

The Implementation of the caption package*

Axel Sommerfeldt
caption@sommerfee.de

2010/01/14

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 v3.1m, last revised 2010/01/09.

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	17
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	30
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	32
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	68
2.14	Document class & Babel package support	70
2.14.1	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	70
2.14.2	The beamer class	70
2.14.3	The KOMA-Script classes	71
2.14.4	The frenchb Babel option	71
2.14.5	The frenchle/pro package	71
2.15	Package support	72
2.15.1	The float package	74
2.15.2	The floatflt package	77
2.15.3	The ftpage package	78
2.15.4	The hyperref package	80
2.15.5	The hypcap package	83
2.15.6	The listings package	84
2.15.7	The longtable package	85
2.15.8	The picinpar package	88
2.15.9	The picins package	89
2.15.10	The rotating package	90
2.15.11	The sidecap package	91
2.15.12	The subfigure package	93
2.15.13	The supertabular and xtab packages	93
2.15.14	The threeparttable package	95
2.15.15	The wrapfig package	96

1 Kernel

1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2010/01/14 v3.1m 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\relax

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

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 \providecommand*\caption@ifkomaclass{%
389     \ifundefined{scr@caption}\@gobble\@firstofone}
390 \@onlypreamble\caption@ifkomaclass
391 \caption@ifkomaclass{%
392     \let\caption@KV@parindent\KV@caption@parindent
393     \DeclareCaptionOption{parindent}[]{%
394         \ifx,#1,%
395             \caption@Debug{Option 'parindent' ignored}%
396         \else
397             \caption@KV@parindent{#1}%
398         \fi}%
399     \let\caption@KV@parskip\KV@caption@parskip
400     \DeclareCaptionOption{parskip}[]{%
401         \ifx,#1,%
402             \caption@Debug{Option 'parskip' ignored}%
403         \else
404             \caption@KV@parskip{#1}%
405         \fi}%
406 }

```

1.7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
407 \newcommand*\DeclareCaptionStyle[1]{%
408     \@testopt{\caption@declarestyle{#1}}{}}
409 \@onlypreamble\DeclareCaptionStyle
410 \def\caption@declarestyle#1[#2]#3{%
411     \global\@namedef{caption@sls@#1}{#2}%
412     \global\@namedef{caption@sty@#1}{#3}}
413 \@onlypreamble\caption@declarestyle

```

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!)

```

414 \DeclareCaptionOption{style}{\caption@setstyle{#1}}
415 \DeclareCaptionOption{style*}{\caption@setstyle*{#1}}
416 \DeclareCaptionOption{singlelinecheck}[1]{\caption@set@bool\caption@ifslc{#1}}
417 \DeclareCaptionOption{slc}[1]{\KV@caption@singlelinecheck{#1}}

```

```

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

```

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

The starred version will give no error message if the given style is not defined.

```

418 \newcommand*\caption@setstyle{%
419   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

420 \newcommand*\caption@@setstyle[2]{%
421   \ifundefined{caption@sty@#2}%
422     {#1{\caption@Error{Undefined style `#2'}}}%
423     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
424       \ifx\caption@setstyle@flag\@undefined
425         \let\caption@setstyle@flag\relax
426         \caption@resetstyle
427         \caption@xsetup\caption@sty
428         \let\caption@setstyle@flag\@undefined
429       \else
430         \caption@xsetup\caption@sty
431       \fi
432     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
433     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
434     \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!*

```

435 \newcommand*\caption@resetstyle{%
436   \caption@setup{%
437     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
438     justification=justified,font=,size=,labelfont=,textfont=,%
439     margin=0pt,minmargin=0,maxmargin=0,%
440     indent=0pt,parindent=0pt,hangindent=0pt,%
441     slc,rule,strut}%
442   \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 \LaTeX document classes (and was called ‘default’ in the caption package *v3.0*), the second one matches the document class actually used.

```

443 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
444 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
445   format=default,labelsep=default,textformat=default,%
446   justification=default,font=default,labelfont=default,textfont=default}

```


1.8 Formats

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

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

```
447 \newcommand*\DeclareCaptionFormat{%
448   \caption@teststar\caption@declareformat\@gobble\@firstofone}
449 \@onlypreamble\DeclareCaptionFormat

450 \newcommand*\caption@declareformat[2]{%
451   \@dblarg{\caption@@declareformat#1{#2}}
452 \@onlypreamble\caption@declareformat

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

458 \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`).

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

```
\DeclareCaptionDefaultFormat
```

```
465 \newcommand*\DeclareCaptionDefaultFormat[1]{%
466   \expandafter\def\expandafter\caption@fmt@default\expandafter
467   {\csname caption@fmt@#1\endcsname}%
468   \expandafter\def\expandafter\caption@slfmt@default\expandafter
469   {\csname caption@slfmt@#1\endcsname}%
470   \expandafter\def\expandafter\caption@ifh@default\expandafter
471   {\csname caption@ifh@#1\endcsname}}
472 \@onlypreamble\DeclareCaptionDefaultFormat
```

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

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

‘default’ usually maps to ‘plain’.

```
483 \DeclareCaptionDefaultFormat{plain}
```

1.9 Label formats

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

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

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

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

```
492 \DeclareCaptionLabelFormat{empty}{}
493 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
494 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace} (#2)}
495 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace}#2}
```

‘default’ usually maps to ‘simple’.

```
496 \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.
497 \newcommand\DeclareCaptionLabelSeparator{%
498   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
499 \@onlypreamble\DeclareCaptionLabelSeparator

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

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

510 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
511 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

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

```

516 \expandafter\let\expandafter\caption@iflfl\csname caption@iflfl@#1\endcsname
517 \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’.

