

The Implementation of the caption package*

Axel Sommerfeldt
caption@sommerfee.de

2009/10/09

Abstract

The caption package consists of two parts – the kernel (`caption3.sty`) and the main package (`caption.sty`).

The kernel provides all the user commands and internal macros which are necessary for typesetting captions and setting parameters regarding these. While the standard L^AT_EX document classes provide an internal command called `\@makecaption` and no options to control its behavior (except the vertical skips above and below the caption itself), we provide similar commands called `\caption@make` and `\caption@@make`, but with a lot of options which can be selected with `\captionsetup`. Loading the kernel part do not change the output of a L^AT_EX document – it just provides functionality which can be used by L^AT_EX 2_ε packages which typesets captions, for example the caption and subfig packages.

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

User manuals

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

[caption-eng.pdf](#) The English documentation
[caption-rus.pdf](#) The Russian documentation¹
[caption-deu.pdf](#) The German documentation

*This package has version number v3.1k, last revised 2009/10/09.

¹Thanks a lot to Olga Lapko for this translation

Contents

1	Kernel	4
1.1	Identification	4
1.2	Generic helpers	4
1.3	Errors	7
1.4	Using the keyval package	8
1.5	Margin resp. width	12
1.6	Indentions	14
1.7	Styles	15
1.8	Formats	16
1.9	Label formats	18
1.10	Label separators	18
1.11	Text formats	19
1.12	Fonts	20
1.13	Justifications	21
1.13.1	The ragged2e package	22
1.14	Vertical spaces before and after captions	23
1.15	Positioning	24
1.16	Hooks	25
1.17	Lists	25
1.18	Debug option	26
1.19	Document classes & Babel support	26
1.19.1	The standard L ^A T _E X classes	26
1.19.2	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	27
1.19.3	The beamer class	28
1.19.4	The KOMA-Script classes	29
1.19.5	The NTG Dutch classes	29
1.19.6	The thesis class	30
1.19.7	The frenchb Babel option	30
1.19.8	The frenchle/pro package	31
1.20	Execution of options	31
1.21	Making an ‘List of’ entry	32
1.22	Typesetting the caption	32
1.23	Types & sub-types	36
1.24	subfig package adaption	46
2	Main package	48
2.1	Identification	48
2.2	Loading the kernel	48
2.3	Check against incompatible document classes	48

2.4	Check against incompatible packages	48
2.5	Declaration of options	48
2.5.1	Options for figure and table	48
2.5.2	Miscellaneous options	49
2.5.3	caption v1.x compatibility options	50
2.5.4	caption2 v2.x compatibility options	50
2.5.5	Obsolete caption v3.0 options	51
2.5.6	ftpage package support options	51
2.5.7	hyperref package support options	51
2.6	$\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes support	51
2.7	KOMA-Script document classes support	51
2.8	Processing of options	53
2.9	<code>\captionof</code> and <code>\captionlistentry</code>	53
2.10	<code>\ContinuedFloat</code>	56
2.11	Internal helpers	57
2.12	<code>\caption</code> , <code>\@caption</code> , and <code>\@makecaption</code>	60
2.13	Support for sub-captions	67
2.14	Document class & Babel package support	68
2.14.1	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	68
2.14.2	The beamer class	69
2.14.3	The KOMA-Script classes	69
2.14.4	The frenchb Babel option	69
2.14.5	The frenchle/pro package	69
2.15	Package support	70
2.15.1	The float package	72
2.15.2	The floatflt package	75
2.15.3	The ftpage package	76
2.15.4	The hyperref package	78
2.15.5	The hypcap package	81
2.15.6	The listings package	82
2.15.7	The longtable package	83
2.15.8	The picinpar package	86
2.15.9	The picins package	87
2.15.10	The rotating package	88
2.15.11	The sidecap package	89
2.15.12	The subfigure package	91
2.15.13	The supertabular and xtab packages	91
2.15.14	The threeparttable package	93
2.15.15	The wrapfig package	94

1 Kernel

1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2009/10/09 v3.1k 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 \LaTeX 2 ϵ 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` `\bothIfSecond` tests if the first argument is not empty, `\bothIfSecond` tests if the second argument is not empty. If yes both arguments get typeset, otherwise none of them.

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

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

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

would expand to yes.

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

```

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

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

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

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

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

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

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

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

```

```

62 \newcommand*\caption@setbool[1]{%
63   \expandafter\caption@set@bool\csname caption@if#1\endcsname}
64 \newcommand*\caption@set@bool[2]{%
65   \caption@ifinlist{#2}{1,true,yes,on}{%
66     \let#1\@firstoftwo
67   }\caption@ifinlist{#2}{0,false,no,off}{%
68     \let#1\@secondoftwo
69   }{%
70     \caption@Error{Undefined boolean value `#2'}%
71   }}
72 \newcommand*\caption@ifbool[1]{\@nameuse{caption@if#1}}
73 \newcommand*\caption@undefbool[1]{\@nameundef{caption@if#1}}
\caption@teststar \caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
\caption@teststar@{<cmd>}{<star arg>}{<non-star arg>}
74 \newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}
75 \newcommand*\caption@teststar@[3]{%
76   \@ifstar{#1{#2}}{\caption@ifatletter{#1{#2}}{#1{#3}}}}
77 \AtBeginDocument{\let\caption@teststar@\caption@teststar}
78 \newcommand*\caption@ifatletter{%
79   \ifnum\the\catcode`\@=11
80     \expandafter\@firstoftwo
81   \else
82     \expandafter\@secondoftwo
83   \fi}
84 \AtBeginDocument{\let\caption@ifatletter\@secondoftwo}
\caption@withoptargs \caption@withoptargs{<cmd>}
85 \newcommand*\caption@withoptargs[1]{%
86   \@ifstar
87   {\def\caption@tempa{*}\caption@@withoptargs#1}%
88   {\def\caption@tempa{\caption@@withoptargs#1}}
89 \def\caption@@withoptargs#1{%
90   \@ifnextchar[%
91     {\caption@@@withoptargs#1}%
92     {\caption@@@@withoptargs#1}}
93 \def\caption@@@@withoptargs#1[#2]{%
94   \l@addto@macro\caption@tempa{[#2]}%
95   \caption@@withoptargs#1}
96 \def\caption@@@@@withoptargs#1{%
97   \expandafter#1\expandafter{\caption@tempa}}
\caption@CheckCommand \caption@CheckCommand{<macro>}{<definition of macro>}
\caption@IfCheckCommand checks if a command already exists, with the same definition. It can be used more-than-
once to check if one of multiple definitions will finally match. (It redefines itself later on
to \@gobbletwo if the two commands match fine, making further checks harmless.)
\caption@IfCheckCommand{<true>}{<false>}
will execute the <true> code if one match was finally given, the <false> code otherwise.
(It simply checks if \caption@CheckCommand is \@gobbletwo and restores the
starting definition of \caption@CheckCommand.)

```

```

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

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

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

```

119 \let\caption@begindocumenthook\@empty
120 \let\caption@@begindocumenthook\@empty

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

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

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

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

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

```

1.3 Errors

```

\caption@Warning \caption@Error{message}
\caption@WarningNoLine 136 \newcommand*\caption@Warning[1]{%
  \caption@Error 137   \caption@WarningNoLine{#1\on@line}}
\caption@KV@err 138 \newcommand*\caption@WarningNoLine[1]{%

```

```

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

```

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

```

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

```

1.4 Using the keyval package

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

```

149 \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.

```

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

```

```

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

```

Analogous to `\@onlypreamble` from `LATEX 2ε`.

```

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

```

```

\DeclareCaptionOption \DeclareCaptionOption{<option>}[<default value>]{<code>}
\DeclareCaptionOption*{<option>}[<default value>]{<code>}

```

We declare our options using these commands (instead of using `\DeclareOption` offered by `LATEX 2ε`), 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.

```

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

```

```

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

```

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


```

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

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

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

```

`\ifcaptionsetup@star` If the starred form of `\captionsetup` is used, this will be set to `true`. (It will be reset to `false` at the end of `\caption@setkeys`.)

```
181 \newif\ifcaptionsetup@star
```

```

\captionsetup \captionsetup[<type>]{<keyval-list of options>}
\captionsetup* [ <type> ] { <keyval-list of options> }

```

If the optional argument ‘*type*’ is specified, we simply save or append the option list, otherwise we ‘execute’ it with `\setkeys`. (The non-starred variant issues a warning if *<keyval-list of options>* is not used later on.)

Note: The starred variant will be used inside packages automatically.

```

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

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

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

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

```

```

\clearcaptionsetup \clearcaptionsetup[<option>]{<type>}
\clearcaptionsetup* [ <option> ] { <type> }

```

This removes the saved option list associated with *<type>*. If *<option>* is given, only this option will be removed from the list. (The starred variant does not issue warnings.)

Note: The starred variant will be used inside packages automatically.

```

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

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

198 \def\@clearcaptionsetup[#1]#2{%
199   \@bsphack

```

```

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

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

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

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

```

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

243 \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.

```

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

251 \newcommand*\caption@xsetup[1]{\expandafter\caption@setup\expandafter{#1}}

```

`\caption@addtooptlist`
`\caption@removefromoptlist`

`\caption@addtooptlist{⟨type⟩}`
`\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”).

```

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

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

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

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

```

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

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

273   \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
274   \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
275   \expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
276   \ifx\KV@err\relax
277     \def\KV@err##1{\PackageError{#1}{##1}{%
278       See the #1 package documentation for explanation.}}%
279   \fi

```

```

280 \let\KV@errx\KV@err
281 \edef\caption@keydepth{\caption@keydepth i}%
282 \caption@Debug{\protect\setkeys{#2}{#3}}%
283 \setkeys{#2}{#3}%
284 \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
285 \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
286 \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
287 \ifx\caption@keydepth\@empty \captionsetup@starfalse \fi
288 \@esphack}
289 \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.

```

```

290 \newcommand*\caption@ExecuteOptions[2]{%
291 \@expandtwoargs\caption@setkeys{#1}{#2}}%
292 \@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.)

```

```

293 \newcommand*\caption@ProcessOptions{%
294 \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
295 \@onlypreamble\caption@ProcessOptions

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

```

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

```

317 \newdimen\captionmargin
318 \newdimen\captionmargin@
319 \newdimen\captionwidth

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

325 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
326 \DeclareCaptionOption{maxmargin}{\caption@setoptcmd\caption@maxmargin{#1}}

```

`\setcaptionmargin` `\setcaptionmargin{<amount>}`
`\setcaptionmargin*{<amount>}`

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

```

327 \newcommand*\setcaptionmargin{%
328   \caption@teststar\caption@setmargin\@gobble\@firstofone}

329 \newcommand*\caption@setmargin[2]{%
330   #1{\captionwidth\z@}%
331   \caption@@setmargin#2,#2,\@nil}

332 \def\caption@@setmargin#1,#2,#3\@nil{%
333   \setlength\captionmargin@{#2}%
334   \setlength\captionmargin{#1}%
335   \addtolength\captionmargin@{-\captionmargin}}

```

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

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

```

336 \newcommand*\setcaptionwidth{%
337   \captionmargin\z@
338   \captionmargin@\z@
339   \setlength\captionwidth}

```

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

```

340 \newcommand*\caption@thecounter{0}

341 \newcommand*\caption@stepcounter{%
342   \@tempcnta\caption@thecounter
343   \advance\@tempcnta\@ne
344   \xdef\caption@thecounter{\the\@tempcnta}}

```

`\caption@newlabel` This command is a modified version of `\newlabel` from L^AT_EX2e. It will be written to the `.aux` file to pass label information from one run to another. (We use it inside `\caption@ifoddpag` and `\caption@ragged`.)

```

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

```

`\caption@thepage` This command is a modified version of `\thepage` from L^AT_EX2e. It will be used inside `\caption@ifoddpage` only.

```
346 \newcommand*\caption@thepage{\the\c@page}
```

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

```
347 \newcommand*\caption@label[1]{%
348   \caption@@label
349   \protected@write\@auxout{\let\caption@thepage\relax}%
350     {\string\caption@newlabel{#1}{\caption@thepage}}
351 \newcommand*\caption@@label{%
352   \global\let\caption@@label\relax
353   \protected@write\@auxout{}%
354     {\string\providecommand*\string\caption@newlabel[2]{}}}
```

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

```
355 \newcommand*\caption@pageref[1]{%
356   \expandafter\ifx\csname caption@r@#1\endcsname\relax
357     \G@refundefinedtrue % => 'There are undefined references.'
358     \@latex@warning{Reference '#1' on page \thepage \space undefined}%
359   \else
360     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
361   \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[24] and `\ifodd\getpagerefnumber{...}`.
Note: This macro re-defines itself so the `.aux` file will only be used once per group.

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

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

```
373 \newcommand*\caption@setoptcmd[2]{%
374   \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`

```
375 \newdimen\caption@indent
```


`\caption@hangindent`

```
376 \newdimen\caption@parindent
377 \newdimen\caption@hangindent
```

```

378 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
379     \setlength\caption@indent{#1}}
380 \DeclareCaptionOption{indentation}[\leftmargini]{%
381     \setlength\caption@indent{#1}}
382 \DeclareCaptionOption{parindent}{%
383     \setlength\caption@parindent{#1}}
384 \DeclareCaptionOption{hangindent}{%
385     \setlength\caption@hangindent{#1}}
386 \DeclareCaptionOption{parskip}{%
387     \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

```

388 \@ifundefined{scr@caption}{}{%
389     \let\caption@KV@parindent\KV@caption@parindent
390     \DeclareCaptionOption{parindent}[]{%
391         \ifx,#1,%
392             \caption@Debug{Option 'parindent' ignored}%
393         \else
394             \caption@KV@parindent{#1}%
395         \fi}%
396     \let\caption@KV@parskip\KV@caption@parskip
397     \DeclareCaptionOption{parskip}[]{%
398         \ifx,#1,%
399             \caption@Debug{Option 'parskip' ignored}%
400         \else
401             \caption@KV@parskip{#1}%
402         \fi}%
403 }

```

1.7 Styles

```

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

```

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

```
\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.

```
415 \newcommand*\caption@setstyle{%
416   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

417 \newcommand*\caption@@setstyle[2]{%
418   \@ifundefined{caption@sty@#2}%
419     {#1{\caption@Error{Undefined style '#2'}}}%
420     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
421     \ifx\caption@setstyle@flag\@undefined
422       \let\caption@setstyle@flag\relax
423       \caption@resetstyle
424       \caption@xsetup\caption@sty
425       \let\caption@setstyle@flag\@undefined
426     \else
427       \caption@xsetup\caption@sty
428     \fi
429     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
430     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
431     \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!*

```
432 \newcommand*\caption@resetstyle{%
433   \caption@setup{%
434     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
435     justification=justified,font=,size=,labelfont=,textfont=,%
436     margin=0pt,minmargin=0,maxmargin=0,%
437     indent=0pt,parindent=0pt,hangindent=0pt,%
438     slc,rule,strut}%
439   \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.

```
440 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
441 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{}
442   format=default,labelsep=default,textformat=default,%
443   justification=default,font=default,labelfont=default,textfont=default}
```

1.8 Formats

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

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


```

444 \newcommand*\DeclareCaptionFormat{%
445   \caption@teststar\caption@declareformat\@gobble\@firstofone}
446 \@onlypreamble\DeclareCaptionFormat

447 \newcommand*\caption@declareformat[2]{%
448   \@dblarg{\caption@@declareformat#1{#2}}
449 \@onlypreamble\caption@declareformat

450 \long\def\caption@@declareformat#1#2[#3]#4{%
451   \global\expandafter\let\csname caption@ifh#2\endcsname#1%
452   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
453   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
454 \@onlypreamble\caption@@declareformat

455 \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`).

```

456 \newcommand*\caption@setformat[1]{%
457   \@ifundefined{caption@fmt@#1}%
458     {\caption@Error{Undefined format '#1'}}%
459     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh#1\endcsname
460      \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
461      \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

`\DeclareCaptionDefaultFormat`

```

462 \newcommand*\DeclareCaptionDefaultFormat[1]{%
463   \expandafter\def\expandafter\caption@fmt@default\expandafter
464     {\csname caption@fmt@#1\endcsname}%
465   \expandafter\def\expandafter\caption@slfmt@default\expandafter
466     {\csname caption@slfmt@#1\endcsname}%
467   \expandafter\def\expandafter\caption@ifh@default\expandafter
468     {\csname caption@ifh@#1\endcsname}}
469 \@onlypreamble\DeclareCaptionDefaultFormat

```

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

```

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

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

```

‘default’ usually maps to ‘plain’.

```

480 \DeclareCaptionDefaultFormat{plain}

```

1.9 Label formats

```
DeclareCaptionLabelFormat \DeclareCaptionLabelFormat{<name>}{<code with #1 and #2>}
481 \newcommand*\DeclareCaptionLabelFormat[2]{%
482   \global\@namedef{caption@lfmt@#1}##1##2{#2}}
483 \@onlypreamble\DeclareCaptionLabelFormat

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

\caption@setlabelformat \caption@setlabelformat{<name>}
Selecting a caption label format simply means saving the code (in \caption@lfmt).
485 \newcommand*\caption@setlabelformat[1]{%
486   \ifundefined{caption@lfmt@#1}%
487     {\caption@Error{Undefined label format `#1'}}%
488     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}
```

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

```
489 \DeclareCaptionLabelFormat{empty}{}
490 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
491 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace} (#2)}
492 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace}#2}
```

‘default’ usually maps to ‘simple’.

```
493 \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.
494 \newcommand\DeclareCaptionLabelSeparator{%
495   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
496 \@onlypreamble\DeclareCaptionLabelSeparator

497 \newcommand\caption@declarelabelseparator[3]{%
498   \global\expandafter\let\csname caption@iflf@#2\endcsname#1%
499   \global\long\@namedef{caption@lsep@#2}{#3}%
500   \caption@@declarelabelseparator{#2}#3\\@nil}
501 \@onlypreamble\caption@declarelabelseparator

502 \long\def\caption@@declarelabelseparator#1#2\\#3\@nil{%
503   \def\@tempa{#3}\ifx\@tempa\@empty \else
504     \caption@g@addto@list\caption@lsep@#1%
505     \fi}
506 \@onlypreamble\caption@@declarelabelseparator

507 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
508 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

```
caption@setlabelseparator \caption@setlabelseparator{<name>}
Selecting a caption label separator simply means saving the code (in \caption@lsep).
509 \newcommand*\caption@setlabelseparator[1]{%
510   \ifundefined{caption@lsep@#1}%
511     {\caption@Error{Undefined label separator `#1'}}%
512     {\edef\caption@lsepname{#1}%

```

```

513 \expandafter\let\expandafter\caption@iflfl\csname caption@iflfl@#1\endcsname
514 \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’.

```

515 \DeclareCaptionLabelSeparator{none}{}
516 \DeclareCaptionLabelSeparator{colon}{: }
517 \DeclareCaptionLabelSeparator{period}{. }
518 \DeclareCaptionLabelSeparator{space}{ }
519 \DeclareCaptionLabelSeparator*{quad}{\quad}
520 \DeclareCaptionLabelSeparator*{newline}{\}
521 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

```

`\caption@setdefaultlabelsep`

```

522 \newcommand*\caption@setdefaultlabelsep[1]{%
523 \ifx\caption@lsep\caption@lsep@default
524 \caption@set@default@labelsep{#1}%
525 \caption@set@labelseparator{default}%
526 \else
527 \caption@set@default@labelsep{#1}%
528 \fi}

529 \newcommand*\caption@set@default@labelsep[1]{%
530 \def\caption@lsep@default{\@nameuse{caption@lsep@#1}}%
531 \def\caption@iflfl@default{\@nameuse{caption@iflfl@#1}}}}

```

‘default’ usually maps to ‘colon’.

