent
2306 \def\noindent{%
2307 \caption@defrostHref
2308 \let\noindent\caption@ORI@noindent
2309 \noindent}%
2310 \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2311 \let\noindent\caption@ORI@noindent}%
2312 }{%
2313 \let\piccaptiontype\@undefined
2314 }

```

2.19.10 The rotating package

```
2315 \caption@ifPackageLoaded{rotating}[1995/08/22 v2.10]{%
```

\rotcaption Make \rotcaption* work.

```

2316 \def\rotcaption{\let\makecaption\@makerotcaption\caption}%
2317 % \let\@rotcaption\@undefined

```

\rotcaptionof Make \rotcaptionof(*) work.

```

2318 \def\rotcaptionof{%
2319 \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

```

`\@makerotcaption` Original (bugfixed) code:

```
\long\def\@makerotcaption#1#2{%
  \setbox\@tempboxa\hbox{#1: #2}%
  \ifdim \wd\@tempboxa > .8\vsizel
    \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.)

```
2320 \long\def\@makerotcaption#1#2{%
2321   \ifdim\captionwidth=\z@
2322     \setcaptionwidth{.8\textheight}%
2323     \caption@slc{#1}{#2}{.8\vsizel}%
2324     \let\caption@makerot\caption@@make
2325     \caption@clearmargin
2326 %     \long\def\captionbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2327 %     (not needed because \rotatebox uses an \hbox anyway)
2328     \let\captionbox\@secondoftwo%
2329     \caption@set@bool\caption@ifslc0% been there, done that
2330   \fi
2331   \rotatebox{90}{\caption@makerot{#1}{#2}}%
2332   \nobreak\hspace{12pt}}%
2333 \newcommand\caption@makerot[2]{%
2334   \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
2335 {} }
```

2.19.11 The sidecap package

```
2336 \caption@ifPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
2337   \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.)

```
2338 \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`.

```
2339 \let\caption@ORI@SC@zfloat\SC@zfloat
2340 \def\SC@zfloat#1#2#3[#4]{%
2341   \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]}%
```

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.

```

2342     \caption@settype*{#2}%
2343     \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.

2344     \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
2345     \def\endSC@FLOAT{%

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

2346     \let\caption@ORI@settype\caption@settype
2347     \def\caption@settype##1{% will be done in \xfloat
2348         \caption@ORI@settype*{##1}% do not change \@currentlabel
2349         \caption@setSC@justify
2350         \caption@setoptions{SCfloat}%
2351         \caption@setoptions{SC\caption}%
2352         \caption@start}%

Before we can typeset the caption we need to set the margin to zero because any extra
margin would only be disturbing here.
(We don't need to take care about the caption position because the sidecap package set
both \abovecaptionskip and \belowcaptionskip to a skip of zero anyway.)
Furthermore \SC@justify will override the caption justification, if set. The usage of
\SC@justify differs from version to version of the sidecap package:
Version 1.4: \SC@justify is not defined
Version 1.5: \SC@justify is \relax when not set
Version 1.6: \SC@justify is \@empty when not set

2353     \def\caption@setSC@justify{%
2354         \caption@clearmargin
2355         \ifundefined{SC@justify}{}{}%
2356         \ifx\SC@justify\@empty \else
2357             \let\caption@hj\SC@justify
2358             \let\SC@justify\@empty
2359         \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!

2360     \let\caption\SC@orig@caption
2361     \def\SC@orig@caption[##1]##2{\caption@defrost}%

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

2362     \caption@setSC@justify % for compatibility mode
2363     \caption@ORI@endSC@FLOAT}%

2364 }{}

```

2.19.12 The subfigure package

```

2365 \caption@ifPackageLoaded{subfigure}[2002/01/23 v2.1]{%

\sif@ifpositiontop If the subfigure package is loaded, we map \sf@ifpositiontop to \iffiguretopcap
resp. \iftabletopcap, so the subfigure v2.1 options figbotcap etc. will still work.

2366     \def\sf@ifpositiontop{%

```



```

2367 \ifundefined{if\@capttype topcap}{%
2368 \gobbletwo
2369 }{%
2370 \@nameuse{if\@capttype topcap}%
2371 \expandafter\@firstoftwo
2372 \else
2373 \expandafter\@secondoftwo
2374 \fi
2375 } }%

2376 }{}

```

2.19.13 The supertabular package

```

2377 \caption@ifPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.
2378 \let\caption@ORI@tablecaption\tablecaption
2379 \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}

2380 \long\def\ST@caption#1[#2]#3{\par%
2381 \caption@settype*{#1}%
2382 \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.

2383 \def\caption@fixposition{%
2384 \caption@setposition{\if@topcaption t\else b\fi}}%

2385 \caption@beginex{#1}{#2}{#3}%
2386 \caption@boxrestore
2387 \caption@normalsize
2388 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2389 \caption@end}%

2390 }{%
2391 \let\caption@setSTposition\@undefined
2392 }