```

518 \DeclareCaptionLabelSeparator{none}{}
519 \DeclareCaptionLabelSeparator{colon}{: }
520 \DeclareCaptionLabelSeparator{period}{. }
521 \DeclareCaptionLabelSeparator{space}{ }
522 \DeclareCaptionLabelSeparator*{quad}{\quad}
523 \DeclareCaptionLabelSeparator*{newline}{\}
524 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

```

`\caption@setdefaultlabelsep`

```

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

532 \newcommand*\caption@set@default@labelsep[1]{%
533 \def\caption@lsep@default{\@nameuse{caption@lsep@#1}}%
534 \def\caption@iflfl@default{\@nameuse{caption@iflfl@#1}}}}

```

‘default’ usually maps to ‘colon’.

```

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

```

1.11 Text formats

`\DeclareCaptionTextFormat`

```

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

539 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
540 \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`).

```

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

```

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

```

545 \DeclareCaptionTextFormat{simple}{#1}
546 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

1.12 Fonts

```

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

DeclareCaptionDefaultFont \DeclareCaptionDefaultFont {<name>} {<code>}
551 \newcommand*\DeclareCaptionDefaultFont [2] {%
552   \global\@namedef{caption#1@default}{#2}}
553 \@onlypreamble\DeclareCaptionDefaultFont

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

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

560 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
561 \DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
562 \DeclareCaptionDefaultFont{textfont}{}

\caption@setfont \caption@setfont {<name>} {<keyval-list of names>}
Selecting a caption font means saving all the code snippets in \caption{<name>.
563 \newcommand*\caption@setfont [1] {%
564   \expandafter\let\csname caption#1\endcsname\@empty
565   \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.
566 \newcommand*\caption@addtofont [2] {%
567   \begingroup
568     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
569     \define@key{caption@fnt}{default}[]{%
570       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
571     \caption@setkeys[caption]{caption@fnt}{#2}%
572     \global\let\caption@tempa\caption@fnt
573   \endgroup
574   \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.
575 \newcommand*\caption@font {%
576   \caption@teststar\caption@@font\@firstofone
577     {\caption@setkeys[caption]{caption@fnt}}
578 \newcommand*\caption@@font [2] {%
579   \begingroup
580   \def\caption@fnt{\endgroup}%
581   #1{#2}%
582   \caption@fnt}

```

These are the pre-defined font code snippets.

```

583 \DeclareCaptionFont{normalcolor}{\normalcolor}
584 \DeclareCaptionFont{color}{\color{#1}}

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

595 \DeclareCaptionFont{scriptsize}{\scriptsize}
596 \DeclareCaptionFont{footnotesize}{\footnotesize}
597 \DeclareCaptionFont{small}{\small}
598 \DeclareCaptionFont{normalsize}{\normalsize}
599 \DeclareCaptionFont{large}{\large}
600 \DeclareCaptionFont{Large}{\Large}

601 \DeclareCaptionFont{singlespacing}{%
602   \@ifundefined{setspace@singlespace}{}{%
603     \setstretch\setspace@singlespace}}% normally 1
604 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
605 \DeclareCaptionFont{doublespacing}{\doublespacing}
606 \DeclareCaptionFont{stretch}{\setstretch{#1}}

607 %\DeclareCaptionFont{normal}{%
608 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
609 \DeclareCaptionFont{normal}{%
610   \caption@font*{%
611     \KV@caption@fnt@normalcolor\@unused
612     \KV@caption@fnt@normalfont\@unused
613     \KV@caption@fnt@normalsize\@unused
614     \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.

```

615 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
616 \DeclareCaptionDefaultFont{size}{}

```

1.13 Justifications

```

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

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

```

```
625 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
626 \DeclareCaptionDefaultJustification{}
```

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

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

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

These are the pre-defined justification code snippets.

```
628 \DeclareCaptionJustification{justified}{}
629 \DeclareCaptionJustification{centering}{\centering}
630 \DeclareCaptionJustification{centerfirst}{\centerfirst}
631 \DeclareCaptionJustification{centerlast}{\centerlast}
632 \DeclareCaptionJustification{raggedleft}{\raggedleft}
633 \DeclareCaptionJustification{raggedright}{\raggedright}
```

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

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

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

```
645 \providecommand\centerlast{%
646   \let\\\@centercr
647   \leftskip\z@\@plus 1fil%
648   \rightskip\z@\@plus -1fil%
649   \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.

```
650 \DeclareCaptionJustification{Centering}{%
651   \caption@ragged\Centering\centering}
652 \DeclareCaptionJustification{RaggedLeft}{%
653   \caption@ragged\RaggedLeft\raggedleft}
654 \DeclareCaptionJustification{RaggedRight}{%
655   \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.

```

656 \newcommand*\caption@ragged{%
657   \caption@Debug{We need ragged2e}%
658   \protected@write\@auxout{}{\string\caption@newlabel{ragged2e}{}}%
659   \global\let\caption@ragged\caption@@ragged
660   \caption@ragged}

661 \caption@AtBeginDocument{%
662   \@ifundefined{caption@r@ragged2e}{%
663     \newcommand*\caption@@ragged{%
664       \caption@Warning{%
665         `ragged2e' support has been changed.\MessageBreak
666         Rerun to get captions right}%
667       \global\let\caption@ragged\@secondoftwo % suppress further warnings
668       \caption@ragged}%
669     }{%
670       \caption@Debug{We load ragged2e}%
671       \IfFileExists{ragged2e.sty}{%
672         \RequirePackage{ragged2e}%
673         \let\caption@@ragged\@firstoftwo
674       }{%
675         \newcommand*\caption@@ragged[2]{%
676           \@ifundefined{caption\string#1}{%
677             \caption@Warning{%
678               `ragged2e' package not loaded, therefore\MessageBreak
679               substituting \string#2 for \string#1\MessageBreak}%
680             \global\@namedef{caption\string#1}{}%
681             #2}%
682           }%
683         }}

```

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
684 \@ifundefined{abovecaptionskip}{%
685   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
686 \@ifundefined{belowcaptionskip}{%
687   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

688 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
689 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
690 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

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

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

```

691 \newcommand*\caption@rule{\caption@ifrule\caption@hrule\relax}
692 \newcommand*\caption@hrule{\hrule\@height\z@}

693 \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.