```

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

```

1.11 Text formats

`\DeclareCaptionTextFormat`

```

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

536 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
537 \DeclareCaptionOption{strut}[1]{\caption@set@bool\caption@ifstrut{#1}}

```

`\caption@settextformat`

```

\caption@settextformat{<name>}

```

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

```

538 \newcommand*\caption@settextformat[1]{%
539 \@ifundefined{caption@tfmt@#1}%
540 {\caption@Error{Undefined text format `#1'}}%
541 {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

542 \DeclareCaptionTextFormat{simple}{#1}
543 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

1.12 Fonts

```

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

DeclareCaptionDefaultFont \DeclareCaptionDefaultFont {<name>} {<code>}
548 \newcommand*\DeclareCaptionDefaultFont [2] {%
549   \global\@namedef{caption#1@default} {#2}}
550 \@onlypreamble\DeclareCaptionDefaultFont

551 \DeclareCaptionOption{font} {\caption@setfont {font} {#1}}
552 \DeclareCaptionOption{font+} {\caption@addtofont {font} {#1}}
553 \DeclareCaptionDefaultFont {font} {}

554 \DeclareCaptionOption{labelfont} {\caption@setfont {labelfont} {#1}}
555 \DeclareCaptionOption{labelfont+} {\caption@addtofont {labelfont} {#1}}
556 \DeclareCaptionDefaultFont {labelfont} {}

557 \DeclareCaptionOption{textfont} {\caption@setfont {textfont} {#1}}
558 \DeclareCaptionOption{textfont+} {\caption@addtofont {textfont} {#1}}
559 \DeclareCaptionDefaultFont {textfont} {}

\caption@setfont \caption@setfont {<name>} {<keyval-list of names>}
Selecting a caption font means saving all the code snippets in \caption<name>.
560 \newcommand*\caption@setfont [1] {%
561   \expandafter\let\csname caption#1\endcsname \@empty
562   \caption@addtofont {#1}}

\caption@addtofont \caption@addtofont {<name>} {<keyval-list of names>}
Like \caption@setfont, but adds the code snippets to \caption<name>.
Because we use \setkeys recursive here we need to do this inside an extra group.
563 \newcommand*\caption@addtofont [2] {%
564   \begingroup
565     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
566     \define@key{caption@fnt} {default} [] {%
567       \l@addto@macro\caption@fnt {\csname caption#1@default\endcsname}}%
568     \caption@setkeys[caption] {caption@fnt} {#2}%
569     \global\let\caption@tempa\caption@fnt
570   \endgroup
571   \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.
572 \newcommand*\caption@font {%
573   \caption@teststar\caption@@font \@firstofone
574     {\caption@setkeys[caption] {caption@fnt}}
575 \newcommand*\caption@@font [2] {%
576   \begingroup
577   \def\caption@fnt {\endgroup}%
578   #1 {#2}%
579   \caption@fnt}

```

These are the pre-defined font code snippets.

```
580 \DeclareCaptionFont{normalcolor}{\normalcolor}
581 \DeclareCaptionFont{color}{\color{#1}}

582 \DeclareCaptionFont{normalfont}{\normalfont}
583 \DeclareCaptionFont{up}{\upshape}
584 \DeclareCaptionFont{it}{\itshape}
585 \DeclareCaptionFont{sl}{\slshape}
586 \DeclareCaptionFont{sc}{\scshape}
587 \DeclareCaptionFont{md}{\mdseries}
588 \DeclareCaptionFont{bf}{\bfseries}
589 \DeclareCaptionFont{rm}{\rmfamily}
590 \DeclareCaptionFont{sf}{\sffamily}
591 \DeclareCaptionFont{tt}{\ttfamily}

592 \DeclareCaptionFont{scriptsize}{\scriptsize}
593 \DeclareCaptionFont{footnotesize}{\footnotesize}
594 \DeclareCaptionFont{small}{\small}
595 \DeclareCaptionFont{normalsize}{\normalsize}
596 \DeclareCaptionFont{large}{\large}
597 \DeclareCaptionFont{Large}{\Large}

598 \DeclareCaptionFont{singlespacing}{%
599   \@ifundefined{setspace@singlespace}{}{%
600     \setstretch\setspace@singlespace}}% normally 1
601 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
602 \DeclareCaptionFont{doublespacing}{\doublespacing}
603 \DeclareCaptionFont{stretch}{\setstretch{#1}}

604 %\DeclareCaptionFont{normal}{%
605 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
606 \DeclareCaptionFont{normal}{%
607   \caption@font*{%
608     \KV@caption@fnt@normalcolor\@unused
609     \KV@caption@fnt@normalfont\@unused
610     \KV@caption@fnt@normalsize\@unused
611     \KV@caption@fnt@singlespacing\@unused}}
```

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

```
612 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
613 \DeclareCaptionDefaultFont{size}{}
```

1.13 Justifications

```
\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
614 \newcommand*\DeclareCaptionJustification[2]{%
615   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
616   \DeclareCaptionFont{#1}{#2}}
617 \@onlypreamble\DeclareCaptionJustification

\DeclareCaptionDefaultJustification \DeclareCaptionDefaultJustification{<code>}
618 \newcommand*\DeclareCaptionDefaultJustification[1]{%
619   \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
620   \DeclareCaptionDefaultFont{@hj}{#1}}
621 \@onlypreamble\DeclareCaptionDefaultJustification
```

```
622 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
623 \DeclareCaptionDefaultJustification{}
```

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

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

```
624 \newcommand*\caption@setjustification{\caption@setfont{@hj}}
```

These are the pre-defined justification code snippets.

```
625 \DeclareCaptionJustification{justified}{}
626 \DeclareCaptionJustification{centering}{\centering}
627 \DeclareCaptionJustification{centerfirst}{\centerfirst}
628 \DeclareCaptionJustification{centerlast}{\centerlast}
629 \DeclareCaptionJustification{raggedleft}{\raggedleft}
630 \DeclareCaptionJustification{raggedright}{\raggedright}
```

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

```
631 \providecommand\centerfirst{%
632   \let\\\@centercr
633   \edef\caption@normaladjust{%
634     \leftskip\the\leftskip
635     \rightskip\the\rightskip
636     \parfillskip\the\parfillskip\relax}%
637   \leftskip\z@\@plus -1fil%
638   \rightskip\z@\@plus 1fil%
639   \parfillskip\z@skip
640   \noindent\hskip\z@\@plus 2fil%
641   \@setpar{\@@par\@restorepar\caption@normaladjust}}
```

`\centerlast` This is based on code from Anne Brüggemann-Klein[23]

```
642 \providecommand\centerlast{%
643   \let\\\@centercr
644   \leftskip\z@\@plus 1fil%
645   \rightskip\z@\@plus -1fil%
646   \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.

```
647 \DeclareCaptionJustification{Centering}{%
648   \caption@ragged\Centering\centering}
649 \DeclareCaptionJustification{RaggedLeft}{%
650   \caption@ragged\RaggedLeft\raggedleft}
651 \DeclareCaptionJustification{RaggedRight}{%
652   \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.

```

653 \newcommand*\caption@ragged{%
654   \caption@Debug{We need ragged2e}%
655   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
656   \global\let\caption@ragged\caption@@ragged
657   \caption@ragged}

658 \caption@AtBeginDocument{%
659   \ifundefined{caption@r@ragged2e}{%
660     \newcommand*\caption@@ragged{%
661       \caption@Warning{%
662         `ragged2e' support has been changed.\MessageBreak
663         Rerun to get captions right}%
664       \global\let\caption@ragged\@secondoftwo % suppress further warnings
665       \caption@ragged}%
666     }{%
667       \caption@Debug{We load ragged2e}%
668       \IfFileExists{ragged2e.sty}{%
669         \RequirePackage{ragged2e}%
670         \let\caption@@ragged\@firstoftwo
671       }{%
672         \newcommand*\caption@@ragged[2]{%
673           \ifundefined{caption\string#1}{%
674             \caption@Warning{%
675               `ragged2e' package not loaded, therefore\MessageBreak
676               substituting \string#2 for \string#1\MessageBreak}%
677             \global\@namedef{caption\string#1}{}%
678             #2}%
679           }%
680         }}

```

1.14 Vertical spaces before and after captions

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

```

\belowcaptionskip
681 \ifundefined{abovecaptionskip}{%
682   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}{}
683 \ifundefined{belowcaptionskip}{%
684   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}{}

685 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
686 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
687 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

`\caption@rule` `\caption@rule`

Draws an invisible rule to adjust the “skip” setting.

```

688 \newcommand*\caption@rule{\caption@ifrule\caption@hrule\relax}
689 \newcommand*\caption@hrule{\hrule\@height\z@}

690 \DeclareCaptionOption{rule}[1]{\caption@set@bool\caption@ifrule{#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.

```
691 \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.*

```
692 \newcommand*\caption@setposition[1]{%
693   \caption@ifinlist{#1}{d,default}{%
694     \let\caption@position\caption@defaultpos
695   }{\caption@ifinlist{#1}{t,top,above}{%
696     \let\caption@position\@firstoftwo
697   }{\caption@ifinlist{#1}{b,bottom,below}{%
698     \let\caption@position\@secondoftwo
699   }{\caption@ifinlist{#1}{a,auto}{%
700     \let\caption@position\@undefined
701   }{%
702     \caption@Error{Undefined position `#1'}%
703   }}}}
```

```
\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 `LATEX` document classes, but ‘auto’ should give better results in most cases.

```
704 %\caption@setdefaultpos{a}% default = auto
705 \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.)

```
706 \newcommand*\caption@iftop{%
707   \ifx\caption@position\@undefined
708     \let\caption@position\@secondoftwo
709 %   = \caption@setposition b%
710   \fi
711   \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.

```
712 \newcommand*\caption@fixposition{%
713   \ifx\caption@position\@undefined
714     \caption@autoposition
715   \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.

```

716 \newcommand*\caption@autoposition{%
717   \ifvmode
718     \edef\caption@tempa{\the\prevdepth}%
719     \caption@Debug{\protect\prevdepth=\caption@tempa}%
720     \ifdim\prevdepth>-\p@
721       \let\caption@position\@secondoftwo
722     \else
723       \let\caption@position\@firstoftwo
724     \fi
725 % = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
726 \else
727   \caption@Debug{no \protect\prevdepth}%
728   \let\caption@position\@secondoftwo
729 % = \caption@setposition b%
730 \fi}

```

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

```

731 \newcommand*\caption@setautoposition[1]{%
732   \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`.

```

733 \newcommand*\caption@beginhook{}
734 \newcommand*\caption@endhook{}
735 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
736 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

1.17 Lists

```

737 \DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
738 \DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}

```

`\caption@setlist` `\caption@setlist{<boolean>}`

```

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

```

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

```

740 \newcommand*\DeclareCaptionListFormat[2]{%
741   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
742 \@onlypreamble\DeclareCaptionListFormat

```

```

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

```

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

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

```
744 \newcommand*\caption@setlistformat[1]{%
745   \@ifundefined{caption@lstfmt@#1}%
746     {\caption@Error{Undefined list format `#1'}}%
747     {\expandafter\let\expandafter\caption@lstfmt
748       \csname caption@lstfmt@#1\endcsname}}
```

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

```
749 \DeclareCaptionListFormat{empty}{}
750 \DeclareCaptionListFormat{simple}{#1#2}
751 \DeclareCaptionListFormat{parens}{#1(#2)}
752 \DeclareCaptionListFormat{subsimple}{#2}
753 \DeclareCaptionListFormat{subparens}{(#2)}
```

```
\caption@setdefaultlistformat
```

```
754 \newcommand*\caption@setdefaultlistformat[1]{%
755   \ifx\caption@lstfmt\caption@lstfmt@default
756     \caption@set@default@listformat{#1}%
757     \caption@setlistformat{default}%
758   \else
759     \caption@set@default@listformat{#1}%
760   \fi}
761 \newcommand*\caption@set@default@listformat[1]{%
762   \def\caption@lstfmt@default{\@nameuse{caption@lstfmt@#1}}}
```

‘default’ usually maps to ‘subsimple’.

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

1.18 Debug option

```
764 \DeclareCaptionOption{debug}[1]{%
765   \caption@set@bool\caption@ifdebug{#1}%
766   \caption@ifdebug
767     {\def\caption@Debug{\PackageInfo{caption}}}%
768     {\let\caption@Debug@gobble}}
769 \DeclareOption{debug}{\setkeys{caption}{debug}}
770 \setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

1.19.1 The standard L^AT_EX classes

```
771 \caption@CheckCommand\@makecaption{%
772   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
773   \long\def\@makecaption#1#2{%
774     \vskip\abovecaptionskip
775     \sbox\@tempboxa{#1: #2}%
776     \ifdim \wd\@tempboxa >\hsize
777       #1: #2\par
778     \else
779       \global \@minipagefalse
780       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
```

```

781 \fi
782 \vskip\belowcaptionskip}}

```

1.19.2 The \mathcal{AMS} & SMF classes

```

783 \@ifundefined{@captionheadfont}{}{%
784 \caption@CheckCommand\@makecaption{%
785 % amsart|amsproc|amsbook [2004/08/06 v2.20]
786 \long\def\@makecaption#1#2{%
787 \setbox\@tempboxa\vbox{\color@setgroup
788 \advance\hsize-2\captionindent\noindent
789 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
790 {\@cdr#2\@nil}\.\@captionfont\upshape\enspace#2}%
791 \unskip\kern-2\captionindent\par
792 \global\setbox\@ne\lastbox\color@endgroup}%
793 \ifhbox\@ne % the normal case
794 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
795 \fi
796 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
797 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
798 \else % tempboxa contained more than one line
799 \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
800 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
801 \fi
802 \ifnum\@tempcnta<64 % if the float IS a figure...
803 \addvspace\abovecaptionskip
804 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
805 \else % if the float IS NOT a figure...
806 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
807 \nobreak
808 \vskip\belowcaptionskip
809 \fi
810 \relax
811 }}

812 \caption@CheckCommand\@makecaption{%
813 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
814 \long\def\@makecaption#1#2{%
815 \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
816 \setbox\@tempboxa\vbox{\color@setgroup
817 \advance\hsize-2\captionindent\noindent
818 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
819 {\@cdr#2\@nil}\@addpunct{.}\@captionfont\upshape\enspace#2}%
820 \unskip\kern-2\captionindent\par
821 \global\setbox\@ne\lastbox\color@endgroup}%
822 \ifhbox\@ne % the normal case
823 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
824 \fi
825 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
826 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
827 \@tempdima\wd\@ne\advance\@tempdima-\captionindent
828 \wd\@ne\@tempdima
829 \else % tempboxa contained more than one line
830 \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
831 \unvbox\@tempboxa\parskip\z@skip

```

```

832         \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
833     \fi
834     \ifnum\@tempcnta<64 % if the float IS a figure...
835         \addvspace\abovecaptionskip
836         \noindent\kern\captionindent\box\@ne
837     \else % if the float IS NOT a figure...
838         \noindent\kern\captionindent\box\@ne
839         \nobreak
840         \vskip\belowcaptionskip
841     \fi
842     \relax
843 }}

844 \let\captionmargin\captionindent % set to 3pc by AMS class
845 \begingroup\edef\@tempa{\endgroup
846     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
847     {margin=\the\captionmargin
848     \@ifundefined{smf@makecaption}{}{,maxmargin=.1\linewidth}}}
849 \@tempa
850 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
851 \DeclareCaptionLabelSeparator{default}{.\enspace}
852 \DeclareCaptionDefaultFont{font}{\@captionfont}
853 \DeclareCaptionDefaultFont{labelfont}{\@captionheadfont}
854 \DeclareCaptionDefaultFont{textfont}{\@captionfont\upshape}
855 \captionsetup[figure]{position=b}
856 \captionsetup[table]{position=t}

857 }

```

1.19.3 The beamer class

```

858 \@ifclassloaded{beamer}{%
859     \caption@CheckCommand\beamer@makecaption{%
860         % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
861         \long\def\beamer@makecaption#1#2{%
862             \def\insertcaptionname{\csname#1name\endcsname}%
863             \def\insertcaptionnumber{\csname the#1\endcsname}%
864             \def\insertcaption{#2}%
865             \nobreak\vskip\abovecaptionskip\nobreak
866             \sbox\@tempboxa{\usebeamertemplate**{caption}}%
867             \ifdim \wd\@tempboxa >\hsize
868                 \usebeamertemplate**{caption}\par
869             \else
870                 \global \@minipagefalse
871                 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
872             \fi
873             \nobreak\vskip\belowcaptionskip\nobreak}}

874 \DeclareCaptionLabelFormat{default}{#1}
875 \DeclareCaptionDefaultJustification{\raggedright}
876 \DeclareCaptionDefaultFont{font}{%
877     \usebeamerfont*{caption}%
878     \usebeamercolor[fg]{caption}}
879 \DeclareCaptionDefaultFont{labelfont}{%
880     \usebeamercolor[fg]{caption name}%
881     \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.)

