

The Implementation of the caption package^{*}

Axel Sommerfeldt
caption@sommerfee.de

2007/09/22

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 v3.1b, last revised 2007/09/22.

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	7
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	25
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	34
2	Main package	36
2.1	Identification	36
2.2	Loading the kernel	36
2.3	Check against incompatible packages	36
2.4	Check document class	36
2.5	Adaption to the $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes	37

2.6	Emulation of the KOMA-Script commands	37
2.7	Option for configuration files	39
2.8	Options for figure and table	39
2.9	caption v1.x compatibility options	39
2.10	caption2 v2.x compatibility options	40
2.11	Obsolete caption v3.0 options	40
2.12	ftpage package support options	41
2.13	hyperref package support options	41
2.14	Processing of options	41
2.15	Option ‘type’, \captionof and \ContinuedFloat	41
2.16	Internal helpers	44
2.17	\caption, \@caption, and \@makecaption	46
2.18	Document class & Babel package support	53
2.18.1	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	53
2.18.2	The beamer class	53
2.18.3	The KOMA-Script classes	53
2.18.4	The frenchb Babel option	53
2.18.5	The frenchle/pro package	53
2.19	Package support	54
2.19.1	The float package	56
2.19.2	The floatflt package	59
2.19.3	The ftpage package	60
2.19.4	The hyperref package	62
2.19.5	The hypcap package	65
2.19.6	The listings package	66
2.19.7	The longtable package	66
2.19.8	The picinpar package	69
2.19.9	The picins package	70
2.19.10	The rotating package	72
2.19.11	The sidecap package	73
2.19.12	The subfigure package	74
2.19.13	The supertabular package	74
2.19.14	The threeparttable package	75
2.19.15	The wrapfig package	76

1 Kernel

1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2007/09/22 v3.1b 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  $\backslash @removeelement$  is using  $\backslash edef$  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
\caption@set@bool
\caption@ifbool
\caption@undefbool
For setting and testing boolean options we offer these three helper macros:
\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>}
\caption@KV@err 127 \newcommand*\caption@Error[1]{\PackageError{caption}{#1}\caption@eh}
128 \let\caption@KV@err\caption@Error

```

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

```

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

```

1.4 Using the keyval package

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

```

133 \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.

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

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

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

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

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

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

```
154 \newcommand*\clearcaptionsetup{%
155   \caption@teststar\@clearcaptionsetup\@gobble\@firstofone}
156 \newcommand*\@clearcaptionsetup[1]{%
157   \let\caption@tempa#1%
158   \@testopt\@clearcaptionsetup{}}
159 \def\@clearcaptionsetup[#1]#2{%
160   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
161     {\caption@tempa{\PackageWarning{caption}{Option list ‘#2’ undefined}}}%
162     {\ifx,#1,%
```



```

163     \caption@clearsetup{#2}%
164     \else
165     \caption@@removefromsetup{#1}{#2}%
166     \fi}}

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

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

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

```

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

201 \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.

```

202 \newcommand*\caption@setoptions[1]{%
203   \caption@Debug{options=#1}%
204   \expandafter\let\expandafter\@tempa\csname caption@opt@#1\endcsname

```

```

205 \ifx\@tempa\relax \else
206 \let\caption@iftypewarning\@firstoftwo % = true
207 \expandafter\caption@setup\expandafter{\@tempa}%
208 \caption@clearsetup{#1}%
209 \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”).

210 \newcommand*\caption@addtooptlist[1]{%
211 \ifundefined{caption@opt@#1@lineno}{%
212 \caption@dooptlist\caption@g@addto@list{#1}%
213 \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
214 }{}%
215 \newcommand*\caption@removefromoptlist[1]{%
216 \caption@dooptlist\caption@g@removefrom@list{#1}%
217 \global\expandafter\let\csname caption@opt@#1@lineno\endcsname\@undefined}
218 \newcommand*\caption@dooptlist[2]{%
219 \begingroup
220 \edef\@tempa{#2}\@onelevel@sanitize\@tempa
221 \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
222 \endgroup}
223 \AtEndDocument{%
224 \caption@for@list\caption@optlist{%
225 \PackageWarningNoLine{caption}{%
226 Unused \string\captionsetup{#1}
227 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.

228 \newcommand*\caption@setkeys{\@dblarg\caption@@setkeys}
229 \long\def\caption@@setkeys[#1]#2#3{%
230 \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
231 \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
232 \ifundefined{#1@KV@err}%
233 {\let\KV@err\ORI@KV@err\let\KV@errx\ORI@KV@errx}%
234 {\expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
235 \let\KV@errx\KV@err}%
236 \l@addto@macro\caption@keydepth i%
237 \caption@Debug{\protect\setkeys{#2}{#3}}%
238 \setkeys{#2}{#3}%
239 \edef\caption@keydepth{\expandafter\@cdr\caption@keydepth\@nil}%
240 \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
241 \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
242 \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.

243 \newcommand*\caption@ExecuteOptions[2]{%
244   \@expandtwoargs\caption@setkeys{#1}{#2}}%
245 \@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.)

246 \newcommand*\caption@ProcessOptions{%
247   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
248 \@onlypreamble\caption@ProcessOptions

249 \newcommand*\caption@@ProcessOptions[2]{%
250   \let\@tempc\relax
251   \let\caption@tempa\@empty
252   #1{% \@firstofone -or- \@gobble
253     \@for\CurrentOption:=\@classoptionslist\do{%
254       \ifundefined{KV@#2@\CurrentOption}}{%
255         \ifundefined{KV@#2@\CurrentOption @default}}{%
256           \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
257         }{%
258           \PackageInfo{#2}{Global option '\CurrentOption' processed}%
259           \edef\caption@tempa{\caption@tempa,\CurrentOption,%
260             \@expandtwoargs\@removeelement\CurrentOption
261             \@unusedoptionlist\@unusedoptionlist
262           }%
263         }%
264       }%
265     \let\CurrentOption\@empty
266   }%
267   \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
268   \AtEndOfPackage{\let\@unprocessedoptions\relax}
269 \@onlypreamble\caption@@ProcessOptions

```

1.5 Margin resp. width

```

\captionmargin \captionmargin and \captionwidth contain the extra margin resp. the total
\captionwidth width used for captions. Please never set these values in a direct way, they are just acces-
sible 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@.

270 \newdimen\captionmargin
271 \newdimen\captionmargin@
272 \newdimen\captionwidth

273 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
274 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
275 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
276 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}

```

```

277 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
278 \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.
279 \newcommand*\setcaptionmargin{%
280   \caption@teststar\caption@setmargin\@gobble\@firstofone}

281 \newcommand*\caption@setmargin[2]{%
282   #1{\captionwidth\z@}%
283   \caption@@setmargin#2,#2,\@nil}

284 \def\caption@@setmargin#1,#2,#3\@nil{%
285   \setlength\captionmargin@{#2}%
286   \setlength\captionmargin{#1}%
287   \addtolength\captionmargin@{-\captionmargin}}

\setcaptionwidth \setcaptionwidth{<amount>}
Please never use this in user documents, it's just there to provide compatibility to the
caption2 package.
288 \newcommand*\setcaptionwidth{%
289   \captionmargin\z@
290   \captionmargin@\z@
291   \setlength\captionwidth}

\caption@count This counter numbers the captions. At the moment it will be used inside \caption@ifoddpage
only.
292 \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.)
293 \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.
294 \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.
295 \newcommand*\caption@label[1]{%
296   \protected@write\@auxout{\let\caption@thepage\relax}%
297   {\string\caption@newlabel{#1}{\caption@thepage}}

\caption@pageref This command is a modified version of \pageref from LATEX2ε. It will be used inside
\caption@ifoddpage and \FP@helpNote.
298 \newcommand*\caption@pageref[1]{%
299   \expandafter\ifx\csname caption@r@#1\endcsname\relax
300     \G@refundefinedtrue % => 'There are undefined references.'
301     \PackageWarning{caption}{Reference on page \thepage \space undefined}%
302   \else
303     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
304   \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.

```

305 \newcommand*\caption@ifoddpage{%
306   \caption@iftwoside{%
307     \begingroup
308       \@tempcnta\caption@count
309       \advance\@tempcnta\@ne
310       \xdef\caption@count{\the\@tempcnta}%
311     \endgroup
312     \caption@label\caption@count
313     \caption@pageref\caption@count
314     \ifodd\caption@thepage
315       \let\caption@ifoddpage\@firstoftwo
316     \else
317       \let\caption@ifoddpage\@secondoftwo
318     \fi
319   }\let\caption@ifoddpage\@firstoftwo}%
320 \caption@ifoddpage}

```

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

```

321 \newcommand*\caption@setoptcmd[2]{%
322   \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` `\newdimen\caption@indent`
`\caption@hangindent` `\newdimen\caption@parindent`
`\newdimen\caption@hangindent`

```

326 \DeclareCaptionOption{indent}{\leftmargini}{%
327   \setlength\caption@indent{#1}}% obsolete!
328 \DeclareCaptionOption{indentation}{\leftmargini}{%
329   \setlength\caption@indent{#1}}
330 \DeclareCaptionOption{parindent}{%
331   \setlength\caption@parindent{#1}}
332 \DeclareCaptionOption{hangindent}{%
333   \setlength\caption@hangindent{#1}}
334 \DeclareCaptionOption{parskip}{%
335   \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

```

336 \@ifundefined{scr@caption}{}{%
```

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!)

```

337 \let\caption@KV@parindent\KV@caption@parindent
338 \DeclareCaptionOption{parindent}[]{%
339   \ifx,#1,%
340     \caption@Debug{Option 'parindent' ignored}%
341   \else
342     \caption@KV@parindent{#1}%
343   \fi}%
344 \let\caption@KV@parskip\KV@caption@parskip
345 \DeclareCaptionOption{parskip}[]{%
346   \ifx,#1,%
347     \caption@Debug{Option 'parskip' ignored}%
348   \else
349     \caption@KV@parskip{#1}%
350   \fi}%
351 }

```

1.7 Styles

```

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

```

361 \newcommand*\caption@setstyle{%
362   \caption@teststar\caption@@setstyle\@gobble\@firstofone}
363 \newcommand*\caption@@setstyle[2]{%
364   \ifundefined{caption@sty@#2}%
365     {#1{\caption@Error{Undefined style '#2'}}}%
366     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
367     \ifx\caption@setstyle@flag\@undefined
368       \let\caption@setstyle@flag\caption@setstyle
369       \let\caption@iftypewarning\@firstoftwo % = true
370       \caption@resetstyle
371       \expandafter\caption@setup\expandafter{\caption@sty}%
372       \let\caption@setstyle@flag\@undefined
373     \else
374       \expandafter\caption@setup\expandafter{\caption@sty}%

```

```

375 \fi
376 \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
377 \expandafter\caption@l@addto@list\expandafter\caption@opt@SingleLine
378 \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!*

```

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

```

387 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
388 \DeclareCaptionStyle{default}[indent=0pt,justification=centering,format=plain]{}
389   format=default,labelsep=default,textformat=default,%
390   justification=default,font=default,labelfont=default,textfont=default}

```

1.8 Formats

```

\DeclareCaptionFormat \DeclareCaptionFormat{<name>}{<code with #1, #2, and #3>}
\DeclareCaptionFormat*{<name>}{<code with #1, #2, and #3>}

```

The starred form causes the code being typeset in vertical (instead of horizontal) mode, but does not support the `indentation=` option.

```

391 \newcommand\DeclareCaptionFormat{%
392 \caption@teststar\caption@declareformat\@gobble\@firstofone}
393 \@onlypreamble\DeclareCaptionFormat

394 \newcommand\caption@declareformat[3]{%
395 \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
396 \global\long\@namedef{caption@fmt@#2}##1##2##3{#3}}
397 \@onlypreamble\caption@declareformat

398 \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`).

```

399 \newcommand*\caption@setformat[1]{%
400 \ifundefined{caption@fmt@#1}%
401 {\caption@Error{Undefined format '#1'}}%
402 {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
403 \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

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

```

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

```

```

405 \DeclareCaptionFormat{hang}{%
406   \caption@ifin@list\caption@lsep@crlist\caption@lsepname
407   {\caption@Error{%
408     The option 'labelsep=\caption@lsepname' does not work\MessageBreak
409     with 'format=hang'}}}%
410   {\@hangfrom{#1#2}%
411     \advance\caption@parindent\hangindent
412     \advance\caption@hangindent\hangindent
413     \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

414 \def\caption@fmt@default{\caption@fmt@plain}
415 \def\caption@ifh@default{\caption@ifh@plain}

```

1.9 Label formats

```

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

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

```

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

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

```

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

```

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

```

424 \DeclareCaptionLabelFormat{empty}{}
425 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
426 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace}{#2}}

```

‘default’ usually maps to ‘simple’.

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

1.10 Label separators

```

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

```

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

```

428 \newcommand\DeclareCaptionLabelSeparator{%
429   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
430 \@onlypreamble\DeclareCaptionLabelSeparator

431 \newcommand\caption@declarelabelseparator[3]{%
432   \global\expandafter\let\csname caption@iflf@#2\endcsname#1%
433   \global\long\@namedef{caption@lsep@#2}{#3}%
434   \caption@@declarelabelseparator{#2}#3\\@nil}
435 \@onlypreamble\caption@declarelabelseparator

```



```

436 \long\def\caption@@declarelabelseparator#1#2\\#3\@nil{%
437   \def\@tempa{#3}\ifx\@tempa\@empty \else
438     \caption@g@addto@list\caption@lsep@crlist{#1}%
439   \fi}
440 \@onlypreamble\caption@@declarelabelseparator

441 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
442 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}

caption@setlabelseparator \caption@setlabelseparator{<name>}
Selecting a caption label separator simply means saving the code (in \caption@lsep).
443 \newcommand*\caption@setlabelseparator[1]{%
444   \@ifundefined{caption@lsep@#1}%
445     {\caption@Error{Undefined label separator `#1'}}%
446     {\edef\caption@lsepname{#1}%
447       \expandafter\let\expandafter\caption@ifl@csname\caption@ifl@#1\endcsname
448       \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’.