```
694 \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.*

```
695 \newcommand*\caption@setposition[1]{%
696   \caption@ifinlist{#1}{d,default}{%
697     \let\caption@position\caption@defaultpos
698   }{\caption@ifinlist{#1}{t,top,above}{%
699     \let\caption@position\@firstoftwo
700   }{\caption@ifinlist{#1}{b,bottom,below}{%
701     \let\caption@position\@secondoftwo
702   }{\caption@ifinlist{#1}{a,auto}{%
703     \let\caption@position\@undefined
704   }{%
705     \caption@Error{Undefined position `#1'}%
706   }}}}
```

```
\caption@defaultpos
```

The default ‘position’ is ‘auto’, this means that the `caption` package will try to guess the current position of the caption. (But in many cases, for example in `longtables`, this is doomed to fail!)

The setting ‘bottom’ corresponds to the `\@makecaption` implementation in the standard L^AT_EX document classes, but ‘auto’ should give better results in most cases.

```
707 %\caption@setdefaultpos{a}% default = auto
708 \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.)

```
709 \newcommand*\caption@iftop{%
710   \ifx\caption@position\@undefined
711     \let\caption@position\@secondoftwo
712 %   = \caption@setposition b%
713   \fi
714   \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.

```
715 \newcommand*\caption@fixposition{%
716   \ifx\caption@position\@undefined
717     \caption@autoposition
718   \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.

```

719 \newcommand*\caption@autoposition{%
720   \ifvmode
721     \edef\caption@tempa{\the\prevdepth}%
722     \caption@Debug{\protect\prevdepth=\caption@tempa}%
723     \ifdim\prevdepth>-\p@
724       \let\caption@position\@secondoftwo
725     \else
726       \let\caption@position\@firstoftwo
727     \fi
728 % = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
729 \else
730   \caption@Debug{no \protect\prevdepth}%
731   \let\caption@position\@secondoftwo
732 % = \caption@setposition b%
733 \fi}

```

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

```

734 \newcommand*\caption@setautoposition[1]{%
735   \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`.

```

736 \newcommand*\caption@beginhook{}
737 \newcommand*\caption@endhook{}
738 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
739 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

1.17 Lists

```

740 \DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
741 \DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}

```

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

```

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

```

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

```

743 \newcommand*\DeclareCaptionListFormat[2]{%
744   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
745 \@onlypreamble\DeclareCaptionListFormat

```

```

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

```

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

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

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

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

```
752 \DeclareCaptionListFormat{empty}{}
753 \DeclareCaptionListFormat{simple}{#1#2}
754 \DeclareCaptionListFormat{parens}{#1(#2)}
755 \DeclareCaptionListFormat{subsimple}{#2}
756 \DeclareCaptionListFormat{subparens}{(#2)}
```

```
\caption@setdefaultlistformat
```

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

‘default’ usually maps to ‘subsimple’.

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

1.18 Debug option

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

1.19 Document classes & Babel support

1.19.1 The standard L^AT_EX classes

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

```

784 \fi
785 \vskip\belowcaptionskip}}

```

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

```

786 \providecommand*\caption@ifamsclass{%
787 \ifundefined{@captionheadfont}\@gobble\@firstofone}
788 \@onlypreamble\caption@ifamsclass
789 \caption@ifamsclass{%
790 \caption@CheckCommand\@makecaption{%
791 % amsart|amsproc|amsbook [2004/08/06 v2.20]
792 \long\def\@makecaption#1#2{%
793 \setbox\@tempboxa\vbox{\color@setgroup
794 \advance\hsize-2\captionindent\noindent
795 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
796 {\@cdr#2\@nil}\.\@captionfont\upshape\enspace#2}%
797 \unskip\kern-2\captionindent\par
798 \global\setbox\@ne\lastbox\color@endgroup}%
799 \ifhbox\@ne % the normal case
800 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
801 \fi
802 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
803 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
804 \else % tempboxa contained more than one line
805 \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
806 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
807 \fi
808 \ifnum\@tempcnta<64 % if the float IS a figure...
809 \addvspace\abovecaptionskip
810 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
811 \else % if the float IS NOT a figure...
812 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
813 \nobreak
814 \vskip\belowcaptionskip
815 \fi
816 \relax
817 }}
818 \caption@CheckCommand\@makecaption{%
819 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
820 \long\def\@makecaption#1#2{%
821 \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
822 \setbox\@tempboxa\vbox{\color@setgroup
823 \advance\hsize-2\captionindent\noindent
824 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
825 {\@cdr#2\@nil}\@\addpunct{.}\@captionfont\upshape\enspace#2}%
826 \unskip\kern-2\captionindent\par
827 \global\setbox\@ne\lastbox\color@endgroup}%
828 \ifhbox\@ne % the normal case
829 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
830 \fi
831 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
832 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
833 \@tempdima\wd\@ne\advance\@tempdima-\captionindent
834 \wd\@ne\@tempdima

```

```

835     \else % tempboxa contained more than one line
836         \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
837             \unvbox\@tempboxa\parskip\z@skip
838             \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
839     \fi
840     \ifnum\@tempcnta<64 % if the float IS a figure...
841         \addvspace\abovecaptionskip
842         \noindent\kern\captionindent\box\@ne
843     \else % if the float IS NOT a figure...
844         \noindent\kern\captionindent\box\@ne
845         \nobreak
846         \vskip\belowcaptionskip
847     \fi
848     \relax
849 }}