```

882 \defbeamertemplate{caption}{caption3}{%
883   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
884 \DeclareOption{beamer}{%
885   % \usebeamertemplate*{caption} will set font
886   \DeclareCaptionDefaultFont{font}{}%
887   \setbeamertemplate{caption}[caption3]}
888 {}

```

1.19.4 The KOMA-Script classes

```

889 \@ifundefined{scr@caption}{}{%
890   \caption@CheckCommand\@makecaption{%
891     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
892     \long\def\@makecaption#1#2{%
893       \if@captionabove
894         \vskip\belowcaptionskip
895       \else
896         \vskip\abovecaptionskip
897       \fi
898       \@@makecaption\@firstofone{#1}{#2}%
899       \if@captionabove
900         \vskip\abovecaptionskip
901       \else
902         \vskip\belowcaptionskip
903       \fi}}
904 \DeclareCaptionFormat{default}[#1#2#3\par]{%
905   \ifdofullc@p
906     \caption@ifin@list\caption@lsep@clist\caption@lsepname
907     {\caption@Error{%
908       The option ‘labelsep=\caption@lsepname’ does not work\MessageBreak
909       with \noexpand\setcaphanging (which is set by default)}}%
910     {\caption@fmt@hang{#1}{#2}{#3}}%
911   \else
912     #1#2%
913     \ifdim\cap@indent<\z@
914       \par
915       \noindent\hspace*{-\cap@indent}%
916     \else\if@capbreak
917       \par
918     \fi\fi
919     #3\par
920   \fi}
921 \DeclareCaptionLabelSeparator{default}{\captionformat}
922 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
923 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
924 }

```

1.19.5 The NTG Dutch classes

```

925 \@ifundefined{CaptionFonts}{}{%

```

```

926 \caption@CheckCommand\@makecaption{%
927   % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
928   \long\def\@makecaption#1#2{%
929     \vskip\abovecaptionskip
930     \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
931     \ifdim \wd\@tempboxa >\hsize
932       {\CaptionLabelFont#1:} \CaptionTextFont#2\par
933     \else
934       \global \@minipagefalse
935       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
936     \fi
937     \vskip\belowcaptionskip}}
938 \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
939 \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
940 }

```

1.19.6 The thesis class

```

941 \@ifclassloaded{thesis}{%
942   \caption@CheckCommand\@makecaption{%
943     % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
944     \long\def\@makecaption#1#2{%
945       \vskip\abovecaptionskip
946       \setbox\@tempboxa\hbox{\cph@font #1:} {\cpb@font #2}}%
947       \ifdim \wd\@tempboxa >\hsize
948         \@hangfrom{\cph@font #1:} {\cpb@font #2\par}%
949       \else
950         \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
951       \fi
952       \vskip\belowcaptionskip}}
953 \DeclareCaptionDefaultFormat{hang}
954 \DeclareCaptionDefaultFont{labelfont}{\cph@font}
955 \DeclareCaptionDefaultFont{textfont}{\cpb@font}
956 }{}

```

1.19.7 The french Babel option

```

957 \@ifundefined{FB@makecaption}{}{%
958   \caption@CheckCommand\@makecaption{%
959     % frenchb.1df [2005/02/06 v1.6g French support from the babel system]
960     % frenchb.1df [2007/10/05 v2.0e French support from the babel system]
961     \long\def\@makecaption#1#2{%
962       \vskip\abovecaptionskip
963       \sbox\@tempboxa{#1\CaptionSeparator #2}%
964       \ifdim \wd\@tempboxa >\hsize
965         #1\CaptionSeparator #2\par
966       \else
967         \global \@minipagefalse
968         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
969       \fi
970       \vskip\belowcaptionskip}}
971 \ifx\@makecaption\STD@makecaption
972   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}

```

```

973 \def\caption@frenchb{% supress frenchb warning
974 \let\STD@makecaption\@makecaption
975 \let\FB@makecaption\@makecaption}
976 \else
977 \ifx\@makecaption\@undefined\else
978 \PackageInfo{caption}{%
979 The definition of \protect\@makecaption\space
980 has been changed,\MessageBreak
981 frenchb will NOT customize it}%
982 \fi
983 \fi
984 }

```

1.19.8 The frenchle/pro package

```

985 \@ifundefined{frenchTeXmods}{}{%
986 \caption@CheckCommand\@makecaption{%
987 % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
988 % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
989 \def\@makecaption#1#2{%
990 \ifFTY%
991 \def\@secondofmany##1##2\void{##2}%
992 \def\@tempa{\@secondofmany#2\void}%
993 \ifx\@tempa\empty%
994 \let\captionseparator\empty%
995 \fi%
996 \@mcORI{#1}{\relax\captionfont{#2}}%
997 \else
998 \@mcORI{#1}{#2}%
999 \fi}}
1000 \caption@CheckCommand\@makecaption{%
1001 % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
1002 \def\@makecaption#1#2{%
1003 \ifFTY%
1004 \def\@secondofmany##1##2\void{##2}%
1005 \protected@edef\@tempa{\@secondofmany#2\void}%
1006 \ifx\@tempa\empty%
1007 \let\captionseparator\empty%
1008 \fi%
1009 \@mcORI{#1}{\relax\captionfont{#2}}%
1010 \else
1011 \@mcORI{#1}{#2}%
1012 \fi}}
1013 \DeclareCaptionDefaultFont{textfont}{\itshape}%
1014 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
1015 }

```

1.20 Execution of options

```

1016 \captionsetup{style=default,position=default,%
1017 list,listformat=default,twoside=\if@twoside 1\else 0\fi}
1018 \ProcessOptions*
1019 \caption@IfCheckCommand{%

```

```

1020 \caption@setbool{documentclass}{1}%
1021 }{%
1022 \caption@setbool{documentclass}{0}%
1023 \PackageInfo{caption}{%
1024     Unknown document class (or package), \MessageBreak
1025     standard defaults will be used}%
1026 \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1027 }

```

1.21 Making an ‘List of’ entry

`\caption@addcontentsline` `\caption@addcontentsline{<type>}{<list entry>}`

Makes an entry in the list-of-whatever, if requested, i.e. the argument `<list entry>` is not empty and `listof=` was set to `true`.

```

1028 \newcommand*\caption@addcontentsline[2]{%
1029 \caption@iflist
1030   {\def\@tempa{#2}}%
1031   {\let\@tempa\@empty}%
1032 \ifx\@tempa\@empty \else
1033   {\let\\\space
1034     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1035                     {\protect\numberline
1036                      {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1037                      {\ignorespaces #2}}}%
1038 \fi}

```

1.22 Typesetting the caption

`\ifcaption@star` If the starred form of `\caption` is used, this will be set to `true`. (It will be reset to `false` at the end of `\caption@make`.)

```
1039 \newif\ifcaption@star
```

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

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

```
1040 \newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}
```

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

Typesets the caption.

```
1041 \newcommand\caption@make[2]{\caption@make{\caption@lfmt{#1}{#2}}}
```

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

```
1042 \newcommand\caption@@make[2]{%
```

```
1043 \begingroup
```

```
1044 \caption@stepcounter
```

```
1045 \caption@beginhook
```

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

```
1046 \ifx\caption@maxmargin\undefined \else
```

```
1047   \ifdim\captionmargin>\caption@maxmargin\relax
```

```
1048     \captionmargin\caption@maxmargin\relax
```

```
1049   \fi
```

```
1050 \fi
```

```
1051 \ifx\caption@minmargin\undefined \else
```



```

1052 \ifdim\captionmargin<\caption@minmargin\relax
1053 \captionmargin\caption@minmargin\relax
1054 \fi
1055 \fi

```

Special single-line treatment (option singlelinecheck=)

```

1056 \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%

```

Typeset the left margin (option margin=)

```

1057 \caption@calcmargin
1058 \@tempdima\captionmargin
1059 \ifdim\captionmargin@=\z@ \else
1060 \caption@ifoddpages{\advance\@tempdima\captionmargin@}%
1061 \fi
1062 \caption@ifh{\advance\@tempdima\caption@indent}%
1063 \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*`.

```

1064 \@tempdima\captionwidth
1065 \caption@ifh{\advance\@tempdima-\caption@indent}%
1066 \caption@parbox\@tempdima{%

```

Typeset the indentation (option indentation=)

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

```

1067 \caption@ifh{%
1068 \ifdim\caption@indent=\z@
1069 \leavevmode
1070 \else
1071 \hskip-\caption@indent
1072 \fi}%

```

Typeset the caption itself and close the `\caption@parbox`

```

1073 \caption@@@make{#1}{#2}{}%

```

Typeset the right margin (option margin=)

```

1074 \@tempdima\captionmargin
1075 \ifdim\captionmargin@=\z@ \else
1076 \caption@ifoddpages{\advance\@tempdima\captionmargin@}{}%
1077 \fi
1078 \hspace\@tempdima
1079 \caption@endhook
1080 \endgroup
1081 \global\caption@starfalse}

```

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

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

```

1082 \newcommand*\caption@calcmargin{%
1083 \ifdim\captionwidth=\z@
1084 \captionwidth\linewidth
1085 \advance\captionwidth by -2\captionmargin
1086 \advance\captionwidth by -\captionmargin@
1087 \else
1088 \captionmargin\linewidth

```

```

1089     \advance\captionmargin by -\captionwidth
1090     \divide\captionmargin by 2
1091     \captionmargin@z@
1092 \fi

1093 \caption@Debug{%
1094     \string\hsize=\the\hsize,
1095     \string\linewidth=\the\linewidth,\MessageBreak
1096     \string\leftmargin=\the\leftmargin,
1097     \string\rightmargin=\the\rightmargin,\MessageBreak
1098     \string\margin=\the\captionmargin,
1099     \string\margin@=\the\captionmargin@,
1100     \string\width=\the\captionwidth}%
1101 }

```

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

```

1102 \newcommand\caption@slc[4]{%
1103     \caption@Debug{Begin SLC}%
1104     \begingroup
1105     \caption@singleline
1106     \let\caption@hj@\empty
1107     \caption@calcmargin % calculate #3 if necessary
1108     \caption@prepareslc
1109     \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1110     \ifdim\wd\@tempboxa>#3%
1111         \endgroup
1112     \else
1113         \endgroup
1114         \caption@singleline
1115         #4%
1116     \fi
1117     \caption@Debug{End SLC}}

1118 \newcommand*\caption@singleline{%
1119     \caption@xsetup\caption@opt@singleline
1120     \let\caption@fmt\caption@slfmt}

```

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

```

1121 \newcommand*\caption@prepareslc{%
1122     \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1123     \def\label{\caption@withoptargs\@gobbletwo}%
1124     \let\stepcounter\caption@l@stepcounter
1125     \let\refstepcounter\stepcounter\let\H@refstepcounter\stepcounter}

1126 \newcommand*\caption@l@stepcounter[1]{\advance\c@#1\endcsname\@ne\relax}

```

`\caption@parbox` `\caption@parbox{<width>}{<contents>}`
This macro defines the box which surrounds the caption paragraph.

```

1127 \newcommand*\caption@parbox{\parbox[b]}

```

`\caption@@@make` `\caption@@@make{<caption label>}{<caption text>}`
This one finally typesets the caption paragraph, without margin and indentation.

```

1128 \newcommand\caption@@@make[2]{%

```

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

```
1129 \sbox\@tempboxa{#1}%
1130 \ifdim\wd\@tempboxa=\z@
1131 \let\caption@lsep\relax
1132 % \@capbreakfalse
1133 \fi
```

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

```
1134 \caption@ifempty{#2}{%
1135 \let\caption@lsep\relax
1136 % \@capbreakfalse
1137 % \let\caption@ifstrut\@secondoftwo
1138 }%
```

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

```
1139 \@setpar{\@@par\caption@@par}\caption@@par
```

Finally typeset the caption.

```
1140 \caption@hj\captionfont\captionsize\caption@fmt
1141 {\ifcaption@star\else\captionlabelfont#1}\fi}%
1142 {\ifcaption@star\else\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1143 {\captiontextfont
1144 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1145 \nobreak\hskip\z@skip % enable hyphenation
1146 \caption@tfmt{#2}%
1147 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1148 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1149 \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.

```
1150 \newcommand\caption@ifempty[1]{%
1151 \caption@ifempty{#1}%
1152 \caption@ifempty\@unused}
1153 \newcommand\caption@if@empty[1]{%
1154 \def\caption@tempa{#1}%
1155 \ifx\caption@tempa\@empty
1156 \let\caption@ifempty\@secondoftwo
1157 \else
1158 \expandafter\def\expandafter\caption@tempa\expandafter{%
1159 \caption@car#1\caption@ifempty\caption@nil}%
1160 \def\caption@tempb{\caption@ifempty}%
1161 \ifx\caption@tempa\caption@tempb
1162 \let\caption@ifempty\@secondoftwo
1163 \else
1164 \def\caption@tempb{\ignorespaces}%
1165 \ifx\caption@tempa\caption@tempb
1166 \expandafter\caption@ifempty\expandafter{\@gobble#1}%
1167 \else
1168 \def\caption@tempb{\label}%
```

```

1169     \ifx\caption@tempa\caption@tempb
1170     \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1171     \else
1172     \def\caption@tempb{\index}%
1173     \ifx\caption@tempa\caption@tempb
1174     \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1175     \else
1176     \def\caption@tempb{\glossary}%
1177     \ifx\caption@tempa\caption@tempb
1178     \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1179     \else
1180     \let\caption@ifempty@gobbletwo
1181     \fi
1182     \fi
1183     \fi
1184     \fi
1185     \fi
1186     \fi}
1187 \long\def\caption@car#1#2\caption@nil{#1}% same as \@car, but \long

```

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

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

```

1188 \newcommand*\caption@@par{%
1189   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

1.23 Types & sub-types

```

\DeclareCaptionType \DeclareCaptionType [⟨options⟩] [⟨environment⟩] [⟨name⟩] [⟨list name⟩]
1190 \newcommand*\DeclareCaptionType{%
1191   \@testopt\@DeclareCaptionType{} }
1192 \@onlypreamble\DeclareCaptionType
1193 \def\@DeclareCaptionType[#1]#2{%
1194   \def\caption@type{#2}%
1195   \caption@Debug{New type `#2'}%
1196   \newcounter{#2}\@namedef{theH#2}{}%
1197   \KV@caption@DCT@within\caption@within@default
1198   \KV@caption@DCT@placement{tbp}%
1199   \@ifundefined{c@float@type}%
1200     {\newcounter{float@type}%
1201      \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1202     {}%
1203   \caption@Debug{float type `#2'=\the\value{float@type}}%
1204   \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1205   \addtocounter{float@type}{\value{float@type}}%
1206   \KV@caption@DCT@fileext{lo#2}%
1207   \@namedef{fnum#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1208   \newenvironment{#2}{\@float{#2}}{\end@float}%
1209   \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1210   \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1211   \@ifundefined{l@figure}%
1212     {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1213     {\expandafter\let\csname l@#2\endcsname\l@figure}%

```

```

1214 \expandafter\newcommand\csname #2name\endcsname{}%
1215 \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
1216 \uppercase\expandafter{\@tempa}%
1217 \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
1218 \@tempb
1219 \expandafter\let\csname #2name\endcsname\@tempa
1220 \expandafter\newcommand\csname list#2name\endcsname{}%
1221 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1222 \@cons\caption@typelist{#2}}%
1223 \caption@setkeys[caption]{caption@DCT}{#1}%
1224 \ifundefined{float@exts}{\newtoks\float@exts}{}%
1225 \let\float@do=\relax
1226 \edef\@tempa{\noexpand\float@exts{\the\float@exts\float@do{\@nameuse{ext@#2}}}}%
1227 \@tempa
1228 \ifundefined{float@addtolists}{%
1229   \newcommand\float@addtolists[1]{%
1230     \def\float@do###1{\addtocontents{###1}{##1}}\the\float@exts}%
1231   \ifundefined{@chapter}{\caption@PatchChapter}}{}%
1232 \ifnextchar[\@@DeclareCaptionType\relax}
1233 \@onlypreamble\@@DeclareCaptionType
1234 \def\@@DeclareCaptionType[#1]{%
1235   \KV@caption@DCT@name{#1}%
1236   \ifnextchar[\@@@DeclareCaptionType\relax}
1237 \@onlypreamble\@@@DeclareCaptionType
1238 \def\@@@DeclareCaptionType[#1]{%
1239   \KV@caption@DCT@listname{#1}}
1240 \@onlypreamble\@@@DeclareCaptionType
1241 \let\DeclareFloatingEnvironment\DeclareCaptionType % old command name
1242 \@onlypreamble\DeclareFloatingEnvironment

```

`\caption@within@default` The default ‘within’ value.

```

1243 \newcommand*\caption@within@default{\ifundefined{c@chapter}{none}{chapter}}
1244 \@onlypreamble\caption@within@default

```

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

```

1245 \newcommand*\caption@listof[1]{%
1246   \begingroup
1247     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1248     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1249     \let\caption@ORI@starttoc\@starttoc
1250     \renewcommand*\@starttoc[1]{%
1251       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1252     \listoffigures
1253   \endgroup}

```

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

```

1254 \newcommand*\caption@typelist{}

```

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

```

1255 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}

```

```

1256 \@onlypreamble@key{caption@DCT}{fileext}
1257 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1258 \@onlypreamble@key{caption@DCT}{listname}
1259 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1260 \@onlypreamble@key{caption@DCT}{name}
1261 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
1262 \@onlypreamble@key{caption@DCT}{placement}
1263 \define@key{caption@DCT}{within}{%
1264   \@ifundefined{c@chapter}{}{\@removefromreset\caption@type{chapter}}%
1265   \@removefromreset\caption@type{section}}%
1266   \beginngroup
1267     \caption@setkeys[caption]{caption@within}{#1}%
1268   \endngroup
1269 \@onlypreamble@key{caption@DCT}{within}
1270 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1271 \@onlypreamble@key{caption@DCT}{without}

1272 \define@key{caption@within}{none}[]{%
1273   \caption@within{}}
1274 \@onlypreamble@key{caption@within}{none}
1275 \define@key{caption@within}{section}[]{%
1276   \@addtoreset\caption@type{section}}%
1277   \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
1278 \@onlypreamble@key{caption@within}{section}
1279 \@ifundefined{c@chapter}{}{%
1280   \define@key{caption@within}{chapter}[]{%
1281     \@addtoreset\caption@type{chapter}}%
1282     \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
1283 \@onlypreamble@key{caption@within}{chapter}}

```

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

```

1284 \newcommand*\caption@within{%
1285   \expandafter\caption@within\expandafter{\caption@type}}
1286 \@onlypreamble\caption@within
1287 \newcommand*\caption@within@[3]{%
1288   \global\@namedef{the#1}{#2\arabic{#1}}%
1289   \@ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
1290     {\global\@namedef{theH#1}{#3\arabic{#1}}}}
1291 \@onlypreamble\caption@within@

```

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

```

1292 \providecommand*\@removefromreset[2]{%
1293   \expandafter\let\csname c@#1\endcsname\@removefromreset
1294   \def\@elt##1{%
1295     \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1296     \else
1297       \noexpand\@elt{##1}%
1298     \fi}%
1299   \expandafter\xdef\csname cl@#2\endcsname{%
1300     \csname cl@#2\endcsname}}

```

`\caption@PatchChapter` **We try to patch `\@chapter` so `\float@addtolists` will be supported. (Note: The KOMA-Script classes already support `\float@addtolists`.)**

```

1301 \newcommand*\caption@PatchChapter{%
1302   \providecommand*\@chapterlistsgap{10\p@}%

1303   % report.cls [2005/09/16 v1.4f Standard LaTeX document class]
1304   \caption@patch@chapter{report}{%
1305     \ifnum \c@secnumdepth >\m@ne
1306       \refstepcounter{chapter}%
1307       \typeout{\@chapapp\space\thechapter.}%
1308       \addcontentsline{toc}{chapter}%
1309         {\protect\numberline{\thechapter}##1}%
1310     \else
1311       \addcontentsline{toc}{chapter}{##1}%
1312     \fi
1313     \chaptermark{##1}%
1314     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1315     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1316     \if@twocolumn
1317       \topnewpage[\@makechapterhead{##2}]%
1318     \else
1319       \@makechapterhead{##2}%
1320       \@afterheading
1321     \fi
1322   }{%
1323     \ifnum \c@secnumdepth >\m@ne
1324       \refstepcounter{chapter}%
1325       \typeout{\@chapapp\space\thechapter.}%
1326       \addcontentsline{toc}{chapter}%
1327         {\protect\numberline{\thechapter}##1}%
1328     \else
1329       \addcontentsline{toc}{chapter}{##1}%
1330     \fi
1331     \chaptermark{##1}%
1332     \ifdim \@chapterlistsgap>\z@
1333       \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1334       \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1335       \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1336     \fi
1337     \if@twocolumn
1338       \topnewpage[\@makechapterhead{##2}]%
1339     \else
1340       \@makechapterhead{##2}%
1341       \@afterheading
1342     \fi}%

1343   % book.cls [2005/09/16 v1.4f Standard LaTeX document class]
1344   \caption@patch@chapter{book}{%
1345     \ifnum \c@secnumdepth >\m@ne
1346       \if@mainmatter
1347         \refstepcounter{chapter}%
1348         \typeout{\@chapapp\space\thechapter.}%
1349         \addcontentsline{toc}{chapter}%
1350           {\protect\numberline{\thechapter}##1}%
1351       \else
1352         \addcontentsline{toc}{chapter}{##1}%
1353       \fi