```

449 \DeclareCaptionLabelSeparator{none}{}
450 \DeclareCaptionLabelSeparator{colon}{: }
451 \DeclareCaptionLabelSeparator{period}{. }
452 \DeclareCaptionLabelSeparator{space}{ }
453 \DeclareCaptionLabelSeparator*{quad}{\quad}
454 \DeclareCaptionLabelSeparator*{newline}{\\}
455 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

‘default’ usually maps to ‘colon’.
456 \def\caption@lsep@default{\caption@lsep@colon}
457 \def\caption@ifl@default{\caption@ifl@colon}

```

1.11 Text formats

```

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

461 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
462 \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).
463 \newcommand*\caption@settextformat[1]{%
464   \@ifundefined{caption@tfmt@#1}%
465     {\caption@Error{Undefined text format `#1'}}%
466     {\expandafter\let\expandafter\caption@tfmt\csname\caption@tfmt@#1\endcsname}}

```

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

```

467 \DeclareCaptionTextFormat{simple}{#1}
468 \DeclareCaptionTextFormat{period}{#1.}

‘default’ usually maps to ‘simple’.
469 \def\caption@tfmt@default{\caption@tfmt@simple}

```

1.12 Fonts

```

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

DeclareDefaultCaptionFont \DeclareDefaultCaptionFont{<name>}{<code>}
473 \newcommand*\DeclareDefaultCaptionFont[2]{%
474   \global\@namedef{caption#1@default}{#2}}
475 \@onlypreamble\DeclareDefaultCaptionFont

476 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
477 \DeclareDefaultCaptionFont{font}{}

478 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
479 \DeclareDefaultCaptionFont{labelfont}{}

480 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
481 \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.
482 \newcommand*\caption@setfont[2]{%
483   \begingroup
484     \let\caption@fnt\@empty
485     \define@key{caption@fnt}{default}[]{%
486       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
487     \caption@setkeys[caption]{caption@fnt}{#2}%
488     \global\let\caption@tempa\caption@fnt
489   \endgroup
490   \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.
491 \newcommand*\caption@font{%
492   \caption@teststar\caption@@font\@firstofone
493     {\caption@setkeys[caption]{caption@fnt}}
494 \newcommand*\caption@@font[2]{%
495   \begingroup
496   \def\caption@fnt{\endgroup}%
497   #1{#2}%
498   \caption@fnt}

```

These are the pre-defined font code snippets.

```

499 \DeclareCaptionFont{normalcolor}{\normalcolor}
500 \DeclareCaptionFont{color}{\color{#1}}

501 \DeclareCaptionFont{normalfont}{\normalfont}
502 \DeclareCaptionFont{up}{\upshape}
503 \DeclareCaptionFont{it}{\itshape}
504 \DeclareCaptionFont{sl}{\slshape}

```

```

505 \DeclareCaptionFont{sc}{\scshape}
506 \DeclareCaptionFont{md}{\mdseries}
507 \DeclareCaptionFont{bf}{\bfseries}
508 \DeclareCaptionFont{rm}{\rmfamily}
509 \DeclareCaptionFont{sf}{\sffamily}
510 \DeclareCaptionFont{tt}{\ttfamily}

511 \DeclareCaptionFont{scriptsize}{\scriptsize}
512 \DeclareCaptionFont{footnotesize}{\footnotesize}
513 \DeclareCaptionFont{small}{\small}
514 \DeclareCaptionFont{normalsize}{\normalsize}
515 \DeclareCaptionFont{large}{\large}
516 \DeclareCaptionFont{Large}{\Large}

517 \DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespace}% normally 1
518 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
519 \DeclareCaptionFont{doublespacing}{\doublespacing}
520 \DeclareCaptionFont{stretch}{\setstretch{#1}}
521 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }

522 %\DeclareCaptionFont{normal}{%
523 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
524 \DeclareCaptionFont{normal}{%
525   \caption@font*{%
526     \KV@caption@fnt@normalcolor\@unused
527     \KV@caption@fnt@normalfont\@unused
528     \KV@caption@fnt@normalsize\@unused
529     \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.

```

530 \providecommand\captionsize{}

531 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
532 \DeclareDefaultCaptionFont{size}{}

```

1.13 Justifications

```

\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
533 \newcommand*\DeclareCaptionJustification[2]{%
534   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
535   \DeclareCaptionFont{#1}{#2}}
536 \@onlypreamble\DeclareCaptionJustification

\DeclareDefaultCaptionJustification{<code>}
537 \newcommand*\DeclareDefaultCaptionJustification[1]{%
538   \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
539   \DeclareDefaultCaptionFont{@hj}{#1}}
540 \@onlypreamble\DeclareDefaultCaptionJustification

541 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
542 \DeclareDefaultCaptionJustification{}

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

```

These are the pre-defined justification code snippets.

```
544 \DeclareCaptionJustification{justified}{}
545 \DeclareCaptionJustification{centering}{\centering}
546 \DeclareCaptionJustification{centerfirst}{\centerfirst}
547 \DeclareCaptionJustification{centerlast}{\centerlast}
548 \DeclareCaptionJustification{raggedleft}{\raggedleft}
549 \DeclareCaptionJustification{raggedright}{\raggedright}
```

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

```
550 \providecommand\centerfirst{%
551   \let\\@centercr
552   \edef\caption@normaladjust{%
553     \leftskip\the\leftskip
554     \rightskip\the\rightskip
555     \parfillskip\the\parfillskip\relax}%
556   \leftskip\z@\@plus -1fil%
557   \rightskip\z@\@plus 1fil%
558   \parfillskip\z@skip
559   \noindent\hskip\z@\@plus 2fil%
560   \@setpar{\@@par\@restorepar\caption@normaladjust}}
```

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

```
561 \providecommand\centerlast{%
562   \let\\@centercr
563   \leftskip\z@\@plus 1fil%
564   \rightskip\z@\@plus -1fil%
565   \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.

```
566 \DeclareCaptionJustification{Centering}{%
567   \caption@ragged\Centering\centering}
568 \DeclareCaptionJustification{RaggedLeft}{%
569   \caption@ragged\RaggedLeft\raggedleft}
570 \DeclareCaptionJustification{RaggedRight}{%
571   \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.

```
572 \newcommand*\caption@ragged{%
573   \caption@Debug{We need ragged2e}%
574   \protected@write\@auxout{}{\string\caption@newlabel{ragged2e}{}%
575     \global\let\caption@ragged\caption@ragged}
576   \caption@ragged}
```

```

577 \caption@AtBeginDocument{%
578   \@ifundefined{caption@r@ragged2e}{%
579     \newcommand*\caption@@ragged{%
580       \PackageWarning{caption}{%
581         'ragged2e' support has been changed.\MessageBreak
582         Rerun to get captions right}%
583       \global\let\caption@ragged\@secondoftwo % suppress further warnings
584       \caption@ragged}%
585   }{%
586     \caption@Debug{We load ragged2e}%
587     \IfFileExists{ragged2e.sty}{%
588       \RequirePackage{ragged2e}%
589       \let\caption@@ragged\@firstoftwo
590     }{%
591       \newcommand*\caption@@ragged[2]{%
592         \@ifundefined{caption\string#1}{%
593           \PackageWarning{caption}{%
594             'ragged2e' package not loaded, therefore\MessageBreak
595             substituting \string#2 for \string#1\MessageBreak}%
596           \global\@namedef{caption\string#1}}{}%
597         #2}%
598     }%
599   }}

```

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.

```

600 \@ifundefined{abovecaptionskip}{%
601   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
602 \@ifundefined{belowcaptionskip}{%
603   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

604 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
605 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
606 \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.

```

607 \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.*

```

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

```

```

610 \let\caption@position\caption@defaultpos
611 }\caption@ifinlist{#1}{t,top,above}{%
612 \let\caption@position\@firstoftwo
613 }\caption@ifinlist{#1}{b,bottom,below}{%
614 \let\caption@position\@secondoftwo
615 }\caption@ifinlist{#1}{a,auto}{%
616 \let\caption@position\@undefined
617 }{%
618 \caption@Error{Undefined position `#1'}%
619 }}}}

```

`\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.

```

620 %\caption@setdefaultpos{a}% default = auto
621 \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.)

```

622 \newcommand*\caption@iftop{%
623 \ifx\caption@position\@undefined
624 \let\caption@position\@secondoftwo
625 % = \caption@setposition b%
626 \fi
627 \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.

```

628 \newcommand*\caption@fixposition{%
629 \ifx\caption@position\@undefined
630 \caption@autoposition
631 \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.

```

632 \newcommand*\caption@autoposition{%
633 \ifvmode
634 \edef\caption@tempa{\the\prevdepth}%
635 \caption@Debug{\protect\prevdepth=\caption@tempa}%
636 \ifdim\prevdepth>-\p@
637 \let\caption@position\@secondoftwo
638 \else
639 \let\caption@position\@firstoftwo

```

```

640 \fi
641 % = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
642 \else
643 \caption@Debug{no \protect\prevdepth}%
644 \let\caption@position\@secondoftwo
645 % = \caption@setposition b%
646 \fi}

```

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

```

647 \newcommand*\caption@setautoposition[1]{%
648 \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`.

```

649 \newcommand*\caption@beginhook{}
650 \newcommand*\caption@endhook{}
651 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
652 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

1.17 Lists

```

653 \DeclareCaptionOption{listof}{\caption@set@bool\caption@iflof{#1}}
\DeclareCaptionListFormat \DeclareCaptionListFormat {<name>}{<code with #1 and #2>}
654 \newcommand*\DeclareCaptionListFormat [2]{%
655 \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
656 \@onlypreamble\DeclareCaptionListFormat
657 \DeclareCaptionOption{listofformat}{\caption@setlistofformat{#1}}

```

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

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

```

658 \newcommand*\caption@setlistofformat[1]{%
659 \@ifundefined{caption@lstfmt@#1}%
660 {\caption@Error{Undefined list-of format `#1'}}%
661 {\expandafter\let\expandafter\caption@lstfmt
662 \csname caption@lstfmt@#1\endcsname}}

```

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

```

663 \DeclareCaptionListFormat{empty}{}
664 \DeclareCaptionListFormat{simple}{#1#2}
665 \DeclareCaptionListFormat{parens}{#1(#2)}
666 \DeclareCaptionListFormat{subsimple}{#2}
667 \DeclareCaptionListFormat{subparens}{(#2)}
668 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}

```

1.18 Debug option

```
669 \DeclareCaptionOption{debug}[1]{%
670   \caption@set@bool\caption@ifdebug{#1}%
671   \let\caption@iftypewarning\caption@ifdebug
672   \caption@ifdebug
673   {\def\caption@Debug{\PackageInfo{caption}}}%
674   {\let\caption@Debug\@gobble}}
675 \DeclareOption{debug}{\setkeys{caption}{debug}}
676 \setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

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

1.19.1 The standard L^AT_EX classes

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

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

```
690 \ifundefined{@captionheadfont}{}{%
691   \caption@CheckCommand\@makecaption{%
692     % amsart|amsproc|amsbook [2004/08/06 v2.20]
693     \long\def\@makecaption#1#2{%
694       \setbox\@tempboxa\vbox{\color@setgroup
695         \advance\hsize-2\captionindent\noindent
696         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
697           {\@cdr#2\@nil}\@captionfont\upshape\enspace#2}%
698       \unskip\kern-2\captionindent\par
699       \global\setbox\@ne\lastbox\color@endgroup}%
700     \ifhbox\@ne % the normal case
701       \setbox\@ne\hbox{\unhbox\@ne\unskip\unpenalty\unkern}%
702     \fi
703     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
704       \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
705     \else % tempboxa contained more than one line
706       \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
707         \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
708     \fi
709     \ifnum\@tempcnta<64 % if the float IS a figure...
710       \addvspace\abovcaptionskip
711       \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
712     \else % if the float IS NOT a figure...
713       \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
714     \fi
715   }%
716 }
```



```

714         \nobreak
715         \vskip\belowcaptionskip
716     \fi
717     \relax
718 }
719 \caption@CheckCommand\@makecaption{%
720     % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publiés par
721     \long\def\@makecaption#1#2{%
722         \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
723         \setbox\@tempboxa\vbox{\color@setgroup
724             \advance\hsize-2\captionindent\noindent
725             \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
726                 {\@cdr#2\@nil}\@addpunct{.}\@captionfont\upshape\enspace#2}%
727             \unskip\kern-2\captionindent\par
728             \global\setbox\@ne\lastbox\color@endgroup}%
729         \ifhbox\@ne % the normal case
730             \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
731         \fi
732         \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
733             \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
734             \@tempdima\wd\@ne\advance\@tempdima-\captionindent
735             \wd\@ne\@tempdima
736         \else % tempboxa contained more than one line
737             \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
738                 \unvbox\@tempboxa\parskip\z@skip
739                 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
740         \fi
741         \ifnum\@tempcnta<64 % if the float IS a figure...
742             \addvspace\abovecaptionskip
743             \noindent\kern\captionindent\box\@ne
744         \else % if the float IS NOT a figure...
745             \noindent\kern\captionindent\box\@ne
746             \nobreak
747             \vskip\belowcaptionskip
748         \fi
749     \relax
750 }
751 \let\captionmargin\captionindent
752 \begingroup\edef\@tempa{\endgroup
753     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
754     {margin=\the\captionmargin
755     \@ifundefined{smf@makecaption}{},{,maxmargin=.1\linewidth}}}
756 \@tempa
757 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
758 \DeclareCaptionLabelSeparator{default}{.\enspace}
759 \DeclareDefaultCaptionFont{font}{\@captionfont}
760 \DeclareDefaultCaptionFont{labelfont}{\@captionheadfont}
761 \DeclareDefaultCaptionFont{textfont}{\@captionfont\upshape}
762 \captionsetup*[figure]{position=b}
763 \captionsetup*[table]{position=t}
764 }

```

1.19.3 The beamer class

```

765 \@ifclassloaded{beamer}{%
766   \caption@CheckCommand\beamer@makecaption{%
767     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
768     \long\def\beamer@makecaption#1#2{%
769       \def\insertcaptionname{\csname#1name\endcsname}%
770       \def\insertcaptionnumber{\csname the#1\endcsname}%
771       \def\insertcaption{#2}%
772       \nobreak\vskip\abovecaptionskip\nobreak
773       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
774       \ifdim \wd\@tempboxa >\hsize
775         \usebeamertemplate**{caption}\par
776       \else
777         \global \@minipagefalse
778         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
779       \fi
780       \nobreak\vskip\belowcaptionskip\nobreak}}
781 \DeclareCaptionLabelFormat{default}{#1}
782 \DeclareDefaultCaptionJustification{\raggedright}
783 \DeclareDefaultCaptionFont{font}{%
784   \usebeamerfont*{caption}%
785   \usebeamerfont*{caption}}
786 \DeclareDefaultCaptionFont{labelfont}{%
787   \usebeamerfont*{caption name}%
788   \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.)