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

863 }

```

1.19.3 The beamer class

```

864 \providecommand*\caption@ifbeamerclass{%
865     \@ifclassloaded{beamer}\@firstofone\@gobble}
866 \@onlypreamble\caption@ifbeamerclass

867 \caption@ifbeamerclass{%

868     \caption@CheckCommand\beamer@makecaption{%
869         % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
870         \long\def\beamer@makecaption#1#2{%
871             \def\insertcaptionname{\csname#1name\endcsname}%
872             \def\insertcaptionnumber{\csname the#1\endcsname}%
873             \def\insertcaption{#2}%
874             \nobreak\vskip\abovecaptionskip\nobreak
875             \sbox\@tempboxa{\usebeamertemplate**{caption}}%
876             \ifdim \wd\@tempboxa >\hsize
877                 \usebeamertemplate**{caption}\par
878             \else
879                 \global \@minipagefalse
880                 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
881             \fi
882             \nobreak\vskip\belowcaptionskip\nobreak}}

883 \DeclareCaptionLabelFormat{default}{#1}
884 \DeclareCaptionDefaultJustification{\raggedright}

```

```

885 \DeclareCaptionDefaultFont {font} {%
886   \usebeamerfont*{caption}%
887   \usebeamercolor [fg]{caption}}
888 \DeclareCaptionDefaultFont {labelfont} {%
889   \usebeamercolor [fg]{caption name}%
890   \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.)

```

891 \defbeamertemplate{caption}{caption3} {%
892   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
893 \DeclareOption{beamer} {%
894   % \usebeamerfont*{caption} will set font
895   \DeclareCaptionDefaultFont {font} {} %
896   \setbeamertemplate{caption}[caption3]}
897 }

```

1.19.4 The KOMA-Script classes

```

898 \providecommand*\caption@ifkomaclass{%
899   \@ifundefined{scr@caption}\@gobble\@firstofone}
900 \@onlypreamble\caption@ifkomaclass
901 \caption@ifkomaclass{%
902   \caption@CheckCommand\@makecaption{%
903     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
904     \long\def\@makecaption#1#2{%
905       \if@captionabove
906         \vskip\belowcaptionskip
907       \else
908         \vskip\abovecaptionskip
909       \fi
910       \@@makecaption\@firstofone{#1}{#2}%
911       \if@captionabove
912         \vskip\abovecaptionskip
913       \else
914         \vskip\belowcaptionskip
915       \fi}}
916 \DeclareCaptionFormat {default} [#1#2#3\par] {%
917   \ifdofullc@p
918     \caption@ifin@list\caption@lsepclist\caption@lsepname
919     {\caption@Error{%
920       The option ‘\labelsep=\caption@lsepname’ does not work\MessageBreak
921       with \noexpand\setcaphanging (which is set by default)}}%
922     {\caption@fmt@hang{#1}{#2}{#3}}%
923   \else
924     #1#2%
925     \ifdim\cap@indent<\z@
926       \par
927       \noindent\hspace*{-\cap@indent}%
928     \else\if@capbreak
929       \par
930     \fi\fi

```

```

931     #3\par
932     \fi}
933 \DeclareCaptionLabelSeparator{default}{\captionformat}
934 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
935 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
936 }

```

1.19.5 The NTG Dutch classes

```

937 \providecommand*\caption@ifntgclass{%
938   \@ifundefined{CaptionFonts}{\gobble}{\@firstofone}
939 \@onlypreamble\caption@ifntgclass
940 \caption@ifntgclass{%
941   \caption@CheckCommand\@makecaption{%
942     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
943     \long\def\@makecaption#1#2{%
944       \vskip\abovecaptionskip
945       \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
946       \ifdim \wd\@tempboxa >\hsize
947         {\CaptionLabelFont#1:} \CaptionTextFont#2\par
948       \else
949         \global \@minipagefalse
950         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
951       \fi
952       \vskip\belowcaptionskip}}
953 \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
954 \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
955 }

```

1.19.6 The thesis class

```

956 \providecommand*\caption@ifthesisclass{%
957   \@ifundefined{cph@font}{\gobble}{\@ifundefined{cpb@font}{\gobble}{\@firstofone}}
958 \caption@ifthesisclass{%
959   \caption@CheckCommand\@makecaption{%
960     % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
961     \long\def\@makecaption#1#2{%
962       \vskip\abovecaptionskip
963       \setbox\@tempboxa\hbox{\cph@font #1:} {\cpb@font #2}}%
964       \ifdim \wd\@tempboxa >\hsize
965         \@hangfrom{\cph@font #1:} {\cpb@font #2\par}%
966       \else
967         \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
968       \fi
969       \vskip\belowcaptionskip}}
970 \DeclareCaptionDefaultFormat{hang}
971 \DeclareCaptionDefaultFont{labelfont}{\cph@font}
972 \DeclareCaptionDefaultFont{textfont}{\cpb@font}
973 }

```

1.19.7 The french Babel option

```

974 \@ifundefined{FB@makecaption}{}{%

```

```

975 \caption@CheckCommand\@makecaption{%
976   % frenchb.1df [2005/02/06 v1.6g French support from the babel system]
977   % frenchb.1df [2007/10/05 v2.0e French support from the babel system]
978   \long\def\@makecaption#1#2{%
979     \vskip\abovecaptionskip
980     \sbox\@tempboxa{#1\CaptionSeparator #2}%
981     \ifdim \wd\@tempboxa >\hsize
982       #1\CaptionSeparator #2\par
983     \else
984       \global \@minipagefalse
985       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
986     \fi
987     \vskip\belowcaptionskip}}

988 \ifx\@makecaption\STD@makecaption
989   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
990   \def\caption@frenchb{% supress frenchb warning
991     \let\STD@makecaption\@makecaption
992     \let\FB@makecaption\@makecaption}
993 \else
994   \ifx\@makecaption\@undefined\else
995     \PackageInfo{caption}{%
996       The definition of \protect\@makecaption\space
997       has been changed,\MessageBreak
998       frenchb will NOT customize it}%
999   \fi
1000 \fi
1001 }