```

```

1354 \else
1355 \addcontentsline{toc}{chapter}{##1}%
1356 \fi
1357 \chaptermark{##1}%
1358 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1359 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1360 \if@twocolumn
1361 \topnewpage[\@makechapterhead{##2}]%
1362 \else
1363 \makechapterhead{##2}%
1364 \afterheading
1365 \fi
1366 }{%
1367 \ifnum \c@secnumdepth >\m@ne
1368 \if@mainmatter
1369 \refstepcounter{chapter}%
1370 \typeout{\@chapapp\space\thechapter.}%
1371 \addcontentsline{toc}{chapter}%
1372 {\protect\numberline{\thechapter}##1}%
1373 \else
1374 \addcontentsline{toc}{chapter}{##1}%
1375 \fi
1376 \else
1377 \addcontentsline{toc}{chapter}{##1}%
1378 \fi
1379 \chaptermark{##1}%
1380 \ifdim \@chapterlistsgap>\z@
1381 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1382 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1383 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1384 \fi
1385 \if@twocolumn
1386 \topnewpage[\@makechapterhead{##2}]%
1387 \else
1388 \makechapterhead{##2}%
1389 \afterheading
1390 \fi}%
1391 % amsbook.cls [2004/08/06 v2.20]
1392 % smfbook.cls [1999/11/15 v1.2f Classe LaTeX pour les monographies editees par
1393 \caption@patch@chapter{ams/smfbook}{%
1394 \refstepcounter{chapter}%
1395 \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1396 \else \let\@secnumber\thechapter \fi
1397 \typeout{\chaptername\space\@secnumber}%
1398 \def\@toclevel{0}%
1399 \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1400 \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1401 \chaptermark{##1}%
1402 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1403 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1404 \makechapterhead{##2}\afterheading
1405 }{%
1406 \refstepcounter{chapter}%
1407 \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty

```



```

1408 \else \let\@secnumber\thechapter \fi
1409 \typeout{\chaptername\space\@secnumber}%
1410 \def\@toclevel{0}%
1411 \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1412 \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1413 \chaptermark{##1}%
1414 \ifdim \@chapterlistsgap>\z@
1415 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1416 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1417 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1418 \fi
1419 \@makechapterhead{##2}\@afterheading}%

1420 % scrreprt/scrbook.cls
1421 \@ifundefined{KOMAClassName}{}{%
1422 \caption@Debug{document class '\KOMAClassName' detected}%
1423 \let\caption@patch@chapter\@gobblethree}%

1424 % rapport1/3.cls [2004/06/07 v2.1a NTG LaTeX document class]
1425 \caption@patch@chapter{rapport}{%
1426 \ifnum \c@secnumdepth >\m@ne
1427 \refstepcounter{chapter}%
1428 \typeout{\@chapapp\space\thechapter.}%
1429 \addcontentsline{toc}{chapter}%
1430 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1431 \else
1432 \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1433 \fi
1434 \chaptermark{##1}%
1435 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1436 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1437 \if@twocolumn
1438 \topnewpage[\@makechapterhead{##2}]%
1439 \else
1440 \@makechapterhead{##2}%
1441 \@afterheading
1442 \fi
1443 }{%
1444 \ifnum \c@secnumdepth >\m@ne
1445 \refstepcounter{chapter}%
1446 \typeout{\@chapapp\space\thechapter.}%
1447 \addcontentsline{toc}{chapter}%
1448 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1449 \else
1450 \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1451 \fi
1452 \chaptermark{##1}%
1453 \ifdim \@chapterlistsgap>\z@
1454 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1455 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1456 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1457 \fi
1458 \if@twocolumn
1459 \topnewpage[\@makechapterhead{##2}]%
1460 \else

```

```

1461     \@makechapterhead{##2}%
1462     \@afterheading
1463   \fi}%

1464 % boek(3).cls [2004/06/07 v2.1a NTG LaTeX document class]
1465 \caption@patch@chapter{boek}{%
1466   \ifnum \c@secnumdepth >\m@ne
1467     \if@mainmatter
1468       \refstepcounter{chapter}%
1469       \typeout{\@chapapp\space\thechapter.}%
1470       \addcontentsline{toc}{chapter}%
1471         {\protect\numberline{\thechapter}\toc@font0 ##1}%
1472     \else
1473       \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1474     \fi
1475   \else
1476     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1477   \fi
1478   \chaptermark{##1}%
1479   \addtocontents{lof}{\protect\advspace{10\p@}}%
1480   \addtocontents{lot}{\protect\advspace{10\p@}}%
1481   \if@twocolumn
1482     \@topnewpage[\@makechapterhead{##2}]%
1483   \else
1484     \@makechapterhead{##2}%
1485     \@afterheading
1486   \fi
1487 }{%
1488   \ifnum \c@secnumdepth >\m@ne
1489     \if@mainmatter
1490       \refstepcounter{chapter}%
1491       \typeout{\@chapapp\space\thechapter.}%
1492       \addcontentsline{toc}{chapter}%
1493         {\protect\numberline{\thechapter}\toc@font0 ##1}%
1494     \else
1495       \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1496     \fi
1497   \else
1498     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1499   \fi
1500   \chaptermark{##1}%
1501   \ifdim \@chapterlistsgap>\z@
1502     \addtocontents{lof}{\protect\advspace{\@chapterlistsgap}}%
1503     \addtocontents{lot}{\protect\advspace{\@chapterlistsgap}}%
1504     \float@addtolists{\protect\advspace{\@chapterlistsgap}}%
1505   \fi
1506   \if@twocolumn
1507     \@topnewpage[\@makechapterhead{##2}]%
1508   \else
1509     \@makechapterhead{##2}%
1510     \@afterheading
1511   \fi}%

1512 % thesis.cls [1996/25/01 1.0g LaTeX document class (wm).]
1513 \caption@patch@chapter{thesis}{%

```

```

1514 \ifnum \c@secnumdepth >\m@ne
1515   \if@mainmatter
1516     \refstepcounter{chapter}%
1517     \typeout{\chaptername\space\thechapter.}
1518     \if@thema
1519       \ifx\@shortauthor\@empty
1520         \addcontentsline{toc}{chapter}{%
1521           \protect\numberline{\thechapter.}##1}%
1522       \else
1523         \addcontentsline{toc}{chapter}{%
1524           \protect\numberline{\thechapter.}%
1525           \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
1526       \fi
1527     \else
1528       \addcontentsline{toc}{chapter}{%
1529         \protect\numberline{\thechapter.}##1}%
1530     \fi
1531   \else
1532     \addcontentsline{toc}{chapter}{##1}
1533   \fi
1534 \else
1535   \addcontentsline{toc}{chapter}{##1}
1536 \fi
1537 \chaptermark{##1}
1538 \addtocontents{lof}{\protect\addvspace{10pt}}
1539 \addtocontents{lot}{\protect\addvspace{10pt}}
1540 \if@twocolumn
1541   \@topnewpage[\@makechapterhead{##2}]
1542 \else
1543   \@makechapterhead{##2}
1544   \@afterheading
1545 \fi
1546 }{%
1547 \ifnum \c@secnumdepth >\m@ne
1548   \if@mainmatter
1549     \refstepcounter{chapter}%
1550     \typeout{\chaptername\space\thechapter.}%
1551     \if@thema
1552       \ifx\@shortauthor\@empty
1553         \addcontentsline{toc}{chapter}{%
1554           \protect\numberline{\thechapter.}##1}%
1555       \else
1556         \addcontentsline{toc}{chapter}{%
1557           \protect\numberline{\thechapter.}%
1558           \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
1559       \fi
1560     \else
1561       \addcontentsline{toc}{chapter}{%
1562         \protect\numberline{\thechapter.}##1}%
1563     \fi
1564   \else
1565     \addcontentsline{toc}{chapter}{##1}%
1566   \fi
1567 \else

```

```

1568     \addcontentsline{toc}{chapter}{##1}%
1569     \fi
1570     \chaptermark{##1}%
1571     \ifdim \@chapterlistsgap>\z@
1572         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1573         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1574         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1575     \fi
1576     \if@twocolumn
1577         \@topnewpage[\@makechapterhead{##2}]%
1578     \else
1579         \@makechapterhead{##2}%
1580         \@afterheading
1581     \fi}%

1582 \ifx\caption@patch@chapter@gobblethree \else
1583     \caption@Debug{%
1584         Unsupported document class detected,\MessageBreak
1585         or \noexpand\@chapter was redefined by another package}%
1586 \fi
1587 \let\caption@PatchChapter\@undefined}
1588 \@onlypreamble\caption@PatchChapter

1589 \newcommand\caption@patch@chapter[3]{%
1590     \begingroup
1591 %     \let\if@twocolumn\iffalse
1592     \let\if@mainmatter\iffalse
1593     \let\if@thema\iffalse
1594     \def\@tempa[##1]##2{#2}%
1595     \ifx\@tempa\@chapter
1596         \caption@Debug{document class `#1' detected}%
1597         \gdef\@chapter[##1]##2{#3}%
1598         \global\let\caption@patch@chapter@gobblethree
1599     \fi
1600     \endgroup}
1601 \@onlypreamble\caption@patch@chapter
1602 \long\def \@gobblethree #1#2#3{}

\@stpelt We patch \@stpelt so a list of ‘connected’ counters will be reset, too. (Like
\stepcounter does in ltcounts.dtx.)
1603 \newcommand*\caption@patch@stpelt{%
1604     \let\caption@stpelt\@stpelt
1605     \def\@stpelt##1{%
1606         \caption@stpelt{##1}%
1607         \begingroup
1608             \let\@elt\caption@stpelt
1609             \csname caption@cl@##1\endcsname
1610         \endgroup}%
1611     \let\caption@patch@stpelt\relax}
1612 \@onlypreamble\caption@patch@stpelt

\caption@addtoreset Like \@addtoreset from ltcounts.dtx
1613 \newcommand*\caption@addtoreset[2]{%
1614     \caption@patch@stpelt
1615     \@ifundefined{caption@cl@#2}{\@namedef{caption@cl@#2}{}}{}%

```

```
1616 \expandafter\@cons\csname caption@cl@#2\endcsname{{#1}}
1617 \@onlypreamble\caption@addtoreset
```

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

```
1618 \newcommand*\caption@removefromreset[2]{%
1619 \begingroup
1620 \expandafter\let\csname c@#1\endcsname\caption@removefromreset
1621 \def\@elt##1{%
1622 \expandafter\ifx\csname c@##1\endcsname\caption@removefromreset
1623 \else
1624 \noexpand\@elt{##1}%
1625 \fi}%
1626 \expandafter\xdef\csname caption@cl@#2\endcsname{%
1627 \csname caption@cl@#2\endcsname}%
1628 \endgroup}
1629 \@onlypreamble\caption@removefromreset
```

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

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

```
1630 \newcommand*\DeclareCaptionSubType{%
1631 \caption@teststar\@DeclareCaptionSubType\@firstoftwo\@secondoftwo}
1632 \@onlypreamble\DeclareCaptionSubType

1633 \newcommand*\@DeclareCaptionSubType[1]{%
1634 \@testopt{\@@DeclareCaptionSubType{#1}}{alph}}
1635 \@onlypreamble\@DeclareCaptionSubType

1636 \def\@@DeclareCaptionSubType#1[#2]#3{%
1637 \@ifundefined{c@#3}%
1638 {\caption@Error{No float type '#3' defined}}%
1639 {\@ifundefined{c@sub#3}%
1640 {\caption@Debug{New subtype `sub#3'}}%
1641 \newcounter{sub#3}%
1642 \caption@addtoreset{sub#3}{#3}%
1643 \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1644 \@ifundefined{l@chapter}%
1645 {\edef\@tempa{\expandafter\expandafter\expandafter\noexpand
1646 \expandafter\@car\l@subsubsection\@nil}%
1647 \def\@tempb{\@dottedtocline}%
1648 \ifx\@tempa\@tempb % \l@subsubsection starts with \@dottedtocline
1649 \expandafter\edef\csname l@sub#3\endcsname{%
1650 \noexpand\@dottedtocline{2}%
1651 \expandafter\expandafter\expandafter\noexpand
1652 \expandafter\@gobbletwo\l@subsubsection}%
1653 \else
1654 \@namedef{l@sub#3}{\@dottedtocline{2}{3.8em}{3.2em}}%
1655 \fi}%
1656 {\expandafter\let\csname l@sub#3\endcsname\l@subsection}%
1657 \@cons\caption@subtypelist{{#3}}}%
1658 {\caption@Debug{Modify caption `sub#3'}}}%

1659 \@namedef{sub#3name}{}%
1660 \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
```

```

1661     #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
1662     {\@namedef{p@sub#3}}}%
1663     \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1664     {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1665     \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1666     \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}}%
1667     }}
1668 \@onlypreamble\@@DeclareCaptionSubType

```

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

```

1669 \newcommand*\caption@subtypelist{}

```

```

\caption@For \caption@For{<elt-list>}{<code with #1>}
\caption@For*{<elt-list>}{<code with #1>}
1670 \newcommand*\caption@For{\caption@withoptargs\caption@@For}
1671 \@onlypreamble\caption@For
1672 \newcommand\caption@@For[3]{%
1673   \caption@AtBeginDocument#1{%
1674     \def\@elt##1{##3}%
1675     \@nameuse{caption@#2}%
1676     \let\@elt\relax}}%
1677 \@onlypreamble\caption@@For

```

1.24 subfig package adaption

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

```

1678 \caption@AtBeginDocument{%
1679   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1680   \ifx\subref\@tempa
1681     \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1682     \let\caption@setfloattype\@gobble
1683     \let\@dottedxxxline\sf@NEW@dottedxxxline
1684     \let\sf@subfloat\sf@NEW@subfloat

```

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

```

1685   \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1686   \fi
1687   \let\sf@NEW@dottedxxxline\@undefined
1688   \let\sf@NEW@subfloat\@undefined}
1689 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1690   \begingroup
1691     \caption@setfloattype{#1}%
1692     \caption@setoptions{subfloat}%
1693     \caption@setoptions{sub#1}%
1694     \ifnum #3>\@nameuse{c@#2depth}\else
1695       \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1696     \fi
1697   \endgroup}

```

```

1698 \def\sf@NEW@subfloat{%
1699   \begingroup
1700     \caption@setfloattype\@capttype
1701     \sf@ifpositiontop{%
1702       \maincaptiontoptrue
1703     }{%
1704       \maincaptiontopfalse
1705     }%
1706     \caption@setoptions{subfloat}%
1707     \caption@setoptions{sub\@capttype}%
1708     \let\sf@oldlabel=\label
1709     \let\label=\subfloat@label
1710     \ifmaincaptiontop\else
1711       \advance\@nameuse{c@\@capttype}\@ne
1712     \fi
1713     \refstepcounter{sub\@capttype}%
1714     \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1715     \@ifnextchar [% %] match left bracket
1716     {\sf@@subfloat}%
1717     {\sf@@subfloat[\@empty]}

```

2 Main package

2.1 Identification

```
1718 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1719 \ProvidesPackage{caption}[2009/10/09 v3.1k Customizing captions (AR)]
1720 %\@ifundefined{PackageRedefines}{}{\PackageRedefines{caption}{caption}}
```

`\caption@Info` *Note:* The `\@gobble` at the end of the 2nd argument of `\PackageInfo` suppresses the line number info. See TLC2[1], A.4.7, p885 for details.

```
1721 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1722 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1723 \RequirePackage{caption3}[2008/08/24] % needs v3.1j or newer
```

2.3 Check against incompatible document classes

```
1724 \caption@ifbool{documentclass}{}{%
1725   \caption@WarningNoLine{%
1726     Unsupported document class (or package) detected,\MessageBreak
1727     usage of the caption package is not recommended}%
1728   \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1729 }
```

2.4 Check against incompatible packages

```
1730 \@ifpackageloaded{caption2}{}%
1731   \caption@Error{%
1732     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1733     the (current) caption package}%
1734   \endinput
1735 }{}

1736 \caption@AtBeginDocument{%
1737   \@ifpackageloaded{ftcap}{\caption@DisablePositionOption{ftcap}}{}%
1738   \@ifpackageloaded{nonfloat}{\caption@DisablePositionOption{nonfloat}}{}%
1739   \@ifpackageloaded{topcapt}{\caption@DisablePositionOption{topcapt}}{}%
\caption@DisablePositionOption \caption@DisablePositionOption{package}
disables the 'position' option.

1740 \newcommand*\caption@DisablePositionOption[1]{%
1741   \caption@Info{%
1742     '#1' package detected; setting 'position=b' for compatibility reasons}%
1743   \caption@setposition b%

1744   \DeclareCaptionOption{position}{}%
1745   \caption@Error{Usage of the 'position' option is incompatible\MessageBreak
1746     to the '#1' package}}

1747 \@onlypreamble\caption@DisablePositionOption
```

2.5 Declaration of options

2.5.1 Options for figure and table


```

1748 \DeclareCaptionOption{figureposition}{%
1749   \captionsetup*[figure]{position=#1}}
1750 \@onlypreamble@key{caption}{figureposition}

1751 \DeclareCaptionOption{tableposition}{%
1752   \captionsetup*[table]{position=#1}}
1753 \@onlypreamble@key{caption}{tableposition}

1754 \DeclareCaptionOption{figurename}{\caption@SetName{figure}{#1}}
1755 \DeclareCaptionOption{tablename}{\caption@SetName{table}{#1}}
1756 \DeclareCaptionOption{name}{\caption@setname\@capttype{#1}}

1757 \DeclareCaptionOption{listfigurename}{\caption@SetName{listfigure}{#1}}
1758 \DeclareCaptionOption{listtablename}{\caption@SetName{listtable}{#1}}

\caption@SetName \caption@SetName{<cmd>}{<value>}
1759 \newcommand*\caption@SetName[2]{%
1760   \caption@setname{#1}{#2}%
1761   \begingroup
1762     \ifundefined{langaugename}{}{%
1763       \ifundefined{captions\langaugename}{}{%
1764         \expandafter\g@addto@macro\csname captions\langaugename\endcsname
1765           {\caption@setname{#1}{#2}}}%
1766   \endgroup}

1767 \newcommand*\caption@setname[2]{\@namedef{#1name}{#2}}

1768 \caption@AtBeginDocument{\let\caption@SetName\caption@setname}

1769 \DeclareCaptionOption{figurewithin}{\caption@Within{figure}{#1}}
1770 \@onlypreamble@key{caption}{figurewithin}
1771 \DeclareCaptionOption{figurewithout}{\KV@caption@figurewithin{none}}
1772 \@onlypreamble@key{caption}{figurewithout}

1773 \DeclareCaptionOption{tablewithin}{\caption@Within{table}{#1}}
1774 \@onlypreamble@key{caption}{tablewithin}
1775 \DeclareCaptionOption{tablewithout}{\KV@caption@tablewithin{none}}
1776 \@onlypreamble@key{caption}{tablewithout}

1777 \DeclareCaptionOption{within}{%
1778   \ifundefined{c@figure}{}{\caption@Within{figure}{#1}}%
1779   \ifundefined{c@table}{}{\caption@Within{table}{#1}}%
1780   \def\caption@within@default{#1}}
1781 \@onlypreamble@key{caption}{within}
1782 \DeclareCaptionOption{without}{\KV@caption@within{none}}
1783 \@onlypreamble@key{caption}{without}

\caption@within
1784 \newcommand*\caption@Within[1]{\def\caption@type{#1}\KV@caption@DCT@within}
1785 \@onlypreamble\caption@Within