```

789 \defbeamertemplate{caption}{caption3}{%
790   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
791 \DeclareOption{beamer}{%
792   % \usebeamertemplate**{caption} will set font
793   \DeclareDefaultCaptionFont{font}{}%
794   \setbeamertemplate{caption}[caption3]}
795 {}

```

1.19.4 The KOMA-Script classes

```

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

```

```

811 \DeclareCaptionFormat{default}{%
812   \ifdofullc@p
813     \caption@ifin@list\caption@lsep@crlist\caption@lsepname
814     {\caption@Error{%
815       The option 'labelsep=\caption@lsepname' does not work\MessageBreak
816       with \noexpand\setcaphanging (which is set by default)}}%
817     {\caption@fmt@hang{#1}{#2}{#3}}%
818   \else
819     #1#2%
820     \ifdim\cap@indent<\z@
821       \par
822       \noindent\hspace*{-\cap@indent}%
823     \else\if@capbreak
824       \par
825     \fi\fi
826     #3\par
827   \fi}
828 \DeclareCaptionLabelSeparator{default}{\captionformat}
829 \DeclareDefaultCaptionFont{font}{\scr@fnt@caption}
830 \DeclareDefaultCaptionFont{labelfont}{\scr@fnt@captionlabel}
831 }

```

1.19.5 The NTG Dutch classes

```

832 \@ifundefined{CaptionFonts}{}{%
833   \caption@CheckCommand\@makecaption{%
834     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
835     \long\def\@makecaption#1#2{%
836       \vskip\abovecaptionskip
837       \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
838       \ifdim \wd\@tempboxa >\hsize
839         {\CaptionLabelFont#1:} \CaptionTextFont#2\par
840       \else
841         \global \@minipagefalse
842         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
843       \fi
844       \vskip\belowcaptionskip}}
845 \DeclareDefaultCaptionFont{labelfont}{\CaptionLabelFont}
846 \DeclareDefaultCaptionFont{textfont}{\CaptionTextFont}
847 }

```

1.19.6 The frenchb Babel option

```

848 \@ifundefined{FB@makecaption}{}{%
849   \caption@CheckCommand\@makecaption{%
850     % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
851     \long\def\@makecaption#1#2{%
852       \vskip\abovecaptionskip
853       \sbox\@tempboxa{#1\CaptionSeparator #2}%
854       \ifdim \wd\@tempboxa >\hsize
855         #1\CaptionSeparator #2\par
856       \else
857         \global \@minipagefalse
858         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%

```

```

859     \fi
860     \vskip\belowcaptionskip}}

861 \ifx\@makecaption\STD@makecaption
862   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
863   \def\caption@frenchb{% supress frenchb warning
864     \let\STD@makecaption\@makecaption
865     \let\FB@makecaption\@makecaption}
866 \else
867   \ifx\@makecaption\@undefined\else
868     \PackageInfo{caption}{%
869       The definition of \protect\@makecaption\space
870       has been changed,\MessageBreak
871       frenchb will NOT customize it}%
872   \fi
873 \fi
874 }

```

1.19.7 The frenchle/pro package

```

875 \@ifundefined{frenchTeXmods}{}{%
876   \caption@CheckCommand\@makecaption{%
877     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
878     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
879     \def\@makecaption#1#2{%
880       \ifFTY%
881         \def\@secondofmany##1##2\void{##2}%
882         \def\@tempa{\@secondofmany#2\void}%
883         \ifx\@tempa\empty%
884           \let\captionseparator\empty%
885         \fi%
886         \@mcORI{#1}{\relax\captionfont{#2}}%
887       \else
888         \@mcORI{#1}{#2}%
889       \fi}}

890 \caption@CheckCommand\@makecaption{%
891   % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
892   \def\@makecaption#1#2{%
893     \ifFTY%
894       \def\@secondofmany##1##2\void{##2}%
895       \protected@edef\@tempa{\@secondofmany#2\void}%
896       \ifx\@tempa\empty%
897         \let\captionseparator\empty%
898       \fi%
899       \@mcORI{#1}{\relax\captionfont{#2}}%
900     \else
901       \@mcORI{#1}{#2}%
902     \fi}}

903 \DeclareDefaultCaptionFont{textfont}{\itshape}%
904 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
905 }

```

1.20 Execution of options

```

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

908 \ProcessOptions*

909 \caption@ifcheckcommand{%
910   \caption@setbool{documentclass}{1}%
911 }{%
912   \caption@setbool{documentclass}{0}%
913   \PackageInfo{caption}{%
914     Unknown document class (or package), \MessageBreak
915     standard defaults will be used}%
916   \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
917 }

```

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.

918 \newcommand*\caption@addcontentsline[2]{%
919   \caption@iflof%
920   {\def\@tempa{#2}}%
921   {\let\@tempa\@empty}%
922   \ifx\@tempa\@empty \else
923     {\let\\space
924       \addcontentsline{\csname ext@#1\endcsname}{#1}%
925       {\protect\numberline
926         {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
927         {\ignorespaces #2}}}%
928   \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.)

929 \newif\ifcaption@star

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

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

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

932 \newcommand\caption@make[2]{%
933   \caption@@make{\caption@lfmt{#1}{#2}}{\theContinuedFloat}}

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

934 \newcommand\caption@@make[2]{%
935   \begingroup
936   \caption@beginhook

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

937   \ifx\caption@maxmargin\undefined \else
938     \ifdim\captionmargin>\caption@maxmargin\relax
939     \captionmargin\caption@maxmargin\relax

```

```

940     \fi
941 \fi
942 \ifx\caption@minmargin\undefined \else
943     \ifdim\captionmargin<\caption@minmargin\relax
944         \captionmargin\caption@minmargin\relax
945     \fi
946 \fi

```

Special single-line treatment (option singlelinecheck=)

```

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

```

Typeset the left margin (option margin=)

```

949 \caption@calcmargin
950 \@tempdima\captionmargin
951 \ifdim\captionmargin@=\z@ \else
952     \caption@ifoddpagel{\advance\@tempdima\captionmargin@}%
953 \fi
954 \caption@ifh{\advance\@tempdima\caption@indent}%
955 \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*.

```

956 \@tempdima\captionwidth
957 \caption@ifh{\advance\@tempdima-\caption@indent}%
958 \captionbox\@tempdima{%

```

Typeset the indentation (option indentation=)

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

```

959 \caption@ifh{%
960     \ifdim\caption@indent=\z@
961         \leavevmode
962     \else
963         \hskip-\caption@indent
964     \fi}%

```

Typeset the caption itself and close the \captionbox

```

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

```

Typeset the right margin (option margin=)

```

966 \@tempdima\captionmargin
967 \ifdim\captionmargin@=\z@ \else
968     \caption@ifoddpagel{\advance\@tempdima\captionmargin@}}{%
969 \fi
970 \hspace\@tempdima
971 \caption@endhook
972 \endgroup
973 \global\caption@starfalse}

```

\caption@calcmargin \caption@calcmargin

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

```

974 \newcommand*\caption@calcmargin{%
975     \ifdim\captionwidth=\z@
976         \captionwidth\linewidth

```

```

977     \advance\captionwidth by -2\captionmargin
978     \advance\captionwidth by -\captionmargin@
979   \else
980     \captionmargin\linewidth
981     \advance\captionmargin by -\captionwidth
982     \divide\captionmargin by 2
983     \captionmargin@\z@
984   \fi

985   \caption@Debug{%
986     \string\hsize=\the\hsize,
987     \string\linewidth=\the\linewidth, \MessageBreak
988     \string\leftmargin=\the\leftmargin,
989     \string\rightmargin=\the\rightmargin, \MessageBreak
990     \string\margin=\the\captionmargin,
991     \string\margin@=\the\captionmargin@,
992     \string\width=\the\captionwidth}%
993 }

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.
994 \newcommand\caption@slc[4]{%
995   \caption@Debug{Begin SLC}%
996   \begingroup
997   \expandafter\caption@setup\expandafter{\caption@opt@SingleLine}%
998   \let\caption@hj\@empty
999   \caption@calcmargin % calculate #3 if necessary
1000   \caption@prepareslc
1001   \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1002   \ifdim\wd\@tempboxa>#3%
1003     \endgroup
1004   \else
1005     \endgroup
1006     \expandafter\caption@setup\expandafter{\caption@opt@SingleLine}%
1007     #4%
1008   \fi
1009   \caption@Debug{End SLC}}

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1010 \newcommand*\caption@prepareslc{%
1011   \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1012   \def\label{\caption@withoptargs\@gobbletwo}%
1013   \def\stepcounter##1{\advance\c@##1\endcsname\@ne\relax}%
1014   \let\refstepcounter\stepcounter\let\H@refstepcounter\stepcounter}

\captionbox \captionbox{<width>}{<contents>}
This macro defines the box which surrounds the caption paragraph.
1015 \newcommand*\captionbox{\parbox[b]}

\caption@@@make \caption@@@make{<caption label>}{<caption text>}
This one finally typesets the caption paragraph, without margin and indentation.
1016 \newcommand\caption@@@make[2]{%

```

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

```
1017 \sbox\@tempboxa{#1}%
1018 \ifdim\wd\@tempboxa=\z@
1019 \let\caption@lsep\relax
1020 % \@capbreakfalse
1021 \fi
```

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

```
1022 \caption@ifempty{#2}{%
1023 \let\caption@lsep\relax
1024 % \@capbreakfalse
1025 % \let\caption@ifstrut\@secondoftwo
1026 }%
```

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

```
1027 \setpar{\@par\caption@@par}\caption@@par
```

Finally typeset the caption.

```
1028 \caption@hj\captionsize\captionfont\caption@fmt
1029 {\ifcaption@star\else\captionlabelfont#1}\fi}%
1030 {\ifcaption@star\else\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1031 {\captiontextfont
1032 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1033 \nobreak\hskip\z@skip % enable hyphenation
1034 \caption@tfmt{#2}%
1035 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1036 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1037 \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.

```
1038 \newcommand\caption@ifempty[1]{%
1039 \def\caption@tempa{#1}%
1040 \def\caption@tempb{\ignorespaces}%
1041 \ifx\caption@tempa\caption@tempb
1042 \let\caption@tempa\@empty
1043 \fi
1044 \ifx\caption@tempa\@empty
1045 \let\caption@ifempty\@secondoftwo
1046 \else
1047 \let\caption@ifempty\@gobbletwo
1048 \fi
1049 \caption@ifempty\@unused}
```

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

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

```
1050 \newcommand*\caption@@par{%
1051 \parindent\caption@parindent\hangindent\caption@hangindent}%

```


1.22 Bonus features

```

\DeclareFloatingEnvironment[⟨file ext⟩]{⟨environment⟩}[⟨name⟩][⟨list name⟩]
1052 \providecommand*\DeclareFloatingEnvironment{%
1053   \@ifnextchar[%]
1054     \caption@@DeclareFloatingEnvironment\caption@DeclareFloatingEnvironment}
1055 \@onlypreamble\DeclareFloatingEnvironment

1056 \newcommand*\caption@DeclareFloatingEnvironment[1]{%
1057   \caption@@DeclareFloatingEnvironment[lo#1]{#1}}
1058 \@onlypreamble\caption@DeclareFloatingEnvironment

1059 \def\caption@@DeclareFloatingEnvironment[#1]#2{%
1060   \@ifundefined{c@chapter}%
1061     {\newcounter{#2}%
1062      \@namedef{the#2}{\arabic{#2}}%
1063      \@namedef{theH#2}{\arabic{#2}}}%
1064     {\newcounter{#2}[chapter]%
1065      \@namedef{the#2}{\ifnum\c@chapter>\z@ \thechapter.\fi \arabic{#2}}%
1066      \@namedef{theH#2}{\theHchapter.\arabic{#2}}}%
1067   \@namedef{fps@#2}{tbp}%
1068   \@ifundefined{c@float@type}%
1069     {\newcounter{float@type}%
1070      \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1071     {}%
1072   \caption@Debug{float type `#2'=\the\value{float@type}}%
1073   \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1074   \addtocounter{float@type}{\value{float@type}}%
1075   \@namedef{ext@#2}{#1}%
1076   \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1077   \newenvironment{#2}{\@float{#2}}{\end@float}%
1078   \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1079   \expandafter\newcommand\csname listof#2s\endcsname{\begingroup
1080     \expandafter\let\expandafter\listfigurename\csname list#2name\endcsname
1081     \expandafter\let\expandafter\ext@figure\csname ext@#2\endcsname
1082     \let\caption@ORI@starttoc\@starttoc
1083     \renewcommand*\@starttoc[1]{%
1084       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1085     \listoffigures
1086     \endgroup}%
1087   \@ifundefined{l@figure}%
1088     {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1089     {\expandafter\let\csname l@#2\endcsname\l@figure}%
1090   \expandafter\newcommand\csname #2name\endcsname{}%
1091   \edef\@tempa{%
1092     \def\noexpand\@tempa{\expandafter\noexpand\@car#2\@nil}}%
1093   \uppercase\expandafter{\@tempa}%
1094   \edef\@tempb{%
1095     \noexpand\g@addto@macro\noexpand\@tempa{\expandafter\@cdr#2\@nil}}%
1096   \@tempb
1097   \expandafter\let\csname #2name\endcsname\@tempa
1098   \expandafter\newcommand\csname list#2name\endcsname{}%
1099   \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1100   \@cons\caption@floatlist{{#2}}%
1101   \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#2}}\relax}