```

1.19.8 The frenchle/pro package

```

1002 \@ifundefined{frenchTeXmods}{}{%
1003   \caption@CheckCommand\@makecaption{%
1004     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
1005     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
1006     \def\@makecaption#1#2{%
1007       \ifFTY%
1008         \def\@secondofmany##1##2\void{##2}%
1009         \def\@tempa{\@secondofmany#2\void}%
1010         \ifx\@tempa\empty%
1011           \let\captionseparator\empty%
1012         \fi%
1013         \@mcORI{#1}{\relax\captionfont{#2}}%
1014       \else
1015         \@mcORI{#1}{#2}%
1016       \fi}}

1017 \caption@CheckCommand\@makecaption{%
1018   % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
1019   \def\@makecaption#1#2{%
1020     \ifFTY%
1021       \def\@secondofmany##1##2\void{##2}%
1022       \protected@edef\@tempa{\@secondofmany#2\void}%
1023       \ifx\@tempa\empty%
1024         \let\captionseparator\empty%

```

```

1025         \fi%
1026         \@mcORI{#1}{\relax\captionfont{#2}}%
1027     \else
1028         \@mcORI{#1}{#2}%
1029     \fi}}

1030 \DeclareCaptionDefaultFont{textfont}{\itshape}%
1031 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
1032 }

```

1.20 Execution of options

```

1033 \captionsetup{style=default,position=default,%
1034             list,listformat=default,twoside=\if@twoside 1\else 0\fi}

1035 \ProcessOptions*

1036 \caption@ifcheckcommand{%
1037     \caption@setbool{documentclass}{1}%
1038 }{%
1039     \caption@setbool{documentclass}{0}%
1040     \PackageInfo{caption}{%
1041         Unknown document class (or package),\MessageBreak
1042         standard defaults will be used}%
1043     \caption@debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1044 }

```

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.

1045 \newcommand*\caption@addcontentsline[2]{%
1046     \caption@iflist
1047     {\def\@tempa{#2}}%
1048     {\let\@tempa\@empty}%
1049     \ifx\@tempa\@empty \else
1050     {\let\\space
1051     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1052                     {\protect\numberline
1053                      {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1054                      {\ignorespaces #2}}}%
1055     \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.)
1056 \newif\ifcaption@star

\caption@fnum \caption@fnum{<float type>}
Typesets the caption label; as replacement for \fnum{<float type>}.
1057 \newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}

```



```

\caption@make \caption@make{<float name>}{<ref. number>}{<text>}
Typesets the caption.
1058 \newcommand\caption@make[2]{\caption@@make{\caption@lfmt{#1}{#2}}}

\caption@@make \caption@@make{<caption label>}{<caption text>}
1059 \newcommand\caption@@make[2]{%
1060 \begingroup
1061 \caption@stepcounter
1062 \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
1063 \ifx\caption@maxmargin\undefined \else
1064 \ifdim\captionmargin>\caption@maxmargin\relax
1065 \captionmargin\caption@maxmargin\relax
1066 \fi
1067 \fi
1068 \ifx\caption@minmargin\undefined \else
1069 \ifdim\captionmargin<\caption@minmargin\relax
1070 \captionmargin\caption@minmargin\relax
1071 \fi
1072 \fi

Special single-line treatment (option singlelinecheck=)
1073 \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%

Typeset the left margin (option margin=)
1074 \caption@calcmargin
1075 \@tempdima\captionmargin
1076 \ifdim\captionmargin@=\z@ \else
1077 \caption@ifoddpages{\advance\@tempdima\captionmargin@}%
1078 \fi
1079 \caption@ifh{\advance\@tempdima\caption@indent}%
1080 \hspace\@tempdima

We actually use a \vbox of width \captionwidth - \caption@indent to typeset the caption.
Note: \caption@indent is not supported if the caption format was defined with \DeclareCaptionFormat*.
1081 \@tempdima\captionwidth
1082 \caption@ifh{\advance\@tempdima-\caption@indent}%
1083 \caption@parbox\@tempdima{%

Typeset the indentation (option indentation=)
Bugfix 04-05-05: \hskip-\caption@indent replaced by \ifdim\caption@indent=\z@...
1084 \caption@ifh{%
1085 \ifdim\caption@indent=\z@
1086 \leavevmode
1087 \else
1088 \hskip-\caption@indent
1089 \fi}%

Typeset the caption itself and close the \caption@parbox
1090 \caption@@@make{#1}{#2}}%

Typeset the right margin (option margin=)
1091 \@tempdima\captionmargin
1092 \ifdim\captionmargin@=\z@ \else

```

```

1093     \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
1094     \fi
1095     \hspace\@tempdima
1096     \caption@endhook
1097     \endgroup
1098     \global\caption@starfalse}

```

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

```

1099 \newcommand*\caption@calcmargin{%
1100   \ifdim\captionwidth=\z@
1101     \captionwidth\linewidth
1102     \advance\captionwidth by -2\captionmargin
1103     \advance\captionwidth by -\captionmargin@
1104   \else
1105     \captionmargin\linewidth
1106     \advance\captionmargin by -\captionwidth
1107     \divide\captionmargin by 2
1108     \captionmargin@\z@
1109   \fi