```

2.19.14 The threeparttable package

2393 \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.

```
2394 \let\caption@ORI@threeparttable\threeparttable
2395 \renewcommand*\threeparttable{%
2396   \caption@settype{table}%
2397   \caption@setposition a% ?
2398   \caption@clearmargin
2399   \caption@setoptions{threeparttable}%
2400   \caption@ORI@threeparttable}%
```

\measuredfigure Same here...

```
2401 \let\caption@ORI@measuredfigure\measuredfigure
2402 \renewcommand*\measuredfigure{%
2403   \caption@settype{figure}%
2404   \caption@setposition a% ?
2405   \caption@clearmargin
2406   \caption@setoptions{measuredfigure}%
2407   \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}

2408 \def\TPT@caption#1[#2]#3{%
2409   \gdef\TPT@docapt{%
2410     \global\let\TPT@docapt\@undefined
2411     \caption@setautoposition\caption@TPT@position
2412     \TPT@LA@caption{#1}[#2][#3]}%
2413   \ifx\TPT@hsize\@empty
2414     \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
2415     \gdef\caption@TPT@position{t}%
2416     \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2417   \else
2418     \def\caption@TPT@position{b}%
2419     \TPT@docapt
2420   \fi
2421   \ignorespaces}%
2422   %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2423   \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima}%
2424 }{}
```

2.19.15 The wrapfig package

2425 \caption@ifPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)
 \float@ifstyle \float@ifstyle{<type>}{<if-clause>}{<else-clause>}}

(see float package support for details)

```
2426 \providecommand*\float@ifstyle[1]{%
2427   \expandafter\ifx\csname fst@#1\endcsname\relax
2428   \expandafter\@secondoftwo
2429   \else
2430   \expandafter\@firstoftwo
2431   \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.

```
2432 \newcommand*\caption@restylewrapfloat[1]{%
2433   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2434   \csname wrap#1\endcsname
2435   \caption@Error{%
2436     For a successful cooperation of the 'wrapfig' package\MessageBreak
2437     with the 'float' package you should load the 'wrapfig'\MessageBreak
2438     package *after* the 'float' package}%
2439   \else
2440     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2441     \csname wrap#1\endcsname
2442     \@namedef{wrap#1}{%
2443       \caption@settype*{#1}%
2444       \float@ifstyle{#1}{%
2445         \ifx\WF@floatstyhook\@undefined
2446         \caption@Error{%
2447           For a successful cooperation of the 'wrapfig' package\MessageBreak
2448           with the 'float' package you should use at least\MessageBreak
2449           'wrapfig' version 3.6}%
2450         \else
2451         \float@dostyle{#1}%
2452         \fi}{}%
2453       \caption@clearmargin
2454       \caption@setoptions{wrapfloat}%
2455       \caption@setoptions{wrap#1}%
2456       \@nameuse{caption@ORI@wrap#1}}%
2457     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2458     \csname wrap#1\endcsname
2459   \fi}%
```

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.)

```
2460 \caption@restylewrapfloat{figure}%
2461 \caption@restylewrapfloat{table}%
2462 \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.

```

2463 \ifundefined{float@restyle}{}{}%
2464 \toks@=\expandafter{\float@restyle{#1}% (env may or may not be defined)
2465 \caption@restylewrapfloat{#1}}%
2466 \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2467 \@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`.

```

2468 \let\caption@ORI@wrapfloat\wrapfloat
2469 \def\wrapfloat#1{%
2470 \float@ifstyle{#1}{%
2471 \caption@Error{%
2472 For a successful cooperation of the 'wrapfig' package\MessageBreak
2473 with the 'float' package you should load the 'wrapfig'\MessageBreak
2474 package *right after* the 'float' package}}}%
2475 \caption@ORI@wrapfloat{#1}}%

```

2476 `\fi` % wrapfig v3.6

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

Original code:

```

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

```

Our code:

```

2477 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2478 \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2479 \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
2480 \caption@start
2481 \ifdim\hsize>\z@ \@parboxrestore \else
2482 \setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
2483 \ignorespaces \fi}%
2484 {}

```

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,
2006/07/24
- [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 L^AT_EX,
1997/09/26
- [16] Rolf Niepraschk & Hubert Gäßlein:
The sidecap package,
2003/06/06
- [17] Steven D. Cochran:
The subfigure package,
2002/07/02
- [18] Steven D. Cochran:
The subfig package,
2005/07/05
- [19] Johannes Braams and Theo Jurriens:
The supertabular environment,
2002/07/19
- [20] Donald Arseneau:
Three part tables: title, tabular environment, notes,
2003/06/13
- [21] Donald Arseneau:
WRAPFIG.STY ver 3.6,
2003/01/31
- [22] Anne Brüggemann-Klein:
Einführung in die Dokumentverarbeitung,
B.G. Teubner, Stuttgart, 1989
- [23] Heiko Oberdiek:
The refcount package,
2006/02/20