```

```

1102 \@onlypreamble\caption@@DeclareFloatingEnvironment
1103 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1104   \@namedef{#1name}{#2}%
1105   \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#1}}\relax}
1106 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1107 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1108   \@namedef{list#1name}{#2}}
1109 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1110 \newcommand*\caption@floatlist{}

```

```

\ForFloatingEnvironments \ForFloatingEnvironments{\code with #1}
1111 \providecommand\ForFloatingEnvironments[1]{%
1112   \caption@AtBeginDocument{%
1113     \def\@elt##1{#1}%
1114     \caption@floatlist
1115     \let\@elt\relax}}%
1116 \@onlypreamble\ForFloatingEnvironments

```

1.23 subfig package adoptions

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

```

1117 \caption@AtBeginDocument{%
1118   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1119   \ifx\subref\@tempa
1120     \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1121     \let\caption@setfloattype\@gobble
1122 %   \let\sf@ifinlist\caption@ifinlist
1123   \let\sf@ifpositiontop\caption@iftop
1124   \def\@dottedxxxline#1#2#3#4#5#6#7{%
1125     \begingroup
1126       \caption@setfloattype{#1}%
1127       \caption@setoptions{subfloat}%
1128       \caption@setoptions{sub#1}%
1129       \ifnum #3>\@nameuse{c@#2depth}\else
1130         \@dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1131       \fi
1132     \endgroup}%
1133   \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.

```

1134   \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1135   \caption@removefromoptlist{subfloat}%
1136   \caption@removefromoptlist{subfigure}%
1137   \caption@removefromoptlist{subtable}%
1138 \fi
1139 \let\sf@NEW@subfloat\@undefined}

```

```

1140 \def\sf@NEW@subfloat{%
1141   \begingroup
1142     \caption@setfloattype\@capttype
1143     \sf@ifpositiontop{%
1144       \maincaptiontoptrue
1145     }{%
1146       \maincaptiontopfalse
1147     }%
1148     \caption@setoptions{subfloat}%
1149     \caption@setoptions{sub\@capttype}%
1150     \let\sf@oldlabel=\label
1151     \let\label=\subfloat@label
1152     \ifmaincaptiontop\else
1153       \advance\@nameuse{c@\@capttype}\@ne
1154     \fi
1155     \refstepcounter{sub\@capttype}%
1156     \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1157     \@ifnextchar [% %] match left bracket
1158     {\sf@@subfloat}%
1159     {\sf@@subfloat[\@empty]}}%

```

2 Main package

2.1 Identification

```
1160 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1161 \ProvidesPackage{caption}[2007/09/22 v3.1b 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.

1162 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1163 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1164 \RequirePackage{caption3}[2007/09/16] % needs v3.1a or newer
```

2.3 Check against incompatible packages

```
1165 \@ifpackageloaded{caption2}{%
1166   \caption@Error{%
1167     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1168     the (current) caption package}%
1169   \endinput
1170 }{}

1171 \caption@AtBeginDocument{%
1172   \@ifpackageloaded{ftcap}{%
1173     \caption@IncompatiblePackage{ftcap}{}%
1174   }{}%
1175   \@ifpackageloaded{nonfloat}{%
1176     \caption@IncompatiblePackage{nonfloat}{. %
1177       Furthermore you could use \string\captionof\MessageBreak
1178       instead of \noexpand\figcaption or \string\tabcaption}%
1179   }{}%
1180   \@ifpackageloaded{topcapt}{%
1181     \caption@IncompatiblePackage{topcapt}{}%
1182   }{}%
1183 }

\caption@Incompatible Issue an ‘incompatible’ warning.

1184 \newcommand*\caption@IncompatiblePackage[2]{%
1185   \PackageWarningNoLine{caption}{%
1186     Usage of the #1 package together with the\MessageBreak
1187     caption package is strongly *not* recommended.\MessageBreak
1188     Instead of loading the #1 package you should\MessageBreak
1189     use the caption package option\MessageBreak
1190     \space\space`tableposition=top'\MessageBreak
1191     #2}}

1192 \@onlypreamble\caption@IncompatiblePackage
```

2.4 Check document class

```
1193 \caption@ifbool{documentclass}{}{}%
1194   \PackageWarningNoLine{caption}{%
1195     Unsupported document class (or package) detected,\MessageBreak
```

```

1196     usage of the caption package is *not* recommended}%
1197 \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1198 }

```

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

```

1199 \@ifundefined{@captionheadfont}{}{}%
1200 \caption@Info{AMS or SMF document class}%
1201 \setlength\belowcaptionskip{0pt}% set to 3pc by AMS class
1202 }

```

2.6 Emulation of the KOMA-Script commands

```

1203 \@ifundefined{scr@caption}{}{}%
1204 \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 1205 \def\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1206 \def\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1207 \if@tablecaptionabove
1208 \caption@AtBeginDocument{%
1209 \def\caption@tempa#1{%
1210 \ifpackageloaded{#1}{\caption@Error{%
1211 You can't use the global option\MessageBreak
1212 \space\space'tablecaptionabove'\MessageBreak
1213 and use the #1 package, too}}{}}%
1214 \caption@tempa{ftcap}%
1215 \caption@tempa{nonfloat}%
1216 \caption@tempa{topcapt}}%
1217 \@tablecaptionabovetrue
1218 \else
1219 \@tablecaptionabovefalse
1220 \fi

\onelinecaptionstrue
\onelinecaptionfalse 1221 \def\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1222 \def\onelinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1223 \ifonelinecaptions
1224 \onelinecaptionstrue
1225 \else
1226 \onelinecaptionfalse
1227 \fi

\@captionabovetrue
\@captionabovefalse 1228 \def\@captionabovetrue{\let\caption@position\@firstoftwo}
1229 \def\@captionabovefalse{\let\caption@position\@secondoftwo}

\setcapindent
1230 \let\caption@KOMA@setcapindent\@setcapindent
1231 \renewcommand*\@setcapindent[1]{%
1232 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

```

```

1233 \let\caption@KOMA@@setcapindent\@@setcapindent
1234 \renewcommand*\@@setcapindent[1]{%
1235   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1236 \newcommand*\caption@setcapindent{%
1237   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1238 \@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.)
1239 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1240   \csname\string\setcapwidth\endcsname
1241 \@namedef{\string\setcapwidth}[1]#2{%
1242   \caption@KOMA@setcapwidth[#1]{#2}\caption@setcapwidth{#1}}
1243 \newcommand*\caption@setcapwidth[1]{%
1244   \ifx\#1\\\else
1245     \@ifundefined{cap@margin}{}{%
1246       \def\@tempa{captionbeside}%
1247       \ifx\@tempa\@currentenv\else\PackageWarning{caption}{%
1248         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1249       \fi}%
1250   \fi
1251   \captionsetup{width=\cap@width}}

1252 \def\caption@tempa{\hsize}%
1253 \ifx\caption@tempa\cap@width \else
1254   \caption@setcapwidth{?}
1255 \fi

\setcapmargin
1256 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1257   \csname\string\@setcapmargin\endcsname
1258 \@namedef{\string\@setcapmargin}[1]#2{%
1259   \caption@KOMA@setcapmargin[#1]{#2}\caption@setcapmargin}
1260 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1261   \csname\string\@@setcapmargin\endcsname
1262 \@namedef{\string\@@setcapmargin}[1]#2{%
1263   \caption@KOMA@@setcapmargin[#1]{#2}\caption@setcapmargin}
1264 \newcommand*\caption@setcapmargin{%
1265   \begingroup
1266     \let\onelinecaptionsfalse\relax
1267     \def\@twoside{0}%
1268     \def\if@twoside{\def\@twoside{1}\iffalse}%
1269     \cap@margin
1270     \def\@tempa{\endgroup}%
1271     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1272       \def\hspace##1##{\@firstofone}%
1273       \edef\@tempa{\endgroup
1274         \noexpand\captionsetup{%
1275           twoside=\@twoside,singlelinecheck=0,%
1276           margin={\cap@left,\cap@right}}}%
1277     \fi\fi
1278   \@tempa}

```

```

1279 \ifx\cap@margin\relax \else
1280 \caption@setcapmargin
1281 \fi
1282 }

```

2.7 Option for configuration files

```

1283 \DeclareCaptionOption*{config}[caption]{%
1284 \InputIfFileExists{#1.cfg}%
1285 {\typeout{*** Local configuration file #1.cfg used ***}}%
1286 {\PackageWarning{caption}{Configuration file #1.cfg not found}}}

```

2.8 Options for figure and table

```

1287 \DeclareCaptionOption{figureposition}{%
1288 \captionsetup*[figure]{position=#1}}
1289 \@onlypreamble@key{caption}{figureposition}

1290 \DeclareCaptionOption{tableposition}{%
1291 \caption@AtBeginDocument{%
1292 \def\caption@tempa##1{%
1293 \ifpackageloaded{##1}{\caption@Error{%
1294 You can't use the caption package option\MessageBreak
1295 \space\space`tableposition=#1'\MessageBreak
1296 and use the ##1 package, too}}{}}}%
1297 \caption@tempa{ftcap}%
1298 \caption@tempa{nonfloat}%
1299 \caption@tempa{topcap}}}%
1300 \captionsetup*[table]{position=#1}}
1301 \@onlypreamble@key{caption}{tableposition}

1302 \DeclareCaptionOption{figurename}{\caption@SetFloatName{figure}{#1}}
1303 \@onlypreamble@key{caption}{figurename}
1304 \DeclareCaptionOption{tablename}{\caption@SetFloatName{table}{#1}}
1305 \@onlypreamble@key{caption}{tablename}

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

1306 \newcommand*\caption@SetFloatName[2]{%
1307 \begingroup
1308 \def\@tempa{\endgroup\@firstofone}%
1309 \ifundefined{languagename}{}{%
1310 \ifundefined{captions\languagename}{}{%
1311 \def\@tempa{\endgroup
1312 \expandafter\addto\csname captions\languagename\endcsname}}}%
1313 \@tempa{\@namedef{#1name}{#2}}}
1314 \@onlypreamble\caption@SetFloatName

```

2.9 caption v1.x compatibility options

```

1315 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1316 \@onlypreamble@key{caption}{compatibility}

1317 \DeclareCaptionOption*{normal}[]{%
1318 \caption@setformat{normal}%
1319 \caption@setjustification{justified}}
1320 \DeclareCaptionOption*{isu}[]{%

```

```

1321 \caption@setformat{hang}%
1322 \caption@setjustification{justified}}
1323 \DeclareCaptionOption*{hang}[]{%
1324 \caption@setformat{hang}%
1325 \caption@setjustification{justified}}
1326 \DeclareCaptionOption*{center}[]{%
1327 \caption@setformat{normal}%
1328 \caption@setjustification{centering}}
1329 \DeclareCaptionOption*{anne}[]{%
1330 \caption@setformat{normal}%
1331 \caption@setjustification{centerlast}}
1332 \DeclareCaptionOption*{centerlast}[]{%
1333 \caption@setformat{normal}%
1334 \caption@setjustification{centerlast}}

1335 \DeclareCaptionOption*{scriptsize}[]{\def\captionfont{\scriptsize}}
1336 \DeclareCaptionOption*{footnotesize}[]{\def\captionfont{\footnotesize}}
1337 \DeclareCaptionOption*{small}[]{\def\captionfont{\small}}
1338 \DeclareCaptionOption*{normalsize}[]{\def\captionfont{\normalsize}}
1339 \DeclareCaptionOption*{large}[]{\def\captionfont{\large}}
1340 \DeclareCaptionOption*{Large}[]{\def\captionfont{\Large}}

1341 \DeclareCaptionOption*{up}[]{\l@addto@macro\captionlabelfont\upshape}
1342 \DeclareCaptionOption*{it}[]{\l@addto@macro\captionlabelfont\itshape}
1343 \DeclareCaptionOption*{sl}[]{\l@addto@macro\captionlabelfont\slshape}
1344 \DeclareCaptionOption*{sc}[]{\l@addto@macro\captionlabelfont\scshape}
1345 \DeclareCaptionOption*{md}[]{\l@addto@macro\captionlabelfont\mdseries}
1346 \DeclareCaptionOption*{bf}[]{\l@addto@macro\captionlabelfont\bfseries}
1347 \DeclareCaptionOption*{rm}[]{\l@addto@macro\captionlabelfont\rmfamily}
1348 \DeclareCaptionOption*{sf}[]{\l@addto@macro\captionlabelfont\sffamily}
1349 \DeclareCaptionOption*{tt}[]{\l@addto@macro\captionlabelfont\ttfamily}

1350 \DeclareCaptionOption*{nooneline}[]{\caption@setbool{slc}{0}}

1351 \caption@setbool{ruled}{0}
1352 \DeclareCaptionOption*{ruled}[]{\caption@setbool{ruled}{1}}

```

2.10 caption2 v2.x compatibility options