1110 \caption@Debug{%
1111   \string\hsize=\the\hsize,
1112   \string\linewidth=\the\linewidth, \MessageBreak
1113   \string\leftmargin=\the\leftmargin,
1114   \string\rightmargin=\the\rightmargin, \MessageBreak
1115   \string\margin=\the\captionmargin,
1116   \string\margin@=\the\captionmargin@,
1117   \string\width=\the\captionwidth}%
1118 }

```

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

```

1119 \newcommand\caption@slc[4]{%
1120   \caption@Debug{Begin SLC}%
1121   \begingroup
1122   \caption@singleline
1123   \let\caption@hj\@empty
1124   \caption@calcmargin % calculate #3 if necessary
1125   \caption@prepareslc
1126   \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1127   \ifdim\wd\@tempboxa>#3%
1128     \endgroup
1129   \else
1130     \endgroup
1131     \caption@singleline
1132     #4%
1133   \fi
1134   \caption@Debug{End SLC}}

1135 \newcommand*\caption@singleline{%
1136   \caption@xsetup\caption@opt@singleline
1137   \let\caption@fmt\caption@slfmt}

```

```

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1138 \newcommand*\caption@prepareslc{%
1139 \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1140 \def\label{\caption@withoptargs\@gobbletwo}%
1141 \let\stepcounter\caption@l@stepcounter
1142 \let\refstepcounter\stepcounter\let\H\refstepcounter\stepcounter}
1143 \newcommand*\caption@l@stepcounter[1]{\advance\c@#1\endcsname\@ne\relax}

\caption@parbox \caption@parbox{\langle width \rangle}{\langle contents \rangle}
This macro defines the box which surrounds the caption paragraph.
1144 \newcommand*\caption@parbox{\parbox[b]}

\caption@@@make \caption@@@make{\langle caption label \rangle}{\langle caption text \rangle}
This one finally typesets the caption paragraph, without margin and indention.
1145 \newcommand\caption@@@make[2]{%
If the label is empty, we use no caption label separator.
1146 \sbox\@tempboxa{#1}%
1147 \ifdim\wd\@tempboxa=\z@
1148 \let\caption@lsep\relax
1149 % \@capbreakfalse
1150 \fi

If the text is empty, we use no caption label separator, too.
1151 \caption@ifempty{#2}{%
1152 \let\caption@lsep\relax
1153 % \@capbreakfalse
1154 % \let\caption@ifstrut\@secondoftwo
1155 }%

Take care that \caption@parindent and \caption@hangindent will be used to typeset the paragraph.
1156 \setpar{\@@par\caption@@par}\caption@@par

Finally typeset the caption.
1157 \caption@hj\captionfont\captionsize\caption@fmt
1158 {\ifcaption@star\else{\captionlabelfont#1}\fi}%
1159 {\ifcaption@star\else{\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1160 {\captiontextfont
1161 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1162 \nobreak\hskip\z@skip % enable hyphenation
1163 \caption@tfmt{#2}%
1164 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1165 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1166 \par}}

\caption@ifempty \caption@ifempty{\langle text \rangle}{\langle true \rangle} (no \langle false \rangle)
This one tests if the \langle text \rangle is actually empty.
Note: This will be done without expanding the text, therefore this is far away from being bullet-proof.
Note: This macro is re-defining itself so only the first test (in a group) will actually be done.
1167 \newcommand\caption@ifempty[1]{%
1168 \caption@ifempty{#1}%
1169 \caption@ifempty\@unused}

```

```

1170 \newcommand\caption@if@empty[1]{%
1171   \def\caption@tempa{#1}%
1172   \ifx\caption@tempa\@empty
1173     \let\caption@ifempty\@secondoftwo
1174   \else
1175     \expandafter\def\expandafter\caption@tempa\expandafter{%
1176       \caption@car#1\caption@if@empty\caption@nil}%
1177     \def\caption@tempb{\caption@if@empty}%
1178     \ifx\caption@tempa\caption@tempb
1179       \let\caption@ifempty\@secondoftwo
1180     \else
1181       \def\caption@tempb{\ignorespaces}%
1182       \ifx\caption@tempa\caption@tempb
1183         \expandafter\caption@if@empty\expandafter{\@gobble#1}%
1184       \else
1185         \def\caption@tempb{\label}%
1186         \ifx\caption@tempa\caption@tempb
1187           \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1188         \else
1189           \def\caption@tempb{\index}%
1190           \ifx\caption@tempa\caption@tempb
1191             \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1192           \else
1193             \def\caption@tempb{\glossary}%
1194             \ifx\caption@tempa\caption@tempb
1195               \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1196             \else
1197               \let\caption@ifempty\@gobbletwo
1198             \fi
1199           \fi
1200         \fi
1201       \fi
1202     \fi
1203 \fi}
1204 \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.

```

1205 \newcommand*\caption@@par{%
1206   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

1.23 Types & sub-types

`\DeclareCaptionType` `\DeclareCaptionType` [*options*] [*environment*] [*name*] [*list name*]

```

1207 \newcommand*\DeclareCaptionType{%
1208   \@testopt\@DeclareCaptionType{}%
1209   \@onlypreamble\DeclareCaptionType
1210 \def\@DeclareCaptionType[#1]#2{%
1211   \def\caption@type{#2}%
1212   \caption@Debug{New type `#2'}%
1213   \newcounter{#2}\@namedef{theH#2}{}%
1214   \KV@caption@DCT@within\caption@within@default
1215   \caption@DeclareWithinOption{#2}%

```