```

2.5.2 Miscellaneous options

```

1786 \DeclareCaptionOption*{config}[caption]{%
1787   \InputIfFileExists{#1.cfg}%
1788   {\typeout{*** Local configuration file #1.cfg used ***}}%
1789   {\caption@Warning{Configuration file #1.cfg not found}}}

1790 \DeclareCaptionOption{@minipage}{%

```

```

1791 \caption@ifinlist{#1}{auto,default}%
1792     {\let\caption@if@minipage\@gobbletwo}%
1793     {\caption@set@bool\caption@if@minipage{#1}}
1794 \captionsetup{@minipage=default}

```

2.5.3 caption v1.x compatibility options

```

1795 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1796 \@onlypreamble@key{caption}{compatibility}

1797 \DeclareCaptionOptionNoValue*{normal}{%
1798   \caption@setformat{plain}%
1799   \caption@setjustification{justified}}
1800 \DeclareCaptionOptionNoValue*{isu}{%
1801   \caption@setformat{hang}%
1802   \caption@setjustification{justified}}
1803 \DeclareCaptionOptionNoValue*{hang}{%
1804   \caption@setformat{hang}%
1805   \caption@setjustification{justified}}
1806 \DeclareCaptionOptionNoValue*{center}{%
1807   \caption@setformat{plain}%
1808   \caption@setjustification{centering}}
1809 \DeclareCaptionOptionNoValue*{anne}{%
1810   \caption@setformat{plain}%
1811   \caption@setjustification{centerlast}}
1812 \DeclareCaptionOptionNoValue*{centerlast}{%
1813   \caption@setformat{plain}%
1814   \caption@setjustification{centerlast}}

1815 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1816 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1817 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1818 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1819 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1820 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1821 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1822 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1823 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1824 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1825 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1826 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1827 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1828 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1829 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1830 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}

1831 \caption@setbool{ruled}{0}
1832 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}

```

2.5.4 caption2 v2.x compatibility options

```

1833 \DeclareCaptionOptionNoValue*{flushleft}{%
1834   \caption@setformat{plain}%
1835   \caption@setjustification{raggedright}}
1836 \DeclareCaptionOptionNoValue*{flushright}{%
1837   \caption@setformat{plain}%
1838   \caption@setjustification{raggedleft}}

```

```

1839 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}
1840 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
1841 \caption@WarningNoLine{Obsolete option 'ignoreLTcapwidth' ignored}}

```

2.5.5 Obsolete caption v3.0 options

```

1842 \DeclareCaptionOption*{caption}{%
1843 \caption@setbool{temp}{#1}%
1844 \caption@ifbool{temp}{}{%
1845 \caption@Error{%
1846 The package option 'caption=#1' is obsolete.\MessageBreak
1847 Please pass this option to the subfig package instead\MessageBreak
1848 and do *not* load the caption package anymore}}}}

```

2.5.6 fltpage package support options

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

```

1849 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1850 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1851 \@onlypreamble@key{caption}{FPlist}
1852 \@onlypreamble@key{caption}{FPref}

1853 \newcommand*\caption@setFPoption[2]{%
1854 \edef\caption@tempa{\@car#2@nil}%
1855 \caption@setbool{FP#1cap}{\if c\caption@tempa 1\else 0\fi}}
1856 \@onlypreamble\caption@setFPoption

1857 \captionsetup{FPlist=caption,FPref=figure}

```

2.5.7 hyperref package support options

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

```

1858 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1859 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}

1860 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}

```

2.6 $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes support

```

1861 \@ifundefined{@captionheadfont}{}{%
1862 \caption@Info{AMS or SMF document class}%
1863 \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1864 }

```

2.7 KOMA-Script document classes support

```

1865 \@ifundefined{scr@caption}{}{%
1866 \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 1867 \g@addto@macro\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1868 \g@addto@macro\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

```

```

1869 \if@tablecaptionabove
1870 \@tablecaptionabovetrue
1871 \else
1872 \@tablecaptionabovefalse
1873 \fi

\onelinecaptionstrue
\onelinecaptionfalse 1874 \g@addto@macro\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1875 \g@addto@macro\onelinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1876 \ifonelinecaptions
1877 \onelinecaptionstrue
1878 \else
1879 \onelinecaptionfalse
1880 \fi

\@captionabovetrue Please note that these are stronger than the position setting, therefore we override the
\@captionabovefalse options figureposition and tableposition to typeout a warning.

1881 \g@addto@macro\@captionabovetrue{\let\caption@position\@firstoftwo}
1882 \g@addto@macro\@captionabovefalse{\let\caption@position\@secondoftwo}

1883 \DeclareCaptionOption{figureposition}{%
1884 \caption@WarningNoLine{Option 'figureposition=#1' has no effect\MessageBreak
1885 when used with a KOMA script document class}}
1886 \DeclareCaptionOption{tableposition}{%
1887 \caption@WarningNoLine{Option 'tableposition=#1' has no effect\MessageBreak
1888 when used with a KOMA script document class}}

\setcapindent

1889 \let\caption@KOMA@setcapindent\@setcapindent
1890 \renewcommand*\@setcapindent[1]{%
1891 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

1892 \let\caption@KOMA@@setcapindent\@@setcapindent
1893 \renewcommand*\@@setcapindent[1]{%
1894 \caption@KOMA@@setcapindent{#1}\caption@setcapindent}

1895 \newcommand*\caption@setcapindent{%
1896 \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}

1897 \@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 a negative effect when used by the captionbeside
environment, we suppress the warning here.)

1898 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1899 \csname\string\setcapwidth\endcsname
1900 \@namedef{\string\setcapwidth}[#1]#2{%
1901 \caption@KOMA@setcapwidth[#1]#2}\caption@setcapwidth{#1}}

1902 \newcommand*\caption@setcapwidth[1]{%
1903 \ifx\#1\else
1904 \ifundefined{cap@margin}{}{%
1905 \def\@tempa{captionbeside}%
1906 \ifx\@tempa\@currenvir\else\caption@Warning{%
1907 Ignoring optional argument [#1] of \string\setcapwidth\MessageBreak}%

```

```

1908         \fi}%
1909     \fi
1910     \captionsetup{width=\cap@width}

1911 \def\caption@tempa{\hspace}%
1912 \ifx\caption@tempa\cap@width \else
1913     \caption@setcapwidth{?}
1914 \fi

\setcapmargin

1915 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1916         \csname\string\@setcapmargin\endcsname
1917 \@namedef{\string\@setcapmargin}[#1]#2{%
1918     \caption@KOMA@setcapmargin[#1]{#2}\caption@setcapmargin}

1919 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1920         \csname\string\@@setcapmargin\endcsname
1921 \@namedef{\string\@@setcapmargin}[#1]#2{%
1922     \caption@KOMA@@setcapmargin[#1]{#2}\caption@setcapmargin}

1923 \newcommand*\caption@setcapmargin{%
1924     \begingroup
1925         \let\onelinecaptionsfalse\relax
1926         \def\@twoside{0}%
1927         \def\if@twoside{\def\@twoside{1}\iffalse}%
1928         \cap@margin
1929         \def\@tempa{\endgroup}%
1930         \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1931             \def\hspace##1##{\@firstofone}%
1932             \edef\@tempa{\endgroup
1933                 \noexpand\captionsetup{%
1934                     twoside=\@twoside,slc=0,%
1935                     margin={\cap@left,\cap@right}}}%
1936         \fi\fi
1937         \@tempa}

1938 \ifx\cap@margin\relax \else
1939     \caption@setcapmargin
1940 \fi

1941 }

```

2.8 Processing of options

```
1942 \caption@ProcessOptions*{caption}
```

2.9 \captionof and \captionlistentry

```

1943 \caption@AtBeginDocument{%

1944     \DeclareCaptionOption{type}{\caption@settype{#1}}%
1945     \DeclareCaptionOption{type*}{\caption@settype*{#1}}%

1946     \DeclareCaptionOption{subtype}[sub\@capttype]{\caption@setsubtype{#1}}%
1947     \DeclareCaptionOption{subtype*}[sub\@capttype]{\caption@setsubtype*{#1}}%

1948 }

```

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

```

\caption@settype \caption@settype*{<type>}
sets \@capttype and executes the options associated with it (using \caption@setoptions).
Furthermore we check \currentgrouplevel (if avail), redefine \@currentlabel
so a \label before \caption will result in a hint instead of a wrong reference, and
use the macro \caption@(sub)typehook (which will be used by our float package
support).
The non-starred version sets a hyperref anchor additionally (if hycap=true and the
hycap package is not loaded).
1949 \newcommand*\caption@settype{%
1950   \caption@@settype{}}

1951 \newcommand*\caption@setsubtype{%
1952   \caption@iftype
1953     {\caption@@settype{sub}}%
1954     {\caption@Error{Option `subtype=' outside float}}}%

1955 \newcommand*\caption@@settype[1]{%
1956   \caption@teststar{\caption@@@settype{#1}}\@firstoftwo\@secondoftwo}

1957 \newcommand*\caption@@@settype[3]{%
1958 % #1 = "" or "sub"
1959 % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1960 % #3 = <type>, e.g. "figure" or "table"
1961   \@ifundefined{c@#3}%
1962     {\caption@Error{No float type '#3' defined}}%
1963     {\caption@Debug{#1type=#3}}%
1964     \caption@checkgrouplevel{#1}{%
1965       \captionsetup{#1type#2*\@empty=...}#2{ or
1966         \@backslashchar#1captionof}{}}%

1967   \edef\caption@tempa{#3}%
1968   \expandafter\ifx\csname @#1capttype\endcsname\caption@tempa \else
1969     \ifcaptionsetup@star\else\@nameuse{caption@#1type@warning}\fi
1970   \fi
1971   \expandafter\let\csname @#1capttype\endcsname\caption@tempa
1972   \@nameuse{caption@#1typehook}%

1973   \caption@setoptions{#3}%
1974   \ifx\caption@opt\relax
1975     \@nameundef{caption@#1type@warning}%
1976   \else
1977     \@namedef{caption@#1type@warning}{\caption@Warning{%
1978       The #1caption type was already set to
1979       '\csname @#1capttype\endcsname'\MessageBreak}}%
1980   \fi

1981   \let\caption@ifrefstepcounter\@secondoftwo
1982   #2{}%
1983   \let\@currentlabel\caption@undefinedlabel
1984 % \let\@currentHlabel\@undefined
1985   \ifx\caption@ORI@label\@undefined
1986     \let\caption@ORI@label\label
1987     \let\label\caption@xlabel

```

```
1988     \fi
1989     \caption@start}}}
```

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

```
1990 \newcommand*\caption@typehook{}
```

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

```
1991 \newcommand*\caption@iftype{%
1992   \@ifundefined{@capytype}{\let\@capytype\@undefined\@secondoftwo}\@firstoftwo}
```

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

```
1993 \begingroup\expandafter\expandafter\expandafter\endgroup
1994 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1995   \caption@Debug{TeX engine: TeX}
1996   \let\caption@checkgrouplevel\@gobbletwo
1997 \else
1998   \caption@Debug{TeX engine: e-TeX}
1999   \newcommand*\caption@checkgrouplevel[2]{%
2000     \@ifundefined{#1caption@grouplevel}{%
2001       \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}{}%
2002       \ifnum\currentgrouplevel>\caption@grouplevel\relax
2003         \expandafter\edef\csname #1caption@grouplevel\endcsname{%
2004           \the\currentgrouplevel}%
2005       \else
2006         \caption@Warning{\string#2\MessageBreak outside box or environment}%
2007       \fi
2008     }{}}
2009 \fi
```

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

```
2010 \newcommand*\caption@undefinedlabel{%
2011   \protect\caption@xref{\caption@labelname}{\on@line}}
2012 \DeclareRobustCommand*\caption@xref[2]{%
2013   \caption@WarningNoLine{\noexpand\label before \string\caption#2}%
2014   \@setref\relax\@undefined{#1}}
2015 \newcommand*\caption@labelname{??}
```

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

```
2016 \newcommand*\caption@xlabel[1]{%
2017   \caption@@xlabel
2018   \def\caption@labelname{#1}%
2019   \caption@ORI@label{#1}}
2020 \newcommand*\caption@@xlabel{%
2021   \global\let\caption@@xlabel\@empty
2022   \@bsphack
2023   \protected@write\@auxout{}%
2024   {\string\providecommand*\string\caption@xref[2]{%
```

```

2025         \string\@setref\string\relax\string\@undefined{\string##1}}}%
2026     \@esphack}

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

2027 \caption@AtBeginDocument{%
2028     \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}
2029 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

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

2030 \newcommand*\captionlistentry{%
2031     \caption@teststar\@captionlistentry\@firstoftwo\@secondoftwo}

2032 \newcommand*\@captionlistentry[1]{%
2033     \@testopt{\caption@listentry{#1}}\@capttype}

2034 \def\caption@listentry#1[#2]#3{%
2035     \@bsphack
2036     #1{\def\@currentlabelname{#3}}%
2037         {\caption@refstepcounter{#2}}%
2038         \caption@makecurrent{#2}{#3}}%
2039     \caption@addcontentsline{#2}{#3}%
2040     \@esphack}

```

2.10 \ContinuedFloat

```

\ContinuedFloat \ContinuedFloat
\ContinuedFloat*

```

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

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

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

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

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.

```

2041 \def\ContinuedFloat{%
2042     \@ifnextchar[\@Continued@Float\@ContinuedFloat}

```



```

2043 \def\@Continued@Float[#1]{\addtocounter{#1}\m@ne}
2044 \def\@ContinuedFloat{%
2045   \caption@iftype
2046     {\addtocounter\@capttype\m@ne
2047     \caption@ContinuedFloat\@capttype}%
2048     {\caption@Error{\noexpand\ContinuedFloat outside float}}}
2049 \def\caption@ContinuedFloat#1{%
2050   \ifstar{\caption@Continued@Float@{#1}}{\caption@Continued@Float{#1}}}
2051 \def\caption@Continued@Float@{%
2052   \addtocounter\@capttype\@ne
2053   \@stpelt{ContinuedFloat}\stepcounter{ContinuedFloat}%
2054   \def\caption@resetContinuedFloat##1{\xdef\caption@CFtype{##1}}%
2055   \caption@@ContinuedFloat}
2056 \def\caption@Continued@Float#1{%
2057   \edef\caption@tempa{#1}%
2058   \ifx\caption@tempa\caption@CFtype
2059     \stepcounter{ContinuedFloat}%
2060     \let\caption@resetContinuedFloat\@gobble
2061     \caption@@ContinuedFloat{#1}%
2062     \sf@ContinuedFloat{#1}%
2063   \else
2064     \caption@Error{Continued `#1' after '\caption@CFtype'}%
2065   \fi}
2066 \def\caption@@ContinuedFloat#1{%
2067   \expandafter\l@addto@macro\c@name the#1\endcsname\theContinuedFloat
2068   \@ifundefined{theH#1}{}{%
2069     \expandafter\l@addto@macro\c@name theH#1\endcsname{%
2070       \alph@c@ContinuedFloat}}%
2071   \caption@setoptions{ContinuedFloat}%
2072   \caption@setoptions{continued#1}}
2073 \providecommand*\sf@ContinuedFloat[1]{}
2074 \newcommand*\caption@CFtype{??}

```

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

```

2075 \newcounter{ContinuedFloat}
2076 \let\theContinuedFloat\@empty

```

`\caption@resetContinuedFloat` `\caption@resetContinuedFloat{<type>}`
 If a continuation counter is defined, we reset it. (This one will be called inside `\@caption`.)

```

2077 \newcommand*\caption@resetContinuedFloat[1]{%
2078   \@stpelt{ContinuedFloat}\xdef\caption@CFtype{#1}}

```

2.11 Internal helpers

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

```

2079 \newcommand*\caption@refstepcounter[1]{%
2080   \caption@resetContinuedFloat{#1}%

```

```

2081 \caption@@refstepcounter{#1}%
2082 \let\caption@ifrefstepcounter\@firstoftwo

2083 \newcommand*\caption@@refstepcounter{\refstepcounter}
2084 \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[]{}).
2085 \@ifundefined{kernel@ifnextchar}%
2086 {\newcommand\caption@dblarg[1]{\@ifnextchar[{\#1}{\caption@xdblarg{\#1}}]}%
2087 {\newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{\#1}}]}%
2088 \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)

2089 \newcommand*\caption@begin[1]{%
2090 \begingroup
2091 \caption@setfnum{\#1}%
2092 \caption@fixposition
2093 \global\let\caption@fixedposition\caption@position}

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

    4. Make an entry in the list-of-whatever.

    5. Set \caption@ifempty according argument <heading>.

2094 \newcommand\caption@beginex[3]{%
2095 \caption@begin{\#1}%
2096 \caption@addcontentsline{\#1}{\#2}%
2097 \caption@ifempty{\#3}{}}

\caption@end \caption@end closes the group.
2098 \newcommand*\caption@end{%
2099 \endgroup
2100 \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.
2101 \newcommand*\caption@setfnum[1]{%
2102 \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
2103 \@namedef{fnum@#1}{\caption@fnum{\#1}}%
2104 \fi}

```

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

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

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

```
2105 \newcommand*\caption@boxrestore{%
2106   \let\if@nobreak\iffalse
2107   \let\if@noskipsec\iffalse
2108   \let\par\@par
2109 % \let-\@dischyph
2110 % \let'\@acci\let'\@accii\let\=\@acciii
2111   \parindent\z@ \parskip\z@skip
2112   \everypar{}%
2113 % \linewidth\hsize
2114 % \@totalleftmargin\z@
2115   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
2116   \parfillskip\@flushglue \lineskip\normallineskip
2117   \baselineskip\normalbaselineskip
2118   \sloppy
2119   \let\\\@normalcr
2120 }
```

`\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).

```
2121 \newcommand*\caption@normalsize{%
2122   \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.**

```
2123 \let\caption@setfloatcapt\@firstofone
```

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

```
\caption@makeanchor 2124 \newcommand*\caption@makecurrent[2]{}
\caption@start      2125 \let\caption@makeanchor\@firstofone
\caption@@start
\caption@freezeHref
\caption@defrostHref
```

```

2126 \let\caption@start\relax
2127 \let\caption@@start\relax
2128 \let\caption@freezeHref\relax
2129 \let\caption@defrostHref\relax

```

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

`\caption@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.