```

1353 \DeclareCaptionOption*{flushleft}[]{%
1354 \caption@setformat{normal}%
1355 \caption@setjustification{raggedright}}
1356 \DeclareCaptionOption*{flushright}[]{%
1357 \caption@setformat{normal}%
1358 \caption@setjustification{raggedleft}}

1359 \DeclareCaptionOption*{oneline}[]{\caption@setbool{slc}{1}}

1360 \DeclareCaptionOption*{ignoreLTcapwidth}[]{%
1361 \PackageWarning{caption}{Obsolete option 'ignoreLTcapwidth' ignored}}

```

2.11 Obsolete caption v3.0 options

```

1362 \DeclareCaptionOption*{caption}{%
1363 \caption@setbool{temp}{#1}%
1364 \caption@ifbool{temp}{}{%
1365 \caption@Error{%
1366 The package option 'caption=#1' is obsolete.\MessageBreak
1367 Please pass this option to the subfig package instead\MessageBreak

```



```
1368         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.

```
1369 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1370 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1371 \@onlypreamble@key{caption}{FPlist}
1372 \@onlypreamble@key{caption}{FPref}

1373 \newcommand*\caption@setFPoption[2]{%
1374   \edef\@tempa{\@car#2\@nil}%
1375   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1376 \@onlypreamble\caption@setFPoption

1377 \captionsetup{FPlist=caption,FPref=figure}
```

2.13 hyperref package support options

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

```
1378 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1379 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}

1380 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

2.14 Processing of options

```
1381 \caption@ProcessOptions*{caption}
```

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

```
1382 \caption@AtBeginDocument{%
1383   \DeclareCaptionOption{type}{\caption@settype{#1}}
1384   \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 `\@capytype` 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).

```
1385 \newcommand*\caption@settype{%
1386   \caption@@settype{}}

1387 \newcommand*\caption@@settype[1]{%
1388   \caption@teststar{\caption@@@settype{#1}}{\@gobble\@firstofone}}
```

```

1389 \newcommand*\caption@@@settype[3]{%
1390 % #1 = "" or "sub"
1391 % #2 = \@gobble in star form, \@firstofone otherwise
1392 % #3 = <type>, e.g. "figure" or "table"
1393 \caption@Debug{type=#3}%
1394 \caption@checkgrouplevel{#1}{\caption@setup{#1type#2\@gobble*=#3}}%
1395 \@ifundefined{@#1capttype}{}{%
1396 \edef\@tempa{#3}%
1397 \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1398 \caption@iftypewarning{%
1399 \PackageWarning{caption}{%
1400 Caption type was already set to `#3'\MessageBreak}}{}%
1401 \fi}%
1402 \expandafter\edef\csname @#1capttype\endcsname{#3}%
1403 \let\caption@iftypewarning\caption@ifdebug
1404 \@nameuse{#1caption@settypehook}%
1405 \expandafter\caption@setoptions\csname @#1capttype\endcsname
1406 \let\caption@ifrefstepcounter\@secondoftwo
1407 #2{% \@gobble or \@firstofone
1408 \let\@currentlabel\caption@undefinedlabel
1409 % \let\@currentHlabel\@undefined
1410 \ifx\caption@ORI@label\@undefined
1411 \let\caption@ORI@label\label
1412 \def\label{\caption@withoptargs\caption@xlabel}%
1413 \fi
1414 \caption@start}}

```

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

```
1415 \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.

```

1416 \newcommand*\caption@iftype{%
1417 \@ifundefined{@capttype}{\let\@capttype\@undefined\@secondoftwo}\@firstoftwo}

```

`\caption@checkgrouplevel` Checks if `\caption@setup{type=...}` or `\caption` is done inside a group or not – in the latter case an error will be issued. (needs ϵ -TeX)

```

1418 \begingroup\expandafter\expandafter\expandafter\endgroup
1419 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1420 \caption@Debug{TeX engine: TeX}
1421 \let\caption@checkgrouplevel\@gobbletwo
1422 \else
1423 \caption@Debug{TeX engine: e-TeX}
1424 \newcommand*\caption@checkgrouplevel[2]{%
1425 \@ifundefined{#1caption@grouplevel}{%
1426 \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}}{%
1427 \ifnum\currentgrouplevel>\caption@grouplevel\relax
1428 \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1429 \the\currentgrouplevel}%
1430 \else
1431 \caption@Error{\string#2 outside group or environment}%
1432 \fi

```

```

1433     }{} }
1434 \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.
1435 \newcommand*\caption@undefinedlabel{??}

\caption@xlabel
1436 \newcommand*\caption@xlabel[2]{%
1437   \ifx\@currentlabel\caption@undefinedlabel
1438     \PackageWarning{caption}{\noexpand\label before \noexpand\caption ignored}%
1439   \else
1440     \caption@ORI@label#1{#2}%
1441   \fi}

\captionof \captionof{<type>}[<lst.entry>]{<heading>}
\captionof* [<lst.entry>]{<heading>}
1442 \def\captionof{\caption@teststar\caption@of{\caption*}\caption}
1443 \newcommand*\caption@of[2]{%
1444   \let\caption@setfloatcapt\@firstofone
1445   \caption@settype*{#2}#1}

\captionlistentry \captionlistentry[<float type>]{<list entry>}
1446 \newcommand*\captionlistentry{%
1447   \@testopt\caption@listentry\@capttype}
1448 \def\caption@listentry[#1]#2{%
1449   \caption@refstepcounter{#1}%
1450   \caption@makecurrent{#1}{#2}%
1451   \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 pack-
age.
1452 \def\ContinuedFloat{%
1453   \caption@iftype
1454     {\addtocounter\@capttype\m@ne
1455     \caption@ContinuedFloat\@capttype}%
1456     {\caption@Error{\noexpand\ContinuedFloat outside float}}}
1457 \def\caption@ContinuedFloat#1{%
1458   \edef\@tempa{#1}%
1459   \ifx\@tempa\caption@CFtype \else
1460     \caption@Error{Continued `#1' after '\caption@CFtype'}%
1461   \fi
1462   \stepcounter{ContinuedFloat}%
1463   \caption@@ContinuedFloat{#1}%
1464   \caption@setoptions{ContinuedFloat}%
1465   \caption@setoptions{continued#1}%
1466   \let\caption@resetContinuedFloat\@gobble}

```

```

1467 \newcounter{ContinuedFloat}
1468 \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.)
1469 \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.)
1470 \newcommand*\caption@resetContinuedFloat[1]{%
1471 \stpel{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.
1472 \newcommand*\caption@refstepcounter[1]{%
1473 \caption@resetContinuedFloat{#1}%
1474 \caption@@refstepcounter{#1}%
1475 \let\caption@ifrefstepcounter\@firstoftwo}

1476 \newcommand*\caption@@refstepcounter{\refstepcounter}
1477 \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[]{}).
1478 \newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1479 \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)

1480 \newcommand*\caption@begin[1]{%
1481 \begingroup
1482 \caption@setfnum{#1}%
1483 \caption@fixposition
1484 \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>.

```

```

1485 \newcommand\caption@beginex[3]{%
1486   \caption@begin{#1}%
1487   \caption@addcontentsline{#1}{#2}%
1488   \caption@ifempty{#3}{}%

```

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

```

1489 \newcommand*\caption@end{%
1490   \endgroup
1491   \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.

```

1492 \newcommand*\caption@setfnum[1]{%
1493   \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1494   \@namedef{fnum@#1}{\caption@fnum{#1}}%
1495   \fi}

```

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

```

1496 \caption@AtBeginDocument*{%
1497   \@ifundefined{sf@ifpositiontop}{}{%
1498     \let\caption@kernel@addcontentsline\caption@addcontentsline
1499     \renewcommand*\caption@addcontentsline[2]{%
1500       \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
1501       \caption@kernel@addcontentsline{#1}{#2}%
1502       \sf@ifpositiontop{}{\@listsubcaptions{#1}}}%
1503   }}

```

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

```

\def\@parboxrestore{\@arrayparboxrestore\let\\\@normalcr}
\def\@arrayparboxrestore{%
  \let\if@nbreak\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`.

```

1504 \newcommand*\caption@boxrestore{%
1505   \let\if@nbreak\iffalse
1506   \let\if@noskipsec\iffalse
1507   \let\par\@par
1508   \let\-\@dischyph

```

```

1509 % \let'\@acci\let'\@accii\let\=\@acciii
1510 \parindent\z@ \parskip\z@skip
1511 \everypar{ }%
1512 % \linewidth\hsize
1513 % \@totalleftmargin\z@
1514 \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1515 \parfillskip\@flushglue \lineskip\normallineskip
1516 \baselineskip\normalbaselineskip
1517 \sloppy
1518 \let\\\@normalcr
1519 }

```

`\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).

```

1520 \newcommand*\caption@normalsize{%
1521 \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.

```
1522 \let\caption@setfloatcapt\@firstofone
```

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

```

\caption@makestart 1523 \newcommand*\caption@makecurrent[2]{}
\caption@@anchor 1524 \let\caption@makestart\@gobble
\caption@start 1525 \let\caption@@anchor\@firstofone
\caption@@start 1526 \let\caption@start\relax
\caption@freezeHref 1527 \let\caption@@start\relax
\caption@defrostHref 1528 \let\caption@freezeHref\relax
1529 \let\caption@defrostHref\relax

```

2.17 `\caption`, `\@caption`, and `\@makecaption`

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

```

1530 \caption@AtBeginDocument{%
1531 \caption@setbool{incompatible}{0}%
1532 \caption@CheckCommand\caption{%
1533 % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1534 \def\caption{%
1535 \ifx\@capttype\@undefined
1536 \@latex@error{\noexpand\caption outside float}\@ehd

```

```

1537         \expandafter\@gobble
1538     \else
1539         \refstepcounter\@capttype
1540         \expandafter\@firstofone
1541     \fi
1542     {\@dblarg{\@caption\@capttype}}%
1543 } }%

1544 \caption@CheckCommand\caption{%
1545 % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1546 \def\caption{
1547     \ifx\@capttype\@undefined
1548     \@latex@error{\noexpand\caption outside figure or table}\@ehd
1549     \expandafter\@gobble
1550     \else
1551     \refstepcounter\@capttype
1552     \expandafter\@firstofone
1553     \fi
1554     {\@dblarg{\@caption\@capttype}}%
1555 } }%

1556 \caption@CheckCommand\caption{%
1557 % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1558 \renewcommand\caption{%
1559     \ifx\@capttype\@undefined
1560     \@latex@error{\noexpand\caption outside float}\@ehd
1561     \expandafter\@gobble
1562     \else
1563     \refstepcounter\@capttype
1564     \let\@tempf\@caption
1565     \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1566     \expandafter\expandafter\let
1567     \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1568     \fi
1569     \fi
1570     \@dblarg{\@tempf\@capttype}}}%

1571 \caption@CheckCommand\caption{%
1572 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1573 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1574 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1575 \def\caption{%
1576     \ifx\@capttype\@undefined
1577     \@latex@error{\noexpand\caption outside float}\@ehd
1578     \expandafter\@gobble
1579     \else
1580     \H@refstepcounter\@capttype
1581     \@ifundefined{fst@\@capttype}{%
1582         \let\Hy@tempa\@caption
1583     }{%
1584         \let\Hy@tempa\Hy@float@caption
1585     }%
1586     \expandafter\@firstofone
1587     \fi
1588     {\@dblarg{\Hy@tempa\@capttype}}%
1589 } }%

```

```

1590 \caption@CheckCommand\caption{%
1591   % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1592   \def\caption{%
1593     \ifx\@capttype\@undefined
1594       \@latex@error{\noexpand\caption outside float}\@ehd
1595       \expandafter\@gobble
1596     \else
1597       \H@refstepcounter\@capttype
1598       \let\Hy@tempa\@caption
1599       \@ifundefined{float@caption}{%
1600         }{%
1601           \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
1602           \let\Hy@tempa\Hy@float@caption
1603           \fi
1604         }%
1605       \expandafter\@firstofone
1606     \fi
1607     {\@dblarg{\Hy@tempa\@capttype}}%
1608   }%
1609 \caption@ifCheckCommand{}{%
1610   \caption@Info{%
1611     Incompatible package detected (regarding \string\caption).\MessageBreak
1612     \string\caption\space=\space meaning\caption}%
1613   \caption@setbool{incompatible}{1}}%
1614 \caption@CheckCommand\@caption{%
1615   % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1616   \long\def\@caption#1[#2]#3{%
1617     \par
1618     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1619     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1620     \begingroup
1621       \@parboxrestore
1622       \if@minipage
1623         \@setminipage
1624       \fi
1625       \normalsize
1626       \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1627     \endgroup}%
1628 \caption@CheckCommand\@caption{%
1629   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1630   \long\def\@caption#1[#2]#3{% second argument ignored
1631     \par\nobreak
1632     \begingroup
1633       \@parboxrestore
1634       \if@minipage
1635         \@setminipage
1636       \fi
1637       \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
1638     \endgroup}%
1639 % \caption@CheckCommand\float@caption{%
1640 %   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1641 %   \long\def\float@caption#1[#2]#3{%
1642 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%

```



```

1643 %         {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
1644 %         \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
1645 %         \normalsize\@fs@capt{\@nameuse{fnum#1}}{\ignorespaces #3}%
1646 %         \@ifnextchar[{\float@ccon}{\egroup}}%
1647 %         \long\def\float@ccon[#1]{#1\par\egroup}}%

1648 \caption@CheckCommand\@caption{%
1649 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1650 \long\def\@caption#1[#2]#3{%
1651   \hyper@makecurrent{\@capttype}%
1652   \def\@currentlabelname{#2}%
1653   \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1654     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1655   }%
1656   \begingroup
1657     \@parboxrestore
1658     \if@minipage
1659       \setminipage
1660     \fi
1661     \normalsize
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   \par
1671   \endgroup
1672 }}%

1673 \caption@CheckCommand\@caption{%
1674 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1675 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1676 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1677 \long\def\@caption#1[#2]#3{%
1678   \expandafter\ifx\csname if@capstart\expandafter\endcsname
1679     \csname iftrue\endcsname
1680     \global\let\@currentHref\hc@currentHref
1681   \else
1682     \hyper@makecurrent{\@capttype}%
1683   \fi
1684   \def\@currentlabelname{#2}%
1685   \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1686     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1687   }%
1688   \begingroup
1689     \@parboxrestore
1690     \if@minipage
1691       \setminipage
1692     \fi
1693     \normalsize
1694     \expandafter\ifx\csname if@capstart\expandafter\endcsname
1695       \csname iftrue\endcsname