```

1216 \KV@caption@DCT@placement{tbp}%
1217 \@ifundefined{c@float@type}%
1218   {\newcounter{float@type}%
1219    \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1220   {}%
1221 \caption@Debug{float type `#2'=\the\value{float@type}}%
1222 \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1223 \addtocounter{float@type}{\value{float@type}}%
1224 \KV@caption@DCT@fileext{lo#2}%
1225 \@namedef{fnum#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1226 \newenvironment{#2}{\@float{#2}}{\end@float}%
1227 \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1228 \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1229 \@ifundefined{l@figure}%
1230   {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1231   {\expandafter\let\csname l@#2\endcsname\l@figure}%
1232 \expandafter\newcommand\csname #2name\endcsname{}%
1233 \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
1234 \uppercase\expandafter{\@tempa}%
1235 \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
1236 \@tempb
1237 \expandafter\let\csname #2name\endcsname\@tempa
1238 \expandafter\newcommand\csname list#2name\endcsname{}%
1239 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1240 \@cons\caption@typelist{{#2}}%
1241 \caption@setkeys[caption]{caption@DCT}{#1}%
1242 \@ifundefined{float@exts}{\newtoks\float@exts}{}%
1243 \let\float@do=\relax
1244 \edef\@tempa{\noexpand\float@exts{\the\float@exts\float@do{\@nameuse{ext@#2}}}}%
1245 \@tempa
1246 \@ifundefined{float@addtolists}{%
1247   \newcommand\float@addtolists[1]{%
1248     \def\float@do###1{\addtocontents{###1}{##1}}\the\float@exts}%
1249   \@ifundefined{@chapter}{\caption@PatchChapter}}{}%
1250 \@ifnextchar[\@@DeclareCaptionType\relax}
1251 \@onlypreamble\@@DeclareCaptionType
1252 \def\@@DeclareCaptionType[#1]{%
1253   \KV@caption@DCT@name{#1}%
1254   \@ifnextchar[\@@@DeclareCaptionType\relax}
1255 \@onlypreamble\@@@DeclareCaptionType
1256 \def\@@@DeclareCaptionType[#1]{%
1257   \KV@caption@DCT@listname{#1}}
1258 \@onlypreamble\@@@DeclareCaptionType
1259 \let\DeclareFloatingEnvironment\DeclareCaptionType % old command name
1260 \@onlypreamble\DeclareFloatingEnvironment

```

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

```

1261 \newcommand*\caption@within@default{\@ifundefined{c@chapter}{none}{chapter}}
1262 \@onlypreamble\caption@within@default

```

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

```

1263 \newcommand*\caption@listof[1]{%
1264   \begingroup
1265     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1266     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1267     \let\caption@ORI@starttoc\@starttoc
1268     \renewcommand*\@starttoc[1]{%
1269       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1270     \listoffigures
1271   \endgroup}

```

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

```

1272 \newcommand*\caption@typelist{}

```

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

```

1273 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
1274 \@onlypreamble@key{caption@DCT}{fileext}
1275 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1276 \@onlypreamble@key{caption@DCT}{listname}
1277 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1278 \@onlypreamble@key{caption@DCT}{name}
1279 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
1280 \@onlypreamble@key{caption@DCT}{placement}
1281 \define@key{caption@DCT}{within}{%
1282   \ifundefined{c@chapter}{\@removefromreset\caption@type{chapter}}%
1283   \@removefromreset\caption@type{section}%
1284   \begingroup
1285     \caption@setkeys[caption]{caption@within}{#1}%
1286   \endgroup}
1287 %\@onlypreamble@key{caption@DCT}{within}
1288 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1289 %\@onlypreamble@key{caption@DCT}{without}
1290 \define@key{caption@within}{none}[]{}%
1291 \caption@within{}{}
1292 %\@onlypreamble@key{caption@within}{none}
1293 \define@key{caption@within}{section}[]{}%
1294 \@addtoreset\caption@type{section}%
1295 \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
1296 %\@onlypreamble@key{caption@within}{section}
1297 \ifundefined{c@chapter}{\@addtoreset\caption@type{chapter}}%
1298 \define@key{caption@within}{chapter}[]{}%
1299 \@addtoreset\caption@type{chapter}%
1300 \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
1301 % \@onlypreamble@key{caption@within}{chapter}

```

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

```

1302 \newcommand*\caption@within{%
1303   \expandafter\caption@within@\expandafter{\caption@type}}
1304 %\@onlypreamble\caption@within
1305 \newcommand*\caption@within@[3]{%
1306   \global\@namedef{the#1}{#2\arabic{#1}}%
1307   \ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
1308     { \global\@namedef{theH#1}{#3\arabic{#1}} }
1309 %\@onlypreamble\caption@within@

```

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

```

1310 \providecommand*\@removefromreset[2]{{%
1311   \expandafter\let\csname c@#1\endcsname\@removefromreset
1312   \def\@elt##1{%
1313     \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1314     \else
1315       \noexpand\@elt{##1}%
1316     \fi}%
1317 \expandafter\xdef\csname cl@#2\endcsname{%
1318   \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`.)

```

1319 \newcommand*\caption@PatchChapter{%
1320   \providecommand*\@chapterlistsgap{10\p@}%

1321   % report.cls [2005/09/16 v1.4f Standard LaTeX document class]
1322   \caption@patch@chapter{report}{%
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     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1333     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1334     \if@twocolumn
1335       \@topnewpage[\@makechapterhead{##2}]%
1336     \else
1337       \@makechapterhead{##2}%
1338       \@afterheading
1339     \fi
1340   }{%
1341     \ifnum \c@secnumdepth >\m@ne
1342       \refstepcounter{chapter}%
1343       \typeout{\@chapapp\space\thechapter.}%
1344       \addcontentsline{toc}{chapter}%
1345         {\protect\numberline{\thechapter}##1}%
1346     \else
1347       \addcontentsline{toc}{chapter}{##1}%
1348     \fi
1349     \chaptermark{##1}%
1350     \ifdim \@chapterlistsgap>\z@
1351       \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1352       \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1353       \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1354     \fi
1355     \if@twocolumn
1356       \@topnewpage[\@makechapterhead{##2}]%
1357     \else