```

2130 \def\caption@caption{%
2131   \caption@iftype
2132     {\caption@checkgrouplevel\@empty\caption
2133      \caption@star
2134       {\caption@refstepcounter\@capttype}%
2135       {\caption@dblarg{\@caption\@capttype}}}%
2136     {\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.

```

2137 \newcommand*\caption@star[2]{%
2138   \ifstar{\global\caption@startrue#2[]}{#1#2}}%

```

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

```

2139 \long\def\caption@@caption#1[#2]#3{%
2140   \ifcaption@star \else
2141     \caption@prepareanchor{#1}{#2}%
2142     \fi
2143   \par
2144   \caption@beginex{#1}{#2}{#3}%
2145     \caption@setfloatcapt{%
2146       \caption@boxrestore
2147       \if@minipage
2148         \@setminipage
2149       \fi
2150       \caption@normalsize
2151       \ifcaption@star
2152         \let\caption@makeanchor\@firstofone
2153       \fi
2154       \@makecaption{\csname fnum@#1\endcsname}%
2155         {\ignorespaces\caption@makeanchor{#3}}\par
2156     \caption@if@minipage\@minipagetrue\@minipagefalse}%
2157   \caption@end}%

```

`\caption@prepareanchor`

```

2158 \newcommand*\caption@prepareanchor[2]{%
2159   \caption@makecurrent{#1}{#2}%
2160   \caption@ifhyppcap\caption@@start{}}

```

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

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

```

2161 \long\def\caption@makecaption#1#2{%
2162   \caption@iftop
2163     {\vskip\belowcaptionskip}%
2164     {\caption@rule\vskip\abovecaptionskip}%
2165   \caption@@make{#1}{#2}%
2166   \caption@iftop
2167     {\vskip\abovecaptionskip\caption@rule}%
2168     {\vskip\belowcaptionskip}}

```

\caption@redefine 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.

```

2169 \newcommand*\caption@redefine{}
2170 \g@addto@macro\caption@redefine{%
2171   \caption@setbool{incompatible}{0}%
2172   \caption@CheckCommand\caption{%
2173     % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2174     \def\caption{%
2175       \ifx\@capttype\@undefined
2176         \@latex@error{\noexpand\caption outside float}\@ehd
2177         \expandafter\@gobble
2178       \else
2179         \refstepcounter\@capttype
2180         \expandafter\@firstofone
2181         \fi
2182       {\@dblarg{\@caption\@capttype}}%
2183     }%
2184   \caption@CheckCommand\caption{%
2185     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
2186     \def\caption{
2187       \ifx\@capttype\@undefined
2188         \@latex@error{\noexpand\caption outside figure or table}\@ehd
2189         \expandafter\@gobble
2190       \else
2191         \refstepcounter\@capttype
2192         \expandafter\@firstofone
2193         \fi
2194       {\@dblarg{\@caption\@capttype}}%
2195     }%
2196   \caption@CheckCommand\caption{%
2197     % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2198     \renewcommand\caption{%

```

```

2199 \ifx\@capttype\@undefined
2200 \latex@error{\noexpand\caption outside float}\@ehd
2201 \expandafter\@gobble
2202 \else
2203 \refstepcounter\@capttype
2204 \let\@tempf\@caption
2205 \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
2206 \expandafter\expandafter\let
2207 \expandafter\@tempf\csname @float@c@\@capttype\endcsname
2208 \fi
2209 \fi
2210 \@dblarg{\@tempf\@capttype}}}%
2211 \caption@CheckCommand\caption{%
2212 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2213 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2214 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2215 \def\caption{%
2216 \ifx\@capttype\@undefined
2217 \latex@error{\noexpand\caption outside float}\@ehd
2218 \expandafter\@gobble
2219 \else
2220 \H@refstepcounter\@capttype
2221 \@ifundefined{fst@\@capttype}{%
2222 \let\Hy@tempa\@caption
2223 }{%
2224 \let\Hy@tempa\Hy@float@caption
2225 }%
2226 \expandafter\@firstofone
2227 \fi
2228 {\@dblarg{\Hy@tempa\@capttype}}%
2229 }}%
2230 \caption@CheckCommand\caption{%
2231 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2232 \def\caption{%
2233 \ifx\@capttype\@undefined
2234 \latex@error{\noexpand\caption outside float}\@ehd
2235 \expandafter\@gobble
2236 \else
2237 \H@refstepcounter\@capttype
2238 \let\Hy@tempa\@caption
2239 \@ifundefined{float@caption}{%
2240 }{%
2241 \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
2242 \let\Hy@tempa\Hy@float@caption
2243 \fi
2244 }%
2245 \expandafter\@firstofone
2246 \fi
2247 {\@dblarg{\Hy@tempa\@capttype}}%
2248 }}%
2249 \caption@IfCheckCommand{}{%
2250 \caption@Info{%
2251 Incompatible package detected (regarding \string\caption).\MessageBreak

```

```

2252     \string\caption\space=\space\meaning\caption}%
2253     \caption@setbool{incompatible}{1}}%
2254 \caption@CheckCommand\@caption{%
2255   % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2256   \long\def\@caption#1[#2]#3{%
2257     \par
2258     \addcontentsline{\csname ext@#1\endcsname}{#1}%
2259     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
2260     \begingroup
2261     \@parboxrestore
2262     \if@minipage
2263       \setminipage
2264     \fi
2265     \normalsize
2266     \makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2267     \endgroup}}%
2268 \caption@CheckCommand\@caption{%
2269   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
2270   \long\def\@caption#1[#2]#3{% second argument ignored
2271     \par\nobreak
2272     \begingroup
2273     \@parboxrestore
2274     \if@minipage
2275       \setminipage
2276     \fi
2277     \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
2278     \endgroup}}%
2279 % \caption@CheckCommand\float@caption{%
2280 %   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2281 %   \long\def\float@caption#1[#2]#3{%
2282 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%
2283 %     {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
2284 %     \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
2285 %     \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
2286 %     \@ifnextchar[{\float@ccon}{\egroup}}%
2287 %     \long\def\float@ccon[#1]{#1\par\egroup}}%
2288 \caption@CheckCommand\@caption{%
2289   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2290   \long\def\@caption#1[#2]#3{%
2291     \hyper@makecurrent{\@capttype}%
2292     \def\@currentlabelname{#2}%
2293     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2294       \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
2295     }%
2296     \begingroup
2297     \@parboxrestore
2298     \if@minipage
2299       \setminipage
2300     \fi
2301     \normalsize
2302     \makecaption{\csname fnum@#1\endcsname}{%
2303       \ignorespaces
2304       \ifHy@nesting

```

```

2305         \hyper@@anchor{\@currentHref}{#3}%
2306     \else
2307         \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2308     \fi
2309 }%
2310 \par
2311 \endgroup
2312 }}%

2313 \caption@CheckCommand\@caption{%
2314 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2315 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2316 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2317 \long\def\@caption#1[#2]#3{%
2318     \expandafter\ifx\csname if@capstart\expandafter\endcsname
2319         \csname iftrue\endcsname
2320     \global\let\@currentHref\hc@currentHref
2321 \else
2322     \hyper@makecurrent{\@cuptype}%
2323 \fi
2324 \def\@currentlabelname{#2}%
2325 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2326     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2327 }%
2328 \begingroup
2329 \@parboxrestore
2330 \if@minipage
2331     \setminipage
2332 \fi
2333 \normalsize
2334 \expandafter\ifx\csname if@capstart\expandafter\endcsname
2335     \csname iftrue\endcsname
2336     \global\@capstartfalse
2337     \makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2338 \else
2339     \makecaption{\csname fnum@#1\endcsname}{%
2340         \ignorespaces
2341         \ifHy@nesting
2342             \hyper@@anchor{\@currentHref}{#3}%
2343         \else
2344             \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2345         \fi
2346     }%
2347 \fi
2348 \par
2349 \endgroup
2350 }}%

2351 \caption@CheckCommand\@caption{%
2352 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
2353 \long\def\@caption#1[#2]{%
2354     \def\@currentlabelname{#2}%
2355     \NR@{#1}[{#2}]%
2356 }}%

2357 \caption@CheckCommand\@caption{%

```



```

2358 % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
2359 \long\def\@caption#1[#2]#3{%
2360   \@ifundefined{if#1topcap}%
2361     {\subfig@oldcaption{#1}[{#2}]{#3}}%
2362     {\@nameuse{if#1topcap}%
2363       \@listsubcaptions{#1}%
2364       \subfig@oldcaption{#1}[{#2}]{#3}}%
2365   \else
2366     \subfig@oldcaption{#1}[{#2}]{#3}%
2367     \@listsubcaptions{#1}%
2368   \fi}}}%

2369 \caption@CheckCommand\@caption{%
2370 % subfig.sty [2005/06/28 ver: 1.3 subfig package]
2371 \def\@caption{\caption@}%
2372 % \long\def\caption@#1[#2]#3{%
2373 %   \@ifundefined{caption@setfloattype}%
2374 %     \caption@settype
2375 %     \caption@setfloattype
2376 %     \@capttype
2377 %   \sf@ifpositiontop{%
2378 %     \@listsubcaptions{#1}%
2379 %     \sf@old@caption{#1}[{#2}]{#3}%
2380 %   }{%
2381 %     \sf@old@caption{#1}[{#2}]{#3}%
2382 %     \@listsubcaptions{#1}%
2383 %   }}%
2384 }%

2385 \caption@IfCheckCommand{}{%
2386 \caption@Info{%
2387   Incompatible package detected (regarding \string\@caption).\MessageBreak
2388   \string\@caption\space=\space\meaning\@caption}%
2389 \caption@setbool{incompatible}{1}}%

The option compatibility= will override the compatibility mode.
2390 \@ifundefined{caption@ifcompatibility}%
2391   {\let\caption@ifcompatibility\caption@ifincompatible
2392   \let\caption@tempa\caption@WarningNoLine}%
2393   {\let\caption@tempa\@gobble}% suppress warning
2394 \caption@ifcompatibility{%
2395   \caption@tempa{%
2396     \noexpand\caption will not be redefined since it's already\MessageBreak
2397     redefined by a document class or package which is\MessageBreak
2398     unknown to the caption package}%
2399   \renewcommand*\caption@redefine{}%

\ContinuedFloat is not supported in compatibility mode.
2400 \renewcommand*\caption@ContinuedFloat[1]{%
2401 \caption@Error{Not available in compatibility mode}}%

\caption@start is not supported in compatibility mode.
2402 \caption@AtBeginDocument*{%
2403 \let\caption@start\relax
2404 \@ifundefined{caption@ORI@capstart}}}%

```

```

2405     \caption@Debug{%
2406         Restore hypcap definition of \string\capstart\@gobble}%
2407     \let\capstart\caption@ORI@capstart}%
2408     \@ifundefined{caption@ORI@float@makebox}{}{%
2409         \caption@Debug{%
2410             Restore hyperref redefinition of \string\float@makebox\@gobble}%
2411         \let\float@makebox\caption@ORI@float@makebox}%
2412     }%

\caption@star We redefine \caption@star here so it does not make any harm.
2413     \renewcommand*\caption@star[2]{#1#2}%

2414 }{%
2415     \caption@ifincompatible{%
2416         \caption@WarningNoLine{%
2417             Forced redefinition of \noexpand\caption since the\MessageBreak
2418             unsupported(!) package option 'compatibility=false'\MessageBreak
2419             was given}%
2420     }{}%

\caption
\@caption 2421     \renewcommand*\caption@redefine{%
2422         \let\caption\caption@caption
2423         \let\@caption\caption@@caption}%
2424     \caption@redefine

2425 }%
2426 \caption@AtBeginDocument*{%
2427     \let\caption@ORI@capstart\@undefined
2428     \let\caption@ORI@float@makebox\@undefined}%

\@xfloat We redefine \@xfloat so inside floating environments our type-specific options will be
used, a hyperref anchor will be set etc.
2429 \let\caption@ORI@xfloat\@xfloat
2430 \def\@xfloat#1[#2]{%
2431     \caption@ORI@xfloat{#1}[#2]%
2432     \caption@settype{#1}}%

2433 }

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.
2434 \caption@AtBeginDocument{\caption@redefine}

\@makecaption
2435 \let\@makecaption\caption@makecaption

```

2.13 Support for sub-captions

```

\caption@DeclareSubType \caption@DeclareSub initializes the usage of \caption in sub-floats.
2436 \def\caption@DeclareSubType sub#1\@nil{%
2437 \caption@Debug{Initializing subtype for `#1'\@gobble}%
2438 \@namedef{caption@c@#1}{0}%
2439 \@namedef{caption@beginsub#1}{\caption@beginsubfloat{#1}}
2440 \@onlypreamble\caption@DeclareSubType

Initialize the sub-captions defined with \DeclareCaptionSubType...
2441 \caption@For*{subtypelist}{\caption@DeclareSubType sub#1\@nil}

Initialize the sub-captions defined with \newsubfloat[18]...
2442 \caption@AtBeginDocument*{%
2443 \ifundefined{sf@counterlist}{}{%
2444 \for{sf@temp}=\sf@counterlist\do{%
2445 \expandafter\caption@DeclareSubType\sf@temp\@nil}}}

\caption@subtypehook Hook, will be used inside \caption@setsubtype.
2446 \newcommand*\caption@subtypehook{%
2447 \ifx\caption\caption@subcaption \else
2448 \caption@ifrefstepcounter{}{%
2449 % no \caption or \subcaption in this (floating) environment yet
2450 \caption@Debug{Increment \@capttype\ counter =\the\value\@capttype}%
2451 \caption@l@stepcounter\@capttype
2452 \let\addcontentsline\caption@addsubcontentsline}%
2453 \ifnum\c@name caption@c@\@capttype\endcsname=\value\@capttype \else
2454 \caption@Debug{Reset sub\@capttype\ counter}%
2455 \expandafter\xdef\c@name caption@c@\@capttype\endcsname{%
2456 \the\value\@capttype}%
2457 \@stpel\@subcapttype
2458 \fi
2459 \c@ContinuedFloat=0\relax
2460 \let\caption@resetContinuedFloat\@gobble
2461 \let\caption@addcontentsline\caption@kernel@addcontentsline
2462 \let\caption@setfloatcapt\@firstofone
2463 \caption@clearmargin
2464 \caption@iflist{}{\let\caption@setlist\@gobble}%
2465 \caption@setoptions{sub}%
2466 \caption@setoptions{subfloat}% for subfig-package compatibility
2467 \let\caption\caption@subcaption
2468 \let\@makecaption\caption@makecaption
2469 \fi}%

\caption@subcaption Makes a sub-caption.
2470 \newcommand*\caption@subcaption{%
2471 \caption@iftype
2472 {\caption@checkgrouplevel{sub}\subcaption
2473 \caption@star
2474 {\caption@refstepcounter\@subcapttype}%
2475 {\caption@dblarg{\caption\@subcapttype}}}%
2476 {\caption@Error{\noexpand\subcaption outside float}}}

```

```

\caption@addcontentsline We extend \caption@addcontentsline so it handles sub-captions, too.
Note: \sf@ifpositiontop & \@listssubcaptions are defined by the subfigure & subfig
packages.

2477 \let\caption@kernel@addcontentsline\caption@addcontentsline
2478 \renewcommand*\caption@addcontentsline[2]{%
2479   \sf@ifpositiontop{\@listssubcaptions{#1}}{}}%
2480 \caption@kernel@addcontentsline{#1}{#2}%
2481 \sf@ifpositiontop{}{\@listssubcaptions{#1}}%
2482 \caption@addsubcontentslines{#1}}

2483 \newcommand*\caption@addsubcontentslines[1]{%
2484   \begingroup
2485     \caption@subcontentslines
2486   \endgroup
2487   \caption@clearsubcontentslines}%

2488 \caption@AtBeginDocument*{%
2489   \@ifundefined{sf@ifpositiontop}{\let\sf@ifpositiontop\@gobbletwo}{}}%
2490 \caption@clearsubcontentslines
2491 \g@addto@macro\caption@typehook{\caption@checksubcontentslines}%
2492 \AtEndDocument{\caption@checksubcontentslines}}%

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

2493 \newcommand*\caption@addsubcontentsline[3]{%
2494   \begingroup
2495   \let\label\@gobble \let\index\@gobble \let\glossary\@gobble
2496   \protected@edef\@tempa{\endgroup
2497     \noexpand\g@addto@macro\noexpand\caption@subcontentslines{%
2498       \noexpand\@namedef{the#2}{\csname the#2\endcsname}%
2499       \ifx\@currentHref\@undefined \else
2500         \noexpand\def\noexpand\@currentHref{\@currentHref}%
2501       \fi
2502       \protect\addcontentsline{#1}{#2}{#3}}}%
2503   \@tempa}

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

2504 \newcommand*\caption@checksubcontentslines{%
2505   \ifx\caption@subcontentslines\@empty \else
2506     \caption@Error{%
2507       Something's wrong--perhaps a missing \protect\caption\MessageBreak
2508       in the last figure or table}%
2509     \caption@clearsubcontentslines
2510   \fi}

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

2511 \newcommand*\caption@clearsubcontentslines{%
2512   \global\let\caption@subcontentslines\@empty}

```

2.14 Document class & Babel package support

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

```

2513 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}

```

2.14.2 The beamer class

```
2514 \@ifclassloaded{beamer}{%
2515   \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.

```
2516   \captionsetup{list=false}
2517   \DeclareCaptionOption{list}[1]{}
2518   \DeclareCaptionOption{listof}[1]{}
\figure
```

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

```
\table
2519   \expandafter\let\expandafter\caption@ORI@figure
2520   \csname\string\figure\endcsname
2521   \@namedef{\string\figure}[#1]{%
2522     \caption@ORI@figure[#1]%
2523     \caption@settype{figure}}
2524   \expandafter\let\expandafter\caption@ORI@table
2525   \csname\string\table\endcsname
2526   \@namedef{\string\table}[#1]{%
2527     \caption@ORI@table[#1]%
2528     \caption@settype{table}}
2529 }{}
```

2.14.3 The KOMA-Script classes

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

```
2530 \@ifundefined{scr@caption}{}{%
2531   \caption@AtBeginDocument{\let\scr@caption\caption}}
```

2.14.4 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)

```
2532 \@nameuse{caption@frenchb}\@nameundef{caption@frenchb}
```

2.14.5 The frenchle/pro package

```
2533 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
2534   \caption@Info{frenchle/pro package is loaded}%
2535   \let\captionfont@ORI\captionfont
2536   \let\captionlabelfont@ORI\captionlabelfont
2537   \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`.

```
2538   \@ifundefined{GOfrench}%
2539     {\let\caption@tempa\@firstofone}%
2540     {\def\caption@tempa{\g@addto@macro\GOfrench}}%
```

```

2541 \caption@tempa{%
2542     \let\captionfont\captionfont@ORI
2543     \let\captionfont@ORI\@undefined
2544     \let\captionlabelfont\captionlabelfont@ORI
2545     \let\captionlabelfont@ORI\@undefined
2546     \let\@makecaption\@makecaption@ORI
2547     \let\@makecaption@ORI\@undefined
\@cnORI We update the definition of \@cnORI so it actually reflects our definition of \caption.
2548     \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.
2549     \let\caption@tcORI\@tablescaption
2550     \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.
2551     \let\@eatDP\@undefined
2552     \let\caption@tempa\@empty
2553     \ifx\ffrench\ffrench\ffrench\ffrench
2554         \l@addto@macro\caption@tempa{\let\fnum@figure\ffrench}%
2555     \fi
2556     \ifx\tfrench\ffrench\ffrench\ffrench
2557         \l@addto@macro\caption@tempa{\let\fnum@table\ffrench}%
2558     \fi
2559     \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
2560     \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
2561     \caption@tempa
2562 }%
2563 }}

```

2.15 Package support

```

\caption@IfPackageLoaded \caption@IfPackageLoaded{<package>}[<version>]{<true>}{<false>}
Some kind of combination of \@ifpackageloaded and \@ifpackagelater. If
the <package> is not loaded yet, the check will be (re-)done \AtBeginDocument, so
the <package> could be loaded later on, too.
2564 \newcommand\caption@IfPackageLoaded[1]{%
2565     \@testopt{\caption@@IfPackageLoaded{#1}}{}}
2566 \@onlypreamble\caption@IfPackageLoaded
2567 \long\def\caption@@IfPackageLoaded#1[#2]#3#4{%
2568     \@ifpackageloaded{#1}\@firstofone{%
2569         \caption@Debug{#1 package is not loaded (yet)\@gobble}%
2570         \caption@AtBeginDocument}{%
2571             \caption@@ifpackageloaded{#1}[#2]{#3}{#4}}
2572 \@onlypreamble\caption@@IfPackageLoaded

```

```

2573 \newcommand\caption@ifpackageloaded[1]{%
2574   \@testopt{\caption@ifpackageloaded{#1}}{}}
2575 \@onlypreamble\caption@ifpackageloaded

2576 \long\def\caption@@ifpackageloaded#1[#2]{%
2577   \ifpackageloaded{#1}{%
2578     \caption@Info{#1 package is loaded}%
2579     \ifpackagelater{#1}{#2}\@firstoftwo{%
2580       \caption@Error{%
2581         For a successful cooperation we need at least version\MessageBreak
2582           `#2' of package #1,\MessageBreak
2583         but only version\MessageBreak
2584           '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
2585         is available}%
2586       \@secondoftwo}%
2587     }\@secondoftwo}}
2588 \@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.