```

```

1696         \global\@capstartfalse
1697         \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1698     \else
1699         \@makecaption{\csname fnum@#1\endcsname}{%
1700             \ignorespaces
1701             \ifHy@nesting
1702                 \hyper@@anchor{\@currentHref}{#3}%
1703             \else
1704                 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1705             \fi
1706         }%
1707     \fi
1708     \par
1709 \endgroup
1710 }%

1711 \caption@CheckCommand\@caption{%
1712     % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1713     \long\def\@caption#1[#2]{%
1714         \def\@currentlabelname{#2}%
1715         \NR@@caption{#1}[{#2}]%
1716     }%

1717 \caption@CheckCommand\@caption{%
1718     % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1719     \long\def\@caption#1[#2]#3{%
1720         \ifundefined{if#1topcap}%
1721             {\subfig@oldcaption{#1}[{#2}]{#3}}%
1722             {\@nameuse{if#1topcap}%
1723                 \@listsubcaptions{#1}%
1724                 \subfig@oldcaption{#1}[{#2}]{#3}%
1725             \else
1726                 \subfig@oldcaption{#1}[{#2}]{#3}%
1727                 \@listsubcaptions{#1}%
1728             \fi}}%

1729 \caption@CheckCommand\@caption{%
1730     % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1731     \def\@caption{\caption@}%
1732 % \long\def\caption@#1[#2]#3{%
1733 %     \ifundefined{caption@setfloattype}%
1734 %         \caption@settype
1735 %         \caption@setfloattype
1736 %         \@capttype
1737 %         \sf@ifpositiontop{%
1738 %             \@listsubcaptions{#1}%
1739 %             \sf@old@caption{#1}[{#2}]{#3}%
1740 %         }{%
1741 %             \sf@old@caption{#1}[{#2}]{#3}%
1742 %             \@listsubcaptions{#1}%
1743 %         }%
1744 %     }%

1745 \caption@IfCheckCommand{}{%
1746     \caption@Info{%
1747         Incompatible package detected (regarding \string\@caption).\MessageBreak
1748         \string\@caption\space=\space\meaning\@caption}%

```

```
1749 \caption@setbool{incompatible}{1}}%
```

The option compatibility= will override the compatibility mode.

```
1750 \@ifundefined{caption@ifcompatibility}%
1751 {\let\caption@ifcompatibility\caption@ifincompatible
1752 \let\@tempa\@firstofone}%
1753 {\let\@tempa\@gobble}% suppress warning
1754 \caption@ifcompatibility{%
1755 \@tempa{%
1756 \PackageWarningNoLine{caption}{%
1757 \noexpand\caption will not be redefined since it's already\MessageBreak
1758 redefined by a document class or package which is\MessageBreak
1759 unknown to the caption package.\MessageBreak
1760 As a result some new features of the caption\MessageBreak
1761 package will not work as expected}%
1762 }%
```

\ContinuedFloat is not supported in compatibility mode.

```
1763 \renewcommand*\caption@ContinuedFloat[1]{%
1764 \caption@Error{Not available in compatibility mode}}%
```

\caption@start is not supported in compatibility mode.

```
1765 \caption@AtBeginDocument*{%
1766 \let\caption@start\relax
1767 \@ifundefined{caption@ORI@capstart}{}{%
1768 \caption@Debug{%
1769 Undo redefinition of \string\capstart\@gobble}%
1770 \let\capstart\caption@ORI@capstart
1771 \let\caption@ORI@capstart\@undefined}%
1772 \@ifundefined{Hy@float@makebox}{}{%
1773 \caption@Debug{%
1774 Restore hyperref redefinition of \string\float@makebox\@gobble}%
1775 \let\float@makebox\Hy@float@makebox
1776 \let\Hy@float@makebox\@undefined}%
1777 }%
```

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

```
1778 \newcommand*\caption@star[2]{#1#2}%
1779 }{%
1780 \caption@ifincompatible{%
1781 \PackageWarning{caption}{%
1782 Forced redefinition of \noexpand\caption since the\MessageBreak
1783 unsupported(!) package option 'compatibility=false'\MessageBreak
1784 was given. Please fasten your seat belts, as we may\MessageBreak
1785 experience turbulence!\@gobbletwo}%
1786 }{}%
```

\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.

```
1787 \def\caption{%
```

```

1788     \caption@iftype
1789     {\caption@checkgrouplevel\@empty\caption
1790     \caption@star
1791     {\caption@refstepcounter\@capttype}%
1792     {\caption@dblarg{\@caption\@capttype}}}%
1793     {\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.
1794     \newcommand*\caption@star[2]{%
1795     \ifstar{\global\caption@startrue#2[]}{#1#2}}%

\@caption As above, our version has been adapted to the packages we support. Addition-
ally our code is nested by \caption@beginex & \caption@end instead of
\begingroup & \endgroup. Furthermore we use \caption@boxrestore in-
stead of \@parboxrestore so this code also works correctly inside list-based environ-
ments like wide & addmargin. (This, and the fact that we use \linewidth instead
of \hsize inside \@makecaption, solves LATEX PR latex/2472.)
1796     \long\def\@caption#1[#2]#3{%
1797     \ifcaption@star \else
1798     \caption@makecurrent{#1}{#2}%
1799     \caption@ifhypcap{\caption@@start}{}%
1800     \fi
1801     \par
1802     \caption@beginex{#1}{#2}{#3}%
1803     \caption@setfloatcapt{%
1804     \caption@boxrestore
1805     \caption@normalsize
1806     \ifcaption@star
1807     \let\caption@@anchor\@firstofone
1808     \fi
1809     \@makecaption{\csname fnum@#1\endcsname}%
1810     {\ignorespaces\caption@@anchor{#3}}\par}%
1811     \caption@end}%
1812 }%

\@xfloat We redefine \@xfloat so inside floating environments our type-specific options will be
used, a hyperref anchor will be set etc.
1813 \let\caption@ORI@xfloat\@xfloat
1814 \def\@xfloat#1[#2]{%
1815 \caption@ORI@xfloat{#1}[#2]%
1816 \caption@settype{#1}}%
1817 }

\@makecaption \@makecaption{<label>}{<text>}
We do basically the same as the original code (from the standard LATEX document classes),
but take care of the position= setting and use \caption@@make from the caption
kernel to finally typeset the caption.
1818 \long\def\@makecaption#1#2{%
1819 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
1820 \caption@@make{#1}{#2}%
1821 \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

```
1822 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}
```

2.18.2 The beamer class

```
1823 \@ifclassloaded{beamer}{%
1824   \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.

```
1825   \captionsetup{listof=false}
1826   \undefine@key{caption}{listof}
1827   \undefine@key{caption}{listofformat}
1828 }{ }
```

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.

```
1829 \@ifundefined{scr@caption}{}{%
1830   \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)

```
1831 \@ifundefined{caption@frenchb}{}{\caption@frenchb}
```

2.18.5 The frenchle/pro package

```
1832 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
1833   \caption@Info{frenchle/pro package is loaded}%
1834   \let\captionfont@ORI\captionfont
1835   \let\captionlabelfont@ORI\captionlabelfont
1836   \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`.

```
1837   \@ifundefined{GOfrench}%
1838     {\let\@tempa\@firstofone}%
1839     {\def\@tempa{\g@addto@macro\GOfrench}}%
1840   \@tempa{%
1841     \let\captionfont\captionfont@ORI
1842     \let\captionfont@ORI\@undefined
1843     \let\captionlabelfont\captionlabelfont@ORI
1844     \let\captionlabelfont@ORI\@undefined
1845     \let\@makecaption\@makecaption@ORI
1846     \let\@makecaption@ORI\@undefined
```

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

```
1847   \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.
1848 \let\caption@tcORI\@tablescaption
1849 \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.

1850 \let\eatDP\@undefined
1851 \let\@tempa\@empty
1852 \ifx\ffrench\figure
1853 \l@addto@macro\@tempa{\let\fnum@figure\ffrench}%
1854 \fi
1855 \ifx\tfrench\table
1856 \l@addto@macro\@tempa{\let\fnum@table\tfrench}%
1857 \fi
1858 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
1859 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
1860 \@tempa

1861 }%
1862 }

```

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.

1863 \newcommand\caption@ifpackageloaded[1]{%
1864 \testopt{\caption@ifpackageloaded{#1}}{}%
1865 \onlypreamble\caption@ifpackageloaded

1866 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
1867 \ifpackageloaded{#1}\@firstofone{%
1868 \caption@Debug{#1 package is not loaded (yet)\@gobble}%
1869 \caption@AtBeginDocument}{%
1870 \caption@ifpackageloaded{#1}[#2]{#3}{#4}}%
1871 \onlypreamble\caption@ifpackageloaded

1872 \newcommand\caption@ifpackageloaded[1]{%
1873 \testopt{\caption@ifpackageloaded{#1}}{}%
1874 \onlypreamble\caption@ifpackageloaded

1875 \long\def\caption@ifpackageloaded#1[#2]{%
1876 \ifpackageloaded{#1}{%
1877 \caption@Info{#1 package is loaded}%
1878 \ifpackagelater{#1}{#2}\@firstoftwo{%
1879 \caption@Error{%
1880 For a successful cooperation we need at least version\MessageBreak
1881 `#2' of package #1,\MessageBreak
1882 but only version\MessageBreak

```

```

1883         \csname ver@#1.\@pkgextension\endcsname'\MessageBreak
1884         is available}%
1885         \@secondoftwo}%
1886     }\@secondoftwo}}
1887 \@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.

```

1888 \newcommand*\caption@clearmargin{%
1889     \setcaptionmargin\z@
1890     \let\caption@minmargin\@undefined}

1891 \caption@setbool{needfreeze}{0}
1892 \caption@AtBeginDocument*{%
1893     \caption@ifneedfreeze{%

```

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

```

1894 \newcommand*\caption@freeze{%
1895     \caption@teststar\caption@@freeze\@gobble\@firstofone}%

1896 \newcommand*\caption@@freeze[1]{%
1897     \global\let\caption@SCcontinued\relax
1898     \global\let\caption@SCsetup\@undefined
1899     \global\let\caption@SClentry\@undefined
1900     \global\let\caption@SCtext\@undefined
1901     \global\let\caption@SClabel\@undefined

1902     \let\caption@ORI@ContinuedFloat\ContinuedFloat
1903     \def\ContinuedFloat{%
1904         \let\caption@ORI@setcounter\setcounter
1905         \let\caption@ORI@addtocounter\addtocounter
1906         \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
1907         \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
1908         \caption@ORI@ContinuedFloat
1909         \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
1910         \let\setcounter\caption@ORI@setcounter
1911         \let\addtocounter\caption@ORI@addtocounter}%
1912     \def\captionsetup##1{%
1913         \caption@g@addto@list\caption@SCsetup{##1}%
1914         \caption@setup{##1}}%
1915     \def\caption{%
1916         \let\captionsetup\@gobble
1917         \caption@star
1918         {\let\caption@ORI@refstepcounter\caption@@refstepcounter
1919         \def\caption@@refstepcounter#####1{%
1920             \advance\csname c@#####1\endcsname\@ne\relax}%
1921             \caption@refstepcounter\@captype
1922             \let\caption@@refstepcounter\caption@ORI@refstepcounter}%
1923         {\caption@dblarg\caption@SC}}%
1924     \def\@caption@SC[##1]##2{%
1925         \gdef\caption@SClentry{##1}\gdef\caption@SCtext{##2}%
1926         \ignorespaces}%
1927     #1{% is \@gobble in star form, and \@firstofone otherwise
1928         \def\label##1{\gdef\caption@SClabel{##1}\ignorespaces}}%

```

```

1929 }%

\caption@defrost \caption@defrost
1930 \newcommand*\caption@defrost{%
1931   \ifx\caption@ORI@ContinuedFloat\@undefined
1932     \caption@SCcontinued
1933     \ifx\caption@SCsetup\@undefined \else
1934       \expandafter\captionsetup\expandafter{\caption@SCsetup}%
1935     \fi
1936     \ifx\caption@SCtext\@undefined \else
1937       \expandafter\expandafter\expandafter\caption
1938       \expandafter\expandafter\expandafter[%
1939       \expandafter\expandafter\expandafter[%
1940       \expandafter\caption@SClentry\expandafter]\expandafter]%
1941       \expandafter{\caption@SCtext}%
1942     \fi
1943     \ifx\caption@SClabel\@undefined \else
1944       \expandafter\label\expandafter{\caption@SClabel}%
1945     \fi
1946   \else
1947     \caption@Error{Internal Error:\MessageBreak
1948       \noexpand\caption@defrost in same group as \string\caption@freeze}%
1949   \fi}%

1950 }{}%
1951 \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.)

```

1952 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
1953   \@ifpackageloaded{floatrow}{%
1954     \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
1955   }{}%

```

`\@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>]`.

```

1956 \let\caption@ORI@float@setevery\@float@setevery
1957 \def\@float@setevery#1{%
1958   \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.

```
1959 \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.

```
1960 \renewcommand\caption@setfloatcapt{%
1961 \let\@makecaption\caption@@make
1962 \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.

```
1963 \float@getstyle\float@style{#1}%
1964 \caption@setstyle*\float@style
1965 \caption@setoptions\float@style
1966 }{}%
1967 \caption@freezeHref % will be defrosted in \float@makebox
1968 \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.