```

2589 \newcommand*\caption@clearmargin{%
2590   \setcaptionmargin\z@
2591   \let\caption@minmargin\@undefined}

2592 \caption@setbool{needfreeze}{0}
2593 \caption@AtBeginDocument*{%
2594   \caption@ifneedfreeze{%

```

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

Used by the `fltpage & sidecap` package support.

```

2595 \newcommand*\caption@freeze{%
2596   \caption@teststar\caption@@freeze\@gobble\@firstofone}%

2597 \newcommand*\caption@@freeze[1]{%
2598   \global\let\caption@SCcontinued\relax
2599   \global\let\caption@SCsetup\@undefined
2600   \global\let\caption@SCentry\@undefined
2601   \global\let\caption@SCtext\@undefined
2602   \global\let\caption@SClabel\@undefined

2603   \let\caption@ORI@ContinuedFloat\ContinuedFloat
2604   \def\ContinuedFloat{%
2605     \caption@withoptargs\caption@SC@ContinuedFloat}%
2606   \def\caption@SC@ContinuedFloat##1{%
2607     \let\caption@ORI@setcounter\setcounter
2608     \let\caption@ORI@addtocounter\addtocounter
2609     \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
2610     \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
2611     \caption@ORI@ContinuedFloat##1%
2612     \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
2613     \let\setcounter\caption@ORI@setcounter
2614     \let\addtocounter\caption@ORI@addtocounter}%
2615   \let\caption@ORI@setup\captionsetup
2616   \def\captionsetup{%
2617     \caption@withoptargs\caption@SC@setup}%
2618   \def\caption@SC@setup##1##2{%

```

```

2619     \caption@g@addto@list\caption@SCsetup{##2}%
2620     \caption@ORI@setup##1{##2}}%
2621 \let\caption@ORI\caption
2622 \def\caption{%
2623     \def\caption{\caption@Error{%
2624         Only one \noexpand\caption can be placed in this environment}}%
2625     \let\captionsetup\caption@setup
2626     \let\caption@@refstepcounter\caption@l@stepcounter
2627     \caption@ORI}%
2628 \long\def\@caption##1[##2]##3{%
2629     \@bsphack
2630     \gdef\caption@SCentry{##2}%
2631     \gdef\caption@SCtext{##3}%
2632     \@esphack}%
2633 #1{% is \@gobble in star form, and \@firstofone otherwise
2634     \def\label##1{\@bsphack\gdef\caption@SClabel{##1}\@esphack}}%
2635 }%

\caption@defrost \caption@defrost
2636 \newcommand*\caption@defrost{%
2637     \ifx\caption@ORI@ContinuedFloat\@undefined
2638     \caption@defrost@setup
2639     \ifx\caption@SCtext\@undefined \else
2640     \expandafter\expandafter\expandafter\caption
2641     \expandafter\expandafter\expandafter[%
2642     \expandafter\expandafter\expandafter{%
2643     \expandafter\caption@SCentry\expandafter}\expandafter]%
2644     \expandafter{\caption@SCtext}%
2645     \fi
2646     \ifx\caption@SClabel\@undefined \else
2647     \expandafter\label\expandafter{\caption@SClabel}%
2648     \fi
2649     \else
2650     \caption@Error{Internal Error:\MessageBreak
2651     \noexpand\caption@defrost in same group as \string\caption@freeze}%
2652     \fi}%
2653 \newcommand*\caption@defrost@setup{%
2654     \caption@SCcontinued
2655     \ifx\caption@SCsetup\@undefined \else
2656     \expandafter\captionsetup\expandafter{\caption@SCsetup}%
2657     \fi}%
2658 }{}%
2659 \caption@undefbool{needfreeze}}

```

2.15.1 The float package

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

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

- The caption will be typeset inside a `\savebox` called `\@floatcapt` so it can be placed above or below the float contents afterwards.

- `\@makecaption` will not be used to finally typeset the caption. Instead `\@fs@capt` will be used which definition is part of the float style. (Note that `\@fs@capt` will not typeset any vertical space above or below the caption; instead this space will be typeset by the float style code itself.)

```
2660 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
2661 \@ifpackageloaded{floatrow}{%
2662 \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}}%
2663 }{%
```

`\@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>]`.

```
2664 \let\caption@ORI@float@setevery\@float@setevery
2665 \def\@float@setevery#1{%
2666 \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.

```
2667 \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.

```
2668 \renewcommand\caption@setfloatcapt[1]{%
2669 \let\@makecaption\caption@make
2670 \global\setbox\@floatcapt\vbox{%
2671 \color@begingroup ##1\color@endgroup}}%
```

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.

```
2672 \float@getstyle\float@style{#1}%
2673 \caption@setstyle*\float@style
2674 \caption@setoptions\float@style
2675 }{}}%
2676 \caption@freezeHref % will be defrosted in \@float@makebox
2677 \caption@ORI@float@setevery{#1}}%
```

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

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

But the float package doesn't follow this common naming convention: For floats defined with `\newfloat` it uses `\fname@<type>` instead, which breaks with our code (and with

`\autoref` and some other things as well). So we have to map the float package name to the common one here.

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

```
2678 \g@addto@macro\caption@typehook{%
2679   \expandafter\ifx\csname #1name\endcsname\relax
2680     \expandafter\let\csname #1name\expandafter\endcsname
2681       \csname fname@#1\endcsname
2682   \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.

```
2683 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2684 \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`).

```
2685 \providecommand*\float@ifstyle[1]{%
2686   \expandafter\ifx\csname fst@#1\endcsname\relax
2687     \expandafter\@secondoftwo
2688   \else
2689     \expandafter\@firstoftwo
2690   \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.

```
2691 \providecommand*\float@getstyle[2]{%
2692   \edef#1{%
2693     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2694     \expandafter\expandafter\expandafter\noexpand
2695     \csname fst@#2\endcsname}%
2696   \edef#1{#1}%
2697   \caption@Debug{floatstyle{#2} = '#1'}%}
```

`\float@setstyle` `\float@setstyle{⟨type⟩}{⟨style⟩}`

Sets or changes the float style associated with `⟨type⟩`.

```
2698 \providecommand*\float@setstyle[2]{%
2699   \expandafter\edef\csname fst@#1\endcsname{%
2700     \expandafter\noexpand\csname fs@#2\endcsname}}%
```

```

\float@dostyle \float@dostyle{<type>}
2701 \providecommand*\float@dostyle[1]{%
2702   \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.)

2703 \providecommand*\float@ifcaption[1]{%
2704   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2705   \expandafter\@firstoftwo
2706   \else
2707   \expandafter\@secondoftwo
2708   \fi}%

2709 }{%
2710 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2711 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2712 % \clearcaptionsetup{boxed}% used by the floatrow package?
2713 }

```

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

```
2714 \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’.

```

2715 \caption@ifbool{ruled}{%
2716   \captionsetup[ruled]{margin=0pt,minmargin=0,slc=0}%
2717 }{%
2718   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2719 }
2720 \caption@undefbool{ruled}

```

2.15.2 The floatflt package

```

2721 \caption@IfPackageLoaded{floatflt}[1996/02/27 v1.3]{%
\floatingfigure We patch \floatingfigure so \caption@floatflt will be used.
2722   \let\caption@ORI@floatingfigure\floatingfigure
2723   \def\floatingfigure{%
2724     \caption@floatflt{figure}%
2725     \caption@ORI@floatingfigure}%

\floatingtable Same with \floatingtable...
2726   \let\caption@ORI@floatingtable\floatingtable
2727   \def\floatingtable{%
2728     \caption@floatflt{table}%
2729   % \caption@setautoposition b%
2730     \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.

```

2731 \newcommand*\caption@floatflt[1]{%
2732   \caption@settype{#1}%
2733   \caption@clearmargin
2734   \caption@setoptions{floating#1}%
2735   \let\caption@boxrestore\@parboxrestore}%

2736 }{}

```

2.15.3 The fltpage package

```

2737 \caption@IfPackageLoaded{fltpage}[1998/10/29 v.0.3]{%
2738   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

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

2739 \renewcommand\FP@helpNote[2]{%
2740   \begingroup % save \caption@thepage
2741     \caption@pageref{#2}%
2742     \typeout{FP#1 is inserted on page \caption@thepage!}%
2743   \endgroup}%

```

`\FP@floatBegin` Original code:

```

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

2744 \renewcommand*\FP@floatBegin[1]{%
2745   \def\@cptype{#1}%
2746   \let\FP@LabelText\@empty
2747   \begin{lrbox}{\FP@floatCorpusBOX}%

```

```

2748 \caption@ifFPrefcap
2749 {\caption@freeze\relax}%
2750 {\def\label##1{\@bsphack\gdef\FP@LabelText{##1}\@esphack}%
2751 \caption@freeze*}%
2752 \ignorespaces}%

```

\FP@floatEnd Original code:

```

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

2753 \renewcommand*\FP@floatEnd{%
2754 \end{lrbox}%

2755 \stepcounter{FP@\@capttype C}%
2756 \caption@label\FP@positionLabel
2757 \FP@helpNote\@capttype\FP@positionLabel

2758 \edef\FP@RestoreCounter{%
2759 \noexpand\setcounter{\@capttype}{\the\value\@capttype}%
2760 \noexpand\setcounter{ContinuedFloat}{\the\value{ContinuedFloat}}}%

2761 \FP@float
2762 {\FP@positionLabel}% location label test
2763 {\begin{\@capttype}[p!]}%
2764 \usebox\FP@floatCorpusBOX
2765 \caption@defrost@setup
2766 \caption@ifFPlistcap
2767 {\caption@refstepcounter\@capttype

```

```

2768         \expandafter\caption@makecurrent\expandafter\@capttype
2769                                     \expandafter{\caption@SClentry}}%
2770         {\expandafter\captionlistentry\expandafter{\caption@SClentry}}%
2771         \caption@makeanchor\relax
2772         \ifx\FP@LabelText\@empty \else
2773         \expandafter\label\expandafter{\FP@LabelText}%
2774         \fi
2775     \end\@capttype}%
2776     {\FP@RestoreCounter
2777     \@ifundefined{theH\@capttype}}{%
2778     \expandafter\l@addto@macro\csname theH\@capttype\endcsname{.FP}}}%
2779     {\begin\@capttype[b!]}%
2780     \let\FP@savedSetfnumCommand\caption@setfnum
2781     \def\caption@setfnum##1{%
2782     \FP@savedSetfnumCommand{##1}%
2783     \ifx\FP@guide\@empty \else
2784     \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ {\FP@guide}}%
2785     \fi}%
2786     \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above captio
2787     \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below captio
2788     \caption@setoptions{FP\@capttype}%
2789     \FP@separatorCaption
2790     \caption@ifFPlistcap{}{\let\caption@addcontentsline\@gobbletwo}%
2791     \caption@defrost
2792     \end\@capttype}%
2793 }%

2794 \caption@For{typelist}{%
2795     \newcounter{FP@#1C}%
2796     \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2797 }{%
2798 \let\caption@ifFPlistcap\@undefined
2799 \let\caption@ifFPrefcap\@undefined
2800 }

```

2.15.4 The hyperref package

```

2801 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2802 \@ifundefined{hyper@makecurrent}}{% hyperref has stopped early
2803     \caption@WarningNoLine{%
2804         Hyperref support is turned off\MessageBreak
2805         because hyperref has stopped early}%
2806 }{%
2807     \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`.

```

2808     \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

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

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

```

2809     \renewcommand*\caption@makecurrent[2]{%
2810         \caption@makecurrentHref{#1}%

```

```

2811     \caption@Debug{hyperref current=\@currentHref}%
2812     \def\@currentlabelname{#2}}%
2813     \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

`\caption@makeanchor` We redefine `\caption@makeanchor` so a `hyperref` anchor will be set inside `\@caption`.
Note: Will be redefined by `\caption@start`.

```

2814     \renewcommand\caption@makeanchor[1]{%
2815     \caption@Debug{hyperref anchor: \@currentHref}%
2816     % If we cannot have nesting, the anchor is empty.
2817     \ifHy@nesting
2818     \hyper@@anchor{\@currentHref}{#1}%
2819     \else
2820     \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2821     \fi}%
2822     \g@addto@macro\caption@prepareslc{\let\caption@makeanchor\@firstofone}%

```

The hycap option

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

```

2823     \newif\if@capstart

```

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

```

2824     \def\caption@start{\caption@ifhycap\caption@start@relax}%
2825     \def\caption@start@{%

```

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

```

2826     \caption@makestart\@captype
2827     \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.)

```

2828     \global\@capstarttrue
2829     \let\hc@currentHref\@currentHref
2830     \def\caption@makecurrentHref##1{%
2831     \global\@capstartfalse
2832     \global\let\@currentHref\hc@currentHref}%

```

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

```

2833     \let\caption@makeanchor\@firstofone
2834     }%

```

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

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

-or-

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

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

The second idea has the advantage that it’s very easy to implement, but has some side-effects, e.g. the anchor names don’t follow the figure or table label names anymore.

Since I’m lazy I implemented the second idea, maybe I will revise this later on.

```
2835 \newcommand*\caption@makestart[1]{%
2836 \begingroup
2837 \Hy@hypertextnamesfalse
2838 % \gdef\@currentHlabel{}%
2839 \hyper@makecurrent{#1.caption}%
2840 \endgroup
2841 \caption@Debug{hycap start=\@currentHref}}%
```

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

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

```
2842 \newcommand*\caption@startanchor[1]{%
2843 \ifvmode\begingroup
2844 \caption@Debug{hycap anchor: #1 (vertical mode)}%
2845 \@tempdima\prevdepth
2846 \nointerlineskip
2847 \vspace*{-\caption@hycapSPACE}%
2848 \caption@anchor{#1}%
2849 \vspace*{\caption@hycapSPACE}%
2850 \prevdepth\@tempdima
2851 \endgroup\else
2852 \caption@Debug{hycap anchor: #1 (horizontal mode)}%
2853 \caption@anchor{#1}%
2854 \fi}%
```

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

```
2855 \newcommand*\caption@anchor[1]{%
2856 \ifmeasuring@ \else
2857 \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2858 \fi}%
```

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

```
2859 \ifx\HyperRaiseLinkLength\@tempdima
2860 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2861 \else
2862 \let\caption@raisedlink\Hy@raisedlink
2863 \fi
```


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

```

2864 \def\caption@@start{%
2865 \ifundefined{hc@currentHref}{%
2866 \caption@Warning{%
2867 The option 'hycap=true' will be ignored for this\MessageBreak
2868 particular \string\caption}}}%

```

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

```

2869 \def\caption@freezeHref{%
2870 \let\caption@ORI@start\caption@start
2871 \def\caption@start{\let\caption@start\caption@ORI@start}%
2872 % \let\caption@ORI@@start\caption@@start
2873 % \l@addto@macro\caption@subtypehook{%
2874 % \let\caption@@start\caption@ORI@@start}%
2875 \global\let\caption@currentHref\@undefined
2876 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2877 \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2878 \renewcommand*\caption@setfloatcapt{%
2879 \ifx\caption@currentHref\@undefined \else
2880 \let\caption@makeanchor\@firstofone
2881 \fi
2882 \caption@ORI@setfloatcapt}}%

```

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

```

2883 \def\caption@defrostHref{%
2884 \ifx\caption@currentHref\@undefined \else
2885 \caption@startanchor\caption@currentHref
2886 \global\let\caption@currentHref\@undefined
2887 \fi}%

```

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

```

2888 \ifundefined{HyOrg@float@makebox}{}{%
2889 \caption@Debug{%
2890 Redefining \noexpand\float@makebox (again)\@gobble}%
2891 \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
2892 \renewcommand\float@makebox[1]{%
2893 \HyOrg@float@makebox{#1}\relax \caption@defrostHref}}%
2894 }%
2895 }{}

```

2.15.5 The `hycap` package

```

2896 \caption@IfPackageLoaded{hycap}{% v1.0
2897 \ifx\caption@start\relax \else % hyperref hasn't stopped early

```

If the `hycap` package was loaded, we give up our own hyperlink placement algorithm and give the control over the placement to the `hycap` 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* `\caption`s in the same environment, at least unless a new `\capstart` will be placed.

```

2898 \let\caption@ORI@capstart\capstart % save for compatibility mode
2899 \@ifundefined{capstarttrue}% check for v1.10 of hypcap package
2900   {\def\capstart{\caption@start@}}%
2901   {\def\capstart{\ifcapstart\caption@start@\fi}}%
2902 \let\caption@start\relax
2903 \let\caption@@start\relax

```

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

```

2904 \caption@set@bool\caption@ifhypcap 1%
2905 \renewcommand*\caption@hypcapspace{\hypcapspace}%
2906 \fi}}

```

2.15.6 The listings package

```

2907 \caption@IfPackageLoaded{listings}[2004/02/13 v1.2]{%

```

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

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

```

2908 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2909 \def\lst@MakeCaption#1{% #1 is ‘t’ or ‘b’
2910   \begingroup

```

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

```

2911   \caption@setposition{#1}%
2912   \caption@iftop{%
2913     \@tempdima\belowcaptionskip
2914     \belowcaptionskip\abovecaptionskip
2915     \abovecaptionskip\@tempdima}}%

```

Workaround for issue with wrong skips (should be examined further)

```

2916   \caption@setup{rule=0}%

```

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

```

2917   \caption@setoptions{lstlisting}%

```

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

```

2918   \caption@setautoposition{#1}%

```

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

```

2919   \caption@begin{lstlisting}%
2920   \caption@ORI@lst@MakeCaption{#1}%
2921   \caption@end

```

```

2922     \endgroup}%

\lst@makecaption  Wrapper macros for typesetting the caption= resp. title= value.
\lst@maketitle   2923   \def\lst@makecaption{\caption@starfalse\@makecaption}%
                2924   \def\lst@maketitle{\caption@startrue\@makecaption\@empty}%

\ext@lstlisting  Since the listings package do not define \ext@lstlisting, but we needed it when
                \captionof{lstlisting} will be done by the end user, we define it here.
2925   \providecommand*\ext@lstlisting{lol}%

2926 }{}