```
1969 % \renewcommand*\caption@setfloatcapt{%
1970 % \float@ifcaption\@capttype{%
1971 % \caption@Error{Something is going wrong here...\MessageBreak
1972 % For a removal of this problem please consult the\MessageBreak
1973 % caption package documentation, especially the sections\MessageBreak
1974 % about the 'float' and the 'wrapfig' package}%
1975 % \@gobble
1976 % }{}%
1977 % \@firstofone
1978 % }{}%
```

`\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.

```
1979 \g@addto@macro\caption@settypehook{%
```

```

1980     \expandafter\ifx\csname #1name\endcsname\relax
1981     \expandafter\let\csname #1name\expandafter\endcsname
1982         \csname fname@#1\endcsname
1983     \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.

1984 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
1985 \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).

1986 \providecommand*\float@ifstyle[1]{%
1987     \expandafter\ifx\csname fst@#1\endcsname\relax
1988     \expandafter\@secondoftwo
1989     \else
1990     \expandafter\@firstoftwo
1991     \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.

1992 \providecommand*\float@getstyle[2]{%
1993     \edef#1{%
1994         \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
1995         \expandafter\expandafter\expandafter\noexpand
1996         \csname fst@#2\endcsname}%
1997     \edef#1{#1}%
1998     \caption@Debug{floatstyle{#2} = '#1'}}%

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

1999 \providecommand*\float@setstyle[2]{%
2000     \expandafter\edef\csname fst@#1\endcsname{%
2001         \expandafter\noexpand\csname fs@#2\endcsname}}%

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

2002 \providecommand*\float@dostyle[1]{%
2003     \@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.)
2004 \providecommand*\float@ifcaption[1]{%
2005   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2006   \expandafter\@firstoftwo
2007   \else
2008   \expandafter\@secondoftwo
2009   \fi}%

2010 }{%
2011 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2012 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2013 % \clearcaptionsetup{boxed}% used by the floatrow package?
2014 }

```

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

```
2015 \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’.

```

2016 \caption@ifbool{ruled}{%
2017   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
2018 }{%
2019   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2020 }
2021 \caption@undefbool{ruled}

```

2.19.2 The floatflt package

```
2022 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

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

```

2023 \let\caption@ORI@floatingfigure\floatingfigure
2024 \def\floatingfigure{%
2025   \caption@floatflt{figure}%
2026   \caption@ORI@floatingfigure}%

```

\floatingtable Same with \floatingtable...

```

2027 \let\caption@ORI@floatingtable\floatingtable
2028 \def\floatingtable{%
2029   \caption@floatflt{table}%
2030 % \caption@setautoposition b%
2031   \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.

```
2032 \newcommand*\caption@floatflt[1]{%
2033   \caption@settype{#1}%
2034   \caption@setoptions{floating#1}%
2035   \let\caption@boxrestore\@parboxrestore}%

2036 }{ }
```

2.19.3 The fltpage package

```
2037 \caption@ifpackageloaded{fltpage}[1998/10/29 v.0.3]{%
2038   \caption@setbool{needfreeze}{1}%
```

`\FP@helpNote` Original code:

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

```
2039 \renewcommand\FP@helpNote[2]{%
2040   \begingroup % save \caption@thepage
2041     \caption@pageref{#2}%
2042     \typeout{FP#1 is inserted on page \caption@thepage!}%
2043   \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\fnun@figure}%
    {\global\let\old@Fnum\fnun@table}%
  }
  \let\FP@LabelText\@empty%
  \let\FP@CaptionText\@empty%
  \let\FP@optionalCaptionText\@empty%
  \renewcommand\label[1]{\gdef\FP@LabelText{##1}}%
  \renewcommand\caption[2][1]{%
    \gdef\FP@optionalCaptionText{##1}\gdef\FP@CaptionText{##2}}%
  \begin{lrbox}{\FP@floatCorpusBOX}%
  }%
```

```
2044 \renewcommand*\FP@floatBegin[1]{%
2045   \def\@captive{#1}%
2046   \let\FP@LabelText\@empty
2047   \begin{lrbox}{\FP@floatCorpusBOX}%
2048     \caption@ifFPrefcap\caption@freeze{%
2049       \def\label##1{\gdef\FP@LabelText{##1}\ignorespaces}%
2050     \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}}%
}%

2051 \renewcommand*\FP@floatEnd{%
2052   \end{lrbox}%
2053   \stepcounter{FP@{\@cptype} C}%
2054   \caption@label\FP@positionLabel
2055   \FP@helpNote{\@cptype}\FP@positionLabel
2056   \let\FP@ContinuedFloat\relax
2057   \FP@float
2058     {\FP@positionLabel}% location label test
2059     {\begin{\@cptype}[p!]}%
2060       \FP@ContinuedFloat
2061       \usebox\FP@floatCorpusBOX
2062       \caption@ifFlistcap
2063         {\caption@refstepcounter{\@cptype}
2064           \expandafter\caption@makecurrent\expandafter{\@cptype}
2065             \expandafter{\caption@SClentry}}}%
2066         {\expandafter\captionlistentry\expandafter{\caption@SClentry}}}%
2067       \caption@@anchor\relax
2068       \ifx\FP@LabelText\@empty \else
2069         \expandafter\label\expandafter{\FP@LabelText}%
2070       \fi
2071     \end{\@cptype}%
2072   {\let\FP@ContinuedFloat\ContinuedFloat
2073     \caption@ifcompatibility

```

```

2074         {\let\caption@ContinuedFloat\@gobble}%
2075         {\caption@clearsetup{ContinuedFloat}}%
2076         \caption@clearsetup{continued\@capttype}}}%
2077     {\begin\@capttype[b!]}%
2078     \FP@ContinuedFloat
2079     \let\FP@savedSetfnumCommand\caption@setfnum
2080     \def\caption@setfnum##1{%
2081         \FP@savedSetfnumCommand{##1}%
2082         \ifx\FP@guide\@empty \else
2083             \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}%
2084         \fi}%
2085     \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2086     \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2087     \caption@setoptions{FP\@capttype}%
2088     \FP@separatorCaption
2089     \caption@ifFPlistcap{}\{\caption@set@bool\caption@iflof0}%
2090     \caption@defrost
2091     \end\@capttype}%
2092 }%

2093 \ForFloatingEnvironments{%
2094     \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2095 }{%
2096     \let\caption@ifFPlistcap\@undefined
2097     \let\caption@ifFPrefcap\@undefined
2098 }

```

2.19.4 The hyperref package

```

2099 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2100     \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2101         \PackageWarningNoLine{caption}{%
2102             Hyperref support is turned off\MessageBreak
2103             because hyperref has stopped early}%
2104     }{%
2105         \g@addto@macro\caption@prepareslc{\measuring@true}%

```

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

```

2106     \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

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

```

2107     \renewcommand*\caption@makecurrent[2]{%
2108         \caption@makecurrentHref{#1}%
2109         \caption@Debug{hyper current=\@currentHref}%
2110         \def\@currentlabelname{#2}}%
2111     \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.

```
2112 \renewcommand*\caption@makestart[1]{%
2113 \begingroup
2114 \Hy@hypertextnamesfalse
2115 % \gdef\@currentHlabel{}%
2116 \hyper@makecurrent{#1.xxx}%
2117 \endgroup
2118 \caption@Debug{hyper start=\@currentHref}}%
```

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

```
2119 \renewcommand*\caption@@anchor[1]{%
2120 \caption@Debug{hyper anchor: \@currentHref}%
2121 % If we cannot have nesting, the anchor is empty.
2122 \ifHy@nesting
2123 \hyper@@anchor{\@currentHref}{#1}%
2124 \else
2125 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2126 \fi}%
2127 \g@addto@macro\caption@prepareslc{\let\caption@@anchor\@firstofone}%
```

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

This code was taken from the `hycap` package[10] and adapted.

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

```
2128 \newcommand*\caption@startanchor[1]{%
2129 \caption@Debug{hyper anchor: #1}%
2130 \ifvmode\begingroup
2131 \@tempdima\prevdepth
2132 \nointerlineskip
2133 \vspace*{-\caption@hycapsspace}%
2134 \caption@anchor{#1}%
2135 \vspace*{\caption@hycapsspace}%
2136 \prevdepth\@tempdima
2137 \endgroup\else
2138 \PackageWarning{caption}{Internal Warning: Vertical TeX mode expected}%
2139 \caption@anchor{#1}%
2140 \fi}%
```

`\caption@anchor` Sets a `hyperref` anchor.

```

2141 \newcommand*\caption@anchor[1]{%
2142 \ifmeasuring@ \else
2143 \Hy@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2144 \fi}%

```

`\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.

```

2145 \renewcommand*\caption@@ContinuedFloat[1]{%
2146 \ifundefined{theH#1}{}{%
2147 \expandafter\l@addto@macro\csname theH#1\endcsname{%
2148 \alpha\c@ContinuedFloat}}}%

```

```

2149 \newif\if@capstart

```

`\caption@start` This one emulates the `hycap` package (if `hycap=true`).

```

2150 \def\caption@start{\caption@ifhycap{%

```

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

```

2151 \caption@makestart\@captype
2152 \caption@startanchor\@currentHref

```

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

```

2153 \global\@capstarttrue
2154 \let\hc@currentHref\@currentHref
2155 \def\caption@makecurrentHref##1{%
2156 \global\@capstartfalse
2157 \global\let\@currentHref\hc@currentHref}%

```

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

```

2158 \let\caption@@anchor\@firstofone
2159 %%% \let\caption@startanchor\@gobble
2160 }{}%

```

`\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.

```

2161 \def\caption@@start{%
2162 \ifundefined{hc@currentHref}{}%
2163 \PackageWarning{caption}{%
2164 The option 'hycap=true' will be ignored for this\MessageBreak
2165 particular \string\caption}}}%

```


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

```

2166 \def\caption@freezeHref{%
2167   \let\caption@ORI@start\caption@start
2168   \def\caption@start{\let\caption@start\caption@ORI@start}%
2169   \let\caption@ORI@@start\caption@@start
2170 % \l@addto@macro\subcaption@settypehook{%
2171 %   \let\caption@@start\caption@ORI@@start}%
2172 \global\let\caption@currentHref\@undefined
2173 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2174 \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2175 \renewcommand*\caption@setfloatcapt{%
2176   \ifx\caption@currentHref\@undefined \else
2177     \let\caption@@anchor\@firstofone
2178   \fi
2179   \caption@ORI@setfloatcapt}}%

```

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

```

2180 \def\caption@defrostHref{%
2181   \ifx\caption@currentHref\@undefined \else
2182     \caption@startanchor\caption@currentHref
2183     \global\let\caption@currentHref\@undefined
2184   \fi}%

```

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

```

2185 \@ifundefined{HyOrg@float@makebox}{}{%
2186   \caption@Debug{%
2187     Redefining \noexpand\float@makebox (again)\@gobble}%
2188   \let\Hy@float@makebox\float@makebox
2189   \renewcommand\float@makebox[1]{%
2190     \HyOrg@float@makebox{#1\relax \caption@defrostHref}}%
2191   }%
2192 }}}

```

2.19.5 The `hypcap` package

```

2193 \caption@IfPackageLoaded{hypcap}{% v1.0
2194   \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.

```

2195   \let\caption@ORI@capstart\capstart
2196   \let\capstart\caption@start
2197   \let\caption@start\relax
2198   \let\caption@@start\relax

```

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

```

2199 \renewcommand*\caption@hypcapospace{\hypcapospace}%

\float@makebox Redo the redefinition of \float@makebox, if necessary.
2200 \ifundefined{Hy@float@makebox}{}{}%
2201 \caption@Debug{%
2202     Restore hyperref redefinition of \string\float@makebox\@gobble}%
2203 \let\float@makebox\Hy@float@makebox
2204 \let\Hy@float@makebox\@undefined}%
2205 \fi}{}

```

2.19.6 The listings package

```

2206 \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.
2207 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2208 \def\lst@MakeCaption#1{%
2209     \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.

```

2210 \caption@setautoposition{#1}%
2211 \caption@begin{lstlisting}%
2212 \caption@ORI@lst@MakeCaption{#1}%
2213 \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.
2214 \providecommand*\ext@lstlisting{lol}%
2215 }{}

```

2.19.7 The longtable package

```

\LTcaption \LTcaption is preset to table.
2216 \providecommand*\LTcaption{table}

2217 \caption@IfPackageLoaded{longtable}[1995/05/24 v3.14]{}%
2218 \RequirePackage{ltcaption}[2007/09/01]%
2219 \let\LT@makecaption\@undefined

\LT@array We redefine \LT@array here to get \captionsetup{\<options>} working inside
longtables.
Note: Since the hyperref package patches \LT@array as well and since this only works
with the original definition of \LT@array, we have to do this after the hyperref package,
i.e. \AtBeginDocument.
2220 \caption@AtBeginDocument{%

```

```

2221 \let\caption@ORI@LT@array\LT@array
2222 \renewcommand*\LT@array{%
\captionsetup for longtable:
2223 \global\let\caption@opt@@longtable\@undefined
2224 \def\captionsetup##1{\noalign{\LT@captionsetup{##1}}}%
2225 \def\LT@captionsetup##1{%
2226 \captionsetup\@gobble[@longtable]{##1}%
2227 \global\let\caption@opt@@longtable\caption@opt@@longtable}%
\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2228 \def\captionabovetrue{\LT@captionsetup{position=t}}%
2229 \def\captionabovefalse{\LT@captionsetup{position=b}}%
\captionlistentry for longtable:
2230 \def\captionlistentry##1{\noalign{%
2231 \caption@listentry[\LTcaption]{##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
2232 % \caption@ifhyccap{%
2233 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2234 % \def\hyper@makecurrent##1{%
2235 % \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2236 % \caption@makestart{##1}%
2237 % \let\Hy@LT@currentHlabel\@currentHlabel
2238 % \let\Hy@LT@currentHref\@currentHref
2239 % \def\hyper@makecurrent###1{%
2240 % \let\@currentHlabel\Hy@LT@currentHlabel
2241 % \let\@currentHref\Hy@LT@currentHref}}%
2242 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
2243 % \def\ContinuedFloat{\noalign{%
2244 % \gdef\caption@setContinuedFloat{%
2245 % \let\caption@resetContinuedFloat\@gobble}%
2246 % \def\caption@setoptions####1{%
2247 % \g@addto@macro\caption@setContinuedFloat{%
2248 % \caption@setoptions{####1}}}%
2249 % \let\@caption@LTcaption
2250 % \caption@ORI@ContinuedFloat}}%
2251 % }{%
2252 % \def\ContinuedFloat{\noalign{%
2253 % \caption@Error{%
2254 % \noexpand\ContinuedFloat inside longtables\MessageBreak
2255 % is only available with 'hyccap=true'}}}%
2256 % }%
2257 % \global\let\caption@setContinuedFloat\@empty
2258 % \def\ContinuedFloat{\noalign{%
2259 % \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2260 \caption@ORI@LT@array}}%

```

\LT@caption The original implementation:

```

\def\LT@caption#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}:

```

2261 \def\LT@c@ption#1[#2]#3{%
2262   \LT@makecaption#1{\csname fnum@\LTcapttype\endcsname}{#3}%
2263   \def\@currentlabelname{#2}%
2264   \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:

```

2265 \renewcommand\LT@makecaption[3]{%
2266   \caption@LT@make{%

```

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

```

2267   \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.)

```

2268   \caption@settype*\LTcapttype
2269   \ifdim\LTcapwidth=4in \else
2270     \setcaptionwidth\LTcapwidth
2271   \fi
2272   \caption@setoptions{longtable}%
2273 %   \caption@setContinuedFloat
2274   \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!

```

2275   \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
2276   \caption@resetContinuedFloat\LTcapttype
2277   \caption@begin\LTcapttype
2278   \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...)

```
2279          \vskip-\ht\strutbox
```

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

```
2280          \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2281          \caption@@make{#2}{#3}\endgraf
2282          \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2283          \caption@end}}%

2284 }{ }
```

2.19.8 The picinpar package

```
2285 \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.

```
2286 \long\def\figwindow[#1,#2,#3,#4] {%
2287   \caption@window{figure}%
2288   \caption@setoptions{figwindow}%
2289   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%

2290 \long\def\tabwindow[#1,#2,#3,#4] {%
2291   \caption@window{table}%
2292   \caption@setoptions{tabwindow}%
2293   \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.

```
2294 \newcommand*\caption@window{%
2295   \let\caption@boxrestore\@parboxrestore
2296   \let\@makecaption\caption@@make
2297   \caption@setautoposition b%
2298   \caption@settype}%

```

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

```
2299 \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`.

```
2300   \ifdim\picwd=\z@
2301     \let\caption@makecurrent\@gobbletwo
2302     \let\caption@@start\relax
2303     \caption@prepareslc
2304   \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.

```

2305 \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2306 \if\@tempa*
2307 \let\@tempa\@firstofone
2308 \else\if\@tempa[%
2309 \let\@tempa\@firstofone
2310 \else
2311 \let\@tempa\@empty
2312 \fi\fi
2313 \expandafter\caption\@tempa{#1}}%
2314 {}{}

```

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 \caption, 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.

```

2315 \newcommand*\piccaptiontype[1]{\def\@piccaptiontype{#1}}

```

```

2316 \caption@ifpackageloaded{picins}{%

```

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

```

2317 \ifundefined{\@piccaptiontype}{%
2318 \caption@iftype{%
2319 \let\@piccaptiontype\@captiontype
2320 }{%
2321 \def\@piccaptiontype{figure}%
2322 }%
2323 }{}%
2324 \let\@captiontype\@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.

```

2325 \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.

```

2326 \let\caption@ORI@make@piccaption\make@piccaption
2327 \def\make@piccaption{%
2328   \let\caption@ORI\caption
2329   \long\def\caption[##1]##2{%
2330     \caption@freezeHref % will be defrosted in \ivparpic
2331     \caption@settype\@piccapttype
2332     \ifnum\c@piccaptionpos>2\relax
2333       \caption@clearmargin
2334     \else
2335       \captionwidth\z@ % do not use "width=" setting
2336     \fi
2337     \caption@setoptions{parpic}%
2338     \let\caption@boxrestore\@parboxrestore
2339     \caption@setautoposition b%
2340     \expandafter\expandafter\expandafter\caption@ORI
2341       \expandafter\expandafter\expandafter[%
2342         \expandafter\expandafter\expandafter{%
2343           \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
2344   \caption@ORI@make@piccaption
2345   \let\caption\caption@ORI}%

```

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

```

2346 \let\caption@ORI@ivparpic\ivparpic
2347 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
2348   \let\caption@ORI@noindent\noindent
2349   \def\noindent{%
2350     \caption@defrostHref
2351     \let\noindent\caption@ORI@noindent
2352     \noindent}%

```

```

2353 \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2354 \let\noindent\caption@ORI@noindent}%

2355 }{%
2356 \let\piccaptiontype\@undefined
2357 }

```

2.19.10 The rotating package

```

2358 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%

\rotcaption Make \rotcaption* work.
2359 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
2360 % \let\@rotcaption\@undefined

\rotcaptionof Make \rotcaptionof(*) work.
2361 \def\rotcaptionof{%
2362 \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.)

```

2363 \long\def\@makerotcaption#1#2{%
2364 \ifdim\captionwidth=\z@
2365 \setcaptionwidth{.8\textheight}%
2366 \caption@slc{#1}{#2}{.8\vsizel}{%
2367 \let\caption@makerot\caption@@make
2368 \caption@clearmargin
2369 % \long\def\captionbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2370 % (not needed because \rotatebox uses an \hbox anyway)
2371 \let\captionbox\@secondoftwo}%
2372 \caption@set@bool\caption@ifslc0% been there, done that
2373 \fi
2374 \rotatebox{90}{\caption@makerot{#1}{#2}}%
2375 \nobreak\hspace{12pt}}%

2376 \newcommand\caption@makerot[2]{%
2377 \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%

2378 \ForFloatingEnvironments{%
2379 \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
2380 \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%

```


2381 }{ }

2.19.11 The sidecap package

2382 \caption@ifpackageloaded{sidecap}[1999/05/11 v1.4d]{%
2383 \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.)

2384 \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.

2385 \let\caption@ORI@SC@zfloat\SC@zfloat
2386 \def\SC@zfloat#1#2#3[#4]{%
2387 \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.

2388 \caption@settype*{#2}%
2389 \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.

2390 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
2391 \def\endSC@FLOAT{%

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

2392 \let\caption@ORI@settype\caption@settype
2393 \def\caption@settype##1{% will be done in \@xfloat
2394 \caption@ORI@settype*{##1}% do not change \@currentlabel
2395 \caption@setSC@justify
2396 \caption@setoptions{SCfloat}%
2397 \caption@setoptions{SC@\@captype}%
2398 \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

2399 \def\caption@setSC@justify{%
2400 \caption@clearmargin
2401 \@ifundefined{SC@justify}{}{%
2402 \ifx\SC@justify\@empty \else
2403 \let\caption@hj\SC@justify
2404 \let\SC@justify\@empty
2405 \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!

```

2406 \let\caption\SC@orig@caption
2407 \def\SC@orig@caption[#1]##2{\caption@defrost}%

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

2408 \caption@setSC@justify % for compatibility mode
2409 \caption@ORI@endSC@FLOAT}%

2410 \ForFloatingEnvironments{%
2411 \newenvironment{SC#1}%
2412 {\SC@float[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@float}%
2413 \newenvironment{SC#1*}%
2414 {\SC@dblfloat[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@dblfloat}}%
2415 }{}

```

2.19.12 The subfigure package

```

2416 \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.

2417 \def\sf@ifpositiontop{%
2418 \ifx\@capttype\@undefined
2419 \expandafter\@gobbletwo
2420 \else\ifx\@capttype\relax
2421 \expandafter\expandafter\expandafter\@gobbletwo
2422 \else
2423 \expandafter\expandafter\expandafter\sf@if@position@top
2424 \fi\fi}

2425 \def\sf@if@position@top{%
2426 \@ifundefined{if\@capttype topcap}%
2427 {\@gobbletwo}%
2428 {\@nameuse{if\@capttype topcap}%
2429 \expandafter\@firstoftwo
2430 \else
2431 \expandafter\@secondoftwo
2432 \fi}}

2433 }{}

```

2.19.13 The supertabular package

```

2434 \caption@ifPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.

2435 \let\caption@ORI@tablecaption\tablecaption
2436 \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}

```

```

2437 \long\def\ST@caption#1[#2]#3{\par%
2438 \caption@settype*{#1}%
2439 \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.

```

2440 \def\caption@fixposition{%
2441 \caption@setposition{\if@topcaption t\else b\fi}}%
2442 \caption@beginex{#1}{#2}{#3}%
2443 \caption@boxrestore
2444 \caption@normalsize
2445 \@makecaption{\csname fnum#1\endcsname}{\ignorespaces #3}\par
2446 \caption@end}%
2447 }{%
2448 \let\caption@setSTposition\@undefined
2449 }

```

2.19.14 The threeparttable package

```

2450 \caption@IfPackageLoaded{threeparttable}[2003/06/13 v3.0]{%

```

\threeparttable Unfortunately \@capytype is not set when \TPT@common will be used, so we have to redefine \threeparttable and \measuredfigure instead.

```

2451 \let\caption@ORI@threeparttable\threeparttable
2452 \renewcommand*\threeparttable{%
2453 \caption@settype{table}%
2454 \caption@setposition a% ?
2455 \caption@clearmargin
2456 \caption@setoptions{threeparttable}%
2457 \caption@ORI@threeparttable}%

```

\measuredfigure Same here...

```

2458 \let\caption@ORI@measuredfigure\measuredfigure
2459 \renewcommand*\measuredfigure{%
2460 \caption@settype{figure}%
2461 \caption@setposition a% ?
2462 \caption@clearmargin
2463 \caption@setoptions{measuredfigure}%
2464 \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}

2465 \def\TPT@caption#1[#2]#3{%
2466 \gdef\TPT@docapt{%
2467 \global\let\TPT@docapt\@undefined
2468 \caption@setautoposition\caption@TPT@position
2469 \TPT@LA@caption{#1}[\{#2\}]{#3}}%
2470 \ifx\TPT@hsize\@empty
2471 \let\label\TPT@gatherlabel % Bug: does not work for measured figures
2472 \gdef\caption@TPT@position{t}%
2473 \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2474 \else
2475 \def\caption@TPT@position{b}%
2476 \TPT@docapt
2477 \fi
2478 \ignorespaces}%
2479 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2480 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%
2481 {}

```

2.19.15 The wrapfig package

```

2482 \caption@ifPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)
\float@ifstyle \float@ifstyle{<type>}{<if-clause>}{<else-clause>}
(see float package support for details)
2483 \providecommand*\float@ifstyle[1]{%
2484 \expandafter\ifx\csname fst@#1\endcsname\relax
2485 \expandafter\@secondoftwo
2486 \else
2487 \expandafter\@firstoftwo
2488 \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.

```

2489 \newcommand*\caption@restylewrapfloat[1]{%
2490 \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2491 \csname wrap#1\endcsname
2492 \caption@Error{%
2493 For a successful cooperation of the 'wrapfig' package\MessageBreak
2494 with the 'float' package you should load the 'wrapfig'\MessageBreak
2495 package *after* the 'float' package}%
2496 \else
2497 \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2498 \csname wrap#1\endcsname

```

```

2499     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
2500     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2501         \csname wrap#1\endcsname
2502     \fi}%

```

\caption@wrapfloat

```

2503 \newcommand*\caption@wrapfloat[1]{%
2504     \caption@settype*{#1}%
2505     \float@ifstyle{#1}{%
2506         \ifx\WF@floatstyhook\@undefined
2507             \caption@Error{%
2508                 For a successful cooperation of the 'wrapfig' package\MessageBreak
2509                 with the 'float' package you should use at least\MessageBreak
2510                 'wrapfig' version 3.6}%
2511             \else
2512                 \float@dostyle{#1}%
2513             \fi}{}%
2514     \caption@clearmargin
2515     \caption@setoptions{wrapfloat}%
2516     \caption@setoptions{wrap#1}%
2517     \@nameuse{caption@ORI@wrap#1}}%

```

Now we redefine the wrapfig environments we know about.

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

```

2518 \caption@restylewrapfloat{figure}%
2519 \caption@restylewrapfloat{table}%
2520 \ForFloatingEnvironments{%
2521     \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
2522     \caption@restylewrapfloat{#1}}%
2523 \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.

```

2524     \@ifundefined{float@restyle}{}{%
2525         \toks@=\expandafter{\float@restyle{#1}}% (env may or may not be defined)
2526         \caption@restylewrapfloat{#1}}%
2527     \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2528     \@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.

```

2529     \let\caption@ORI@wrapfloat\wrapfloat
2530     \def\wrapfloat#1{%
2531         \float@ifstyle{#1}{%
2532             \caption@Error{%
2533                 For a successful cooperation of the 'wrapfig' package\MessageBreak
2534                 with the 'float' package you should load the 'wrapfig'\MessageBreak

```

```

2535         package *right after* the 'float' package}}{}%
2536         \caption@ORI@wrapfloat{#1}}%

2537     \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:

2538 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2539     \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2540     \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
2541     \caption@start
2542     \ifdim\hsize>\z@ \@parboxrestore \else
2543     \setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
2544     \ignorespaces \fi}%

2545 }{}

```

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