```

2.15.7 The longtable package

```

\LTcapttype     \LTcapttype is preset to table.
2927 \providecommand*\LTcapttype{table}

2928 \caption@IfPackageLoaded{longtable}[1995/05/24 v3.14]{%
2929   \RequirePackage{ltcaption}[2007/09/01]%
2930   \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.

2931   \caption@AtBeginDocument{%
2932     \let\caption@ORI@LT@array\LT@array
2933     \renewcommand*\LT@array{%

\captionsetup for longtable:
2934     \global\let\caption@opt@@longtable\@undefined
2935     \def\captionsetup{%
2936       \noalign\bgroup
2937       \@ifstar\@captionsetup\@captionsetup}% gobble *
2938     \def\@captionsetup##1{\LT@captionsetup{##1}\egroup}%
2939     \def\LT@captionsetup##1{%
2940       \captionsetup@startrue\caption@setup@options[@longtable]{##1}%
2941       \global\let\caption@opt@@longtable\caption@opt@@longtable}%

\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2942     \def\@captionabovetrue{\LT@captionsetup{position=t}}%
2943     \def\@captionabovefalse{\LT@captionsetup{position=b}}%

\captionlistentry for longtable:
2944     \def\captionlistentry{%
2945       \noalign\bgroup
2946       \@ifstar{\egroup\LT@captionlistentry}% gobble *
2947       {\egroup\LT@captionlistentry}}%
2948     \def\LT@captionlistentry##1{%
2949       \caption@listentry\@firstoftwo[\LTcapttype]{##1}}%

```

`\ContinuedFloat` for longtable:

(Commented out, since it's not deeply tested and quite useless anyway)

Note: hyperref versions < v6.76j uses 2× `\hyper@makecurrent`

```
2950 % \caption@ifhyccap{%
2951 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2952 % \def\hyper@makecurrent##1{%
2953 % \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2954 % \caption@makestart{##1}%
2955 %% \let\Hy@LT@currentHlabel\@currentHlabel
2956 % \let\Hy@LT@currentHref\@currentHref
2957 % \def\hyper@makecurrent###1{%
2958 %% \let\@currentHlabel\Hy@LT@currentHlabel
2959 % \let\@currentHref\Hy@LT@currentHref}}%
2960 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
2961 % \def\ContinuedFloat{\noalign{%
2962 % \gdef\caption@setContinuedFloat{%
2963 % \let\caption@resetContinuedFloat\@gobble}%
2964 % \def\caption@setoptions###1{%
2965 % \g@addto@macro\caption@setContinuedFloat{%
2966 % \caption@setoptions{###1}}}%
2967 % \let\@capttype\LTcapttype
2968 % \caption@ORI@ContinuedFloat}}%
2969 % }{%
2970 % \def\ContinuedFloat{\noalign{%
2971 % \caption@Error{%
2972 % \noexpand\ContinuedFloat inside longtables\MessageBreak
2973 % is only available with 'hyccap=true'}}}%
2974 % }%
2975 % \global\let\caption@setContinuedFloat\@empty
2976 % \def\ContinuedFloat{\noalign{%
2977 % \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2978 % \caption@ORI@LT@array}}%
```

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

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

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

```
2979 \long\def\LT@c@ption#1[#2]#3{%
2980 \LT@makecaption#1{\csname fnun@LTcapttype\endcsname}{#3}%
2981 \LT@captionlistentry{#2}}%
```

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

The original definition:

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

```

% Based on article class "\@makecaption", "#1" is "\@gobble" in star
% form, and "\@firstofone" otherwise.
\sbbox\@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:

```

2982 \renewcommand\LT@makecaption[3]{%
2983   \caption@LT@make{%

```

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

```

2984   \caption@settype*\LTcaptype
2985   \ifdim\LTcapwidth=4in \else
2986     \setcaptionwidth\LTcapwidth
2987   \fi
2988   \caption@setoptions{longtable}%
2989 %   \caption@setContinuedFloat
2990   \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!

```

2991   \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%

```

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

```

2992   \caption@startrue#1\caption@starfalse
2993   \caption@resetContinuedFloat\LTcaptype
2994   \caption@begin\LTcaptype
2995   \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...)

```

2996   \vskip-\ht\strutbox

```

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

```

2997   \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2998   \caption@@make{#2}{#3}\endgraf
2999   \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
3000   \caption@end}}%

```

```

3001 }{}

```

2.15.8 The picinpar package

```
3002 \caption@ifpackageloaded{picinpar}{%
```

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

`\tabwindow`

```
3003   \long\def\figwindow[#1,#2,#3,#4] {%
3004     \caption@window{figure}%
3005     \caption@setoptions{figwindow}%
3006     \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
3007   \long\def\tabwindow[#1,#2,#3,#4] {%
3008     \caption@window{table}%
3009     \caption@setoptions{tabwindow}%
3010     \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.**

```
3011   \newcommand*\caption@window[1]{%
3012     \let\caption@boxrestore\@parboxrestore
3013     \let\@makecaption\caption@@make
3014     \caption@setautoposition b%
3015     \caption@settype{#1}%
3016     \caption@clearmargin}%
```

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

```
3017   \newcommand\caption@wincaption[1]{%
This will be done twice for every figwindow & tabwindow caption – on the first run
\picwd is Opt, on the second run \picwd is \hsize.
3018     \ifdim\picwd=\z@
3019       \let\caption@makecurrent\@gobbletwo
3020       \let\caption@@start\relax
3021       \caption@prepareslc
3022     \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.

```
3023     \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
3024     \if\@tempa*%
3025       \let\@tempa\@firstofone
3026     \else\if\@tempa[%]
3027       \let\@tempa\@firstofone
3028     \else
3029       \let\@tempa\@empty
3030     \fi\fi
3031     \expandafter\caption\@tempa{#1}}%
```

```
3032 }{ }
```

2.15.9 The picins package

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

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

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

```
3033 \newcommand*\piccaptiontype[1]{\def\@piccapttype{#1}}
```

```
3034 \caption@ifPackageLoaded{picins}{%
```

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

```
3035 \@ifundefined{@piccapttype}{%
```

```
3036 \caption@iftype{%
```

```
3037 \let\@piccapttype\@capttype
```

```
3038 }{%
```

```
3039 \def\@piccapttype{figure}%
```

```
3040 }{%
```

```
3041 }{}
```

```
3042 \let\@capttype\@undefined
```

`\piccaption` The original code:

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

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

```
3043 \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.

```
3044 \let\caption@ORI@make@piccaption\make@piccaption
```

```
3045 \def\make@piccaption{%
```

```
3046 \let\caption@ORI\caption
```

```

3047 \long\def\caption[##1]##2{%
3048 \caption@freezeHref % will be defrosted in \ivparpic
3049 \caption@settype\@piccaptiontype
3050 % \ifnum\c@piccaptionpos>2\relax
3051 \caption@clearmargin
3052 % \else
3053 % \captionwidth\z@ % do not use "width=" setting
3054 % \fi
3055 \caption@setoptions{parpic}%
3056 \let\caption@boxrestore\@parboxrestore
3057 \caption@setautoposition b%
3058 \expandafter\expandafter\expandafter\caption@ORI
3059 \expandafter\expandafter\expandafter[%
3060 \expandafter\expandafter\expandafter{%
3061 \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
3062 \caption@ORI@make@piccaption
3063 \let\caption\caption@ORI}%

```

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

```

3064 \let\caption@ORI@ivparpic\ivparpic
3065 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
3066 \let\caption@ORI@noindent\noindent
3067 \def\noindent{%
3068 \caption@defrostHref
3069 \let\noindent\caption@ORI@noindent
3070 \noindent}%
3071 \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
3072 \let\noindent\caption@ORI@noindent}%
3073 }{%
3074 \let\piccaptiontype\@undefined
3075 }

```

2.15.10 The rotating package

```
3076 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%
```

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

```

3077 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
3078 % \let\@rotcaption\@undefined

```

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

```

3079 \def\rotcaptionof{%
3080 \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

```


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

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

```
3081 \long\def\@makerotcaption#1#2{%
3082   \ifdim\captionwidth=\z@
3083     \setcaptionwidth{.8\textheight}%
3084     \caption@slc{#1}{#2}{.8\vszize}{%
3085       \let\caption@makerot\caption@@make
3086       \caption@clearmargin
3087 %     \long\def\caption@parbox##1##2{\hbox{\hszize=.8\textheight\relax##2}}%
3088 %     (not needed because \rotatebox uses an \hbox anyway)
3089     \let\caption@parbox\@secondoftwo}%
3090     \caption@set@bool\caption@ifslc0% been there, done that
3091   \fi
3092   \rotatebox{90}{\caption@makerot{#1}{#2}}%
3093   \nobreak\hspace{12pt}}%
3094 \newcommand\caption@makerot[2]{%
3095   \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
3096 \caption@For{typelist}{%
3097   \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
3098   \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
3099 }
```

2.15.11 The `sidecap` package

```
3100 \caption@IfPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
3101   \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.)

```
3102 \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`.

```
3103 \let\caption@ORI@SC@zfloat\SC@zfloat
3104 \def\SC@zfloat#1#2#3[#4]{%
```

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

```
3105 \let\caption@ORI\caption
3106 \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
3107 \let\caption\caption@ORI
```

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

```
3108 \caption@settype*{#2}%
3109 \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.

```
3110 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
3111 \def\endSC@FLOAT{%
```

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

```
3112 \let\caption@ORI@settype\caption@settype
3113 \def\caption@settype##1{% will be done in \@xfloat
3114 \caption@ORI@settype*{##1}% do not change \@currentlabel
3115 \caption@setSC@justify
3116 %%% \caption@setoptions{SCfloat}%
3117 \caption@setoptions{SC\@capttype}%
3118 \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

```
3119 \def\caption@setSC@justify{%
3120 \caption@clearmargin
3121 \@ifundefined{SC@justify}{}{%
3122 \ifx\SC@justify\@empty \else
3123 \let\caption@hj\SC@justify
3124 \let\SC@justify\@empty
3125 \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!

```
3126 \let\caption\SC@orig@caption
3127 \def\SC@orig@caption[##1]##2{\caption@defrost}%
```

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

```
3128 \caption@setSC@justify % for compatibility mode
3129 \caption@ORI@endSC@FLOAT}%
```

```

3130 \newcommand*\caption@For@SC[2]{%
3131   \def#1{b}% = \sidecaptionvpos{#2}{b} (v1.6)
3132   \newenvironment{SC#2}%
3133     {\SC@float[#1]{#2}}{\endSC@float}%
3134   \newenvironment{SC#2*}%
3135     {\SC@dblfloat[#1]{#2}}{\endSC@dblfloat}}%
3136 \@onlypreamble\caption@For@SC
3137 \caption@For{typelist}{%
3138   \expandafter\caption@For@SC\cscname SC@#1@vpos\endcscname{#1}}%
3139 }{}

```

2.15.12 The subfigure package

```

3140 \caption@IfPackageLoaded{subfigure}[2002/01/23 v2.1]{%

```

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

```

3141 \def\sf@ifpositiontop{%
3142   \ifx\@capttype\@undefined
3143     \expandafter\@gobbletwo
3144   \else\ifx\@capttype\relax
3145     \expandafter\expandafter\expandafter\@gobbletwo
3146   \else
3147     \expandafter\expandafter\expandafter\sf@if@position@top
3148   \fi\fi}
3149 \def\sf@if@position@top{%
3150   \@ifundefined{if\@capttype topcap}%
3151     {\@gobbletwo}%
3152     {\@nameuse{if\@capttype topcap}}%
3153     \expandafter\@firstoftwo
3154   \else
3155     \expandafter\@secondoftwo
3156   \fi}}
3157 }{}

```

2.15.13 The supertabular and xtab packages

```

3158 \caption@IfPackageLoaded{supertabular}[2002/07/19 v4.1e]{%

```

`\tablecaption` Make `\topcaption*` and `\bottomcaption*` work.

```

3159 \renewcommand*\tablecaption{%
3160   \caption@star
3161   {\refstepcounter{table}}%
3162   {\caption@dblarg{\@xtablecaption}}}%

```

`\@xtablecaption` Make `\nameref` and `\autoref` work.

```

3163 \let\caption@ORI@xtablecaption\@xtablecaption
3164 \long\def\@xtablecaption[#1]#2{%
3165   \def\@currentlabelname{#2}%
3166   \caption@ORI@xtablecaption[#1]{#2}}%

```

`\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}
```

```
3167 \long\def\ST@caption#1[#2]#3{\par%
3168 \caption@settype*{#1}%
3169 \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.

```
3170 \def\caption@fixposition{%
3171 \caption@setposition{\if@topcaption t\else b\fi}}%
3172 \caption@beginex{#1}{#2}{#3}%
3173 \caption@boxrestore
3174 \caption@normalsize
3175 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3176 \caption@end}%
```

```
3177 }{ }
```

```
3178 \caption@IfPackageLoaded{xtab}[2000/04/09 v2.3]{%
```

`\tablecaption` **Make \topcaption* and \bottomcaption* work.**

```
3179 \renewcommand*\tablecaption{%
3180 \caption@star
3181 {\refstepcounter{table}}%
3182 {\caption@dblarg{\@tablecaption}}}%
```

`\@xtablecaption` **Make \nameref and \autoref work.**

```
3183 \let\caption@ORI@xtablecaption\@xtablecaption
3184 \long\def\@xtablecaption[#1]#2{%
3185 \def\@currentlabelname{#2}%
3186 \caption@ORI@xtablecaption[#1]{#2}}%
```

`\ST@caption` **The original code:**

```
\long\def\ST@caption#1[#2]#3{\par%
\@initisotab
\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
        \global\advance\ST@pageleft -\PWSTcapht
        \ST@trace\tw@{Added caption. Space left for xtabular: \the\ST@pageleft}}

3187 \long\def\ST@caption#1[#2]#3{\par%
3188   \caption@settype*{#1}%
3189   \caption@setoptions{xtabular}%

3190   \def\caption@fixposition{%
3191     \caption@setposition{\if@topcaption t\else b\fi}}%

3192   \@initisotab
3193   \caption@beginex{#1}{#2}{#3}%
3194   \caption@boxrestore
3195   \caption@normalsize
3196   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3197   \caption@end
3198   \global\advance\ST@pageleft -\PWSTcapht
3199   \ST@trace\tw@{Added caption. Space left for xtabular: \the\ST@pageleft}}%

3200 }}

```

2.15.14 The threeparttable package

```

3201 \caption@IfPackageLoaded{threeparttable}[2003/06/13 v3.0]{%
\threeparttable Unfortunately \@capttype is not set when \TPT@common will be used, so we have to
redefine \threeparttable and \measuredfigure instead.

```

```

3202 \let\caption@ORI@threeparttable\threeparttable
3203 \renewcommand*\threeparttable{%
3204   \caption@settype{table}%
3205   \caption@setposition a% ?
3206   \caption@clearmargin
3207   \caption@setoptions{threeparttable}%
3208   \caption@ORI@threeparttable}%

```

\measuredfigure Same here...

```

3209 \let\caption@ORI@measuredfigure\measuredfigure
3210 \renewcommand*\measuredfigure{%
3211   \caption@settype{figure}%
3212   \caption@setposition a% ?
3213   \caption@clearmargin
3214   \caption@setoptions{measuredfigure}%
3215   \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}

```

```

3216 \def\TPT@caption#1[#2]#3{%
3217   \gdef\TPT@docapt{%
3218     \global\let\TPT@docapt\@undefined
3219     \caption@setautoposition\caption@TPT@position
3220     \TPT@LA@caption{#1}[{#2}]{#3}}%
3221   \ifx\TPT@hsize\@empty
3222     \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
3223     \gdef\caption@TPT@position{t}%
3224     \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
3225   \else
3226     \def\caption@TPT@position{b}%
3227     \TPT@docapt
3228   \fi
3229   \ignorespaces}%

3230 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
3231 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%

3232 }{}

```

2.15.15 The wrapfig package

```

3233 \caption@IfPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)

```

```

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

```

```

3234 \providecommand*\float@ifstyle[1]{%
3235   \expandafter\ifx\csname fst@#1\endcsname\relax
3236     \expandafter\@secondoftwo
3237   \else
3238     \expandafter\@firstoftwo
3239   \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.

```

3240 \newcommand*\caption@restylewrapfloat[1]{%
3241   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
3242     \csname wrap#1\endcsname
3243   \caption@Error{%
3244     For a successful cooperation of the 'wrapfig' package\MessageBreak
3245     with the 'float' package you should load the 'wrapfig'\MessageBreak
3246     package *after* the 'float' package}%
3247   \else
3248     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
3249       \csname wrap#1\endcsname
3250     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
3251     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
3252       \csname wrap#1\endcsname
3253   \fi}%

```

`\caption@wrapfloat`

```
3254 \newcommand*\caption@wrapfloat[1]{%
3255   \caption@settype*{#1}%
3256   \float@ifstyle{#1}{%
3257     \ifx\WF@floatstyhook\@undefined
3258       \caption@Error{%
3259         For a successful cooperation of the 'wrapfig' package\MessageBreak
3260         with the 'float' package you should use at least\MessageBreak
3261         'wrapfig' version 3.6}%
3262     \else
3263       \float@dostyle{#1}%
3264     \fi}{}%
3265   \caption@clearmargin
3266 %%% \caption@setoptions{wrapfloat}%
3267   \caption@setoptions{wrap#1}%
3268   \@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`.)

```
3269 \caption@restylewrapfloat{figure}%
3270 \caption@restylewrapfloat{table}%
3271 \caption@For{typelist}{%
3272   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
3273   \caption@restylewrapfloat{#1}}%
3274 \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.

```
3275 \@ifundefined{float@restyle}{}{%
3276   \toks@=\expandafter{\float@restyle{#1}}% (env may or may not be defined)
3277   \caption@restylewrapfloat{#1}%
3278   \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
3279   \@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`.

```
3280 \let\caption@ORI@wrapfloat\wrapfloat
3281 \def\wrapfloat#1{%
3282   \float@ifstyle{#1}{%
3283     \caption@Error{%
3284       For a successful cooperation of the 'wrapfig' package\MessageBreak
3285       with the 'float' package you should load the 'wrapfig'\MessageBreak
3286       package *right after* the 'float' package}}{%
3287     \caption@ORI@wrapfloat{#1}}%
3288   \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:

```
3289 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
3290   \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
3291   \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
3292   \caption@start
3293   \ifdim\hsize>\z@ \@parboxrestore \else
3294   \setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
3295   \ignorespaces \fi}%
3296 }{ }
```


References

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

- [14] Joachim Bleser and Edmund Lang:
PicIns-Benutzerhandbuch Version 3.0,
September 1992
- [15] Sebastian Rahtz and Leonor Barroca:
A style option for rotated objects in L^AT_EX,
1997/09/26
- [16] Rolf Niepraschk & Hubert Gäßlein:
The sidecap package,
2003/06/06
- [17] Steven D. Cochran:
The subfigure package,
2002/07/02
- [18] Steven D. Cochran:
The subfig package,
2005/07/05
- [19] Johannes Braams and Theo Jurriens:
The supertabular environment,
2002/07/19
- [20] Donald Arseneau:
Three part tables: title, tabular environment, notes,
2003/06/13
- [21] Donald Arseneau:
WRAPFIG.STY ver 3.6,
2003/01/31
- [22] Peter Wilson:
The xtab package,
2004/05/24
- [23] Anne Brüggemann-Klein:
Einführung in die Dokumentverarbeitung,
B.G. Teubner, Stuttgart, 1989
- [24] Heiko Oberdiek:
The refcount package,
2006/02/20