```

```

1358     \@makechapterhead{##2}%
1359     \@afterheading
1360   \fi}%

1361 % book.cls [2005/09/16 v1.4f Standard LaTeX document class]
1362 \caption@patch@chapter{book}{%
1363   \ifnum \c@secnumdepth >\m@ne
1364     \if@mainmatter
1365       \refstepcounter{chapter}%
1366       \typeout{\@chapapp\space\thechapter.}%
1367       \addcontentsline{toc}{chapter}%
1368         {\protect\numberline{\thechapter}##1}%
1369     \else
1370       \addcontentsline{toc}{chapter}{##1}%
1371     \fi
1372   \else
1373     \addcontentsline{toc}{chapter}{##1}%
1374   \fi
1375   \chaptermark{##1}%
1376   \addtocontents{lof}{\protect\advspace{10\p@}}%
1377   \addtocontents{lot}{\protect\advspace{10\p@}}%
1378   \if@twocolumn
1379     \@topnewpage[\@makechapterhead{##2}]%
1380   \else
1381     \@makechapterhead{##2}%
1382     \@afterheading
1383   \fi
1384 }{%
1385   \ifnum \c@secnumdepth >\m@ne
1386     \if@mainmatter
1387       \refstepcounter{chapter}%
1388       \typeout{\@chapapp\space\thechapter.}%
1389       \addcontentsline{toc}{chapter}%
1390         {\protect\numberline{\thechapter}##1}%
1391     \else
1392       \addcontentsline{toc}{chapter}{##1}%
1393     \fi
1394   \else
1395     \addcontentsline{toc}{chapter}{##1}%
1396   \fi
1397   \chaptermark{##1}%
1398   \ifdim \@chapterlistsgap>\z@
1399     \addtocontents{lof}{\protect\advspace{\@chapterlistsgap}}%
1400     \addtocontents{lot}{\protect\advspace{\@chapterlistsgap}}%
1401     \float@addtolists{\protect\advspace{\@chapterlistsgap}}%
1402   \fi
1403   \if@twocolumn
1404     \@topnewpage[\@makechapterhead{##2}]%
1405   \else
1406     \@makechapterhead{##2}%
1407     \@afterheading
1408   \fi}%

1409 % amsbook.cls [2004/08/06 v2.20]
1410 % smfbook.cls [1999/11/15 v1.2f Classe LaTeX pour les monographies editees par

```



```

1411 \caption@patch@chapter{ams/smfbook}{%
1412   \refstepcounter{chapter}%
1413   \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1414   \else \let\@secnumber\thechapter \fi
1415   \typeout{\chaptername\space\@secnumber}%
1416   \def\@toclevel{0}%
1417   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1418   \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1419   \chaptermark{##1}%
1420   \addtocontents{lof}{\protect\addvspace{10\p@}}%
1421   \addtocontents{lot}{\protect\addvspace{10\p@}}%
1422   \@makechapterhead{##2}\@afterheading
1423 }{%
1424   \refstepcounter{chapter}%
1425   \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1426   \else \let\@secnumber\thechapter \fi
1427   \typeout{\chaptername\space\@secnumber}%
1428   \def\@toclevel{0}%
1429   \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1430   \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1431   \chaptermark{##1}%
1432   \ifdim \@chapterlistsgap>\z@
1433     \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1434     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1435     \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1436   \fi
1437   \@makechapterhead{##2}\@afterheading}%

1438 % scrreprt/scrbook.cls
1439 \@ifundefined{KOMAClassName}{}{%
1440   \caption@Debug{document class '\KOMAClassName' detected}%
1441   \let\caption@patch@chapter\@gobblethree}%

1442 % rapport1/3.cls [2004/06/07 v2.1a NTG LaTeX document class]
1443 \caption@patch@chapter{rapport}{%
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   \addtocontents{lof}{\protect\addvspace{10\p@}}%
1454   \addtocontents{lot}{\protect\addvspace{10\p@}}%
1455   \if@twocolumn
1456     \@topnewpage[\@makechapterhead{##2}]%
1457   \else
1458     \@makechapterhead{##2}%
1459     \@afterheading
1460   \fi
1461 }{%
1462   \ifnum \c@secnumdepth >\m@ne
1463     \refstepcounter{chapter}%

```

```

1464     \typeout{\@chapapp\space\thechapter.}%
1465     \addcontentsline{toc}{chapter}%
1466         {\protect\numberline{\thechapter}\toc@font0 ##1}%
1467     \else
1468         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1469     \fi
1470     \chaptermark{##1}%
1471     \ifdim \@chapterlistsgap>\z@
1472         \addtocontents{lof}{\protect\advspace{\@chapterlistsgap}}%
1473         \addtocontents{lot}{\protect\advspace{\@chapterlistsgap}}%
1474         \float@addtolists{\protect\advspace{\@chapterlistsgap}}%
1475     \fi
1476     \if@twocolumn
1477         \@topnewpage[\@makechapterhead{##2}]%
1478     \else
1479         \@makechapterhead{##2}%
1480         \@afterheading
1481     \fi}%

1482 % boek(3).cls [2004/06/07 v2.1a NTG LaTeX document class]
1483 \caption@patch@chapter{boek}{%
1484     \ifnum \c@secnumdepth >\m@ne
1485         \if@mainmatter
1486             \refstepcounter{chapter}%
1487             \typeout{\@chapapp\space\thechapter.}%
1488             \addcontentsline{toc}{chapter}%
1489                 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1490         \else
1491             \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1492         \fi
1493     \else
1494         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1495     \fi
1496     \chaptermark{##1}%
1497     \addtocontents{lof}{\protect\advspace{10\p@}}%
1498     \addtocontents{lot}{\protect\advspace{10\p@}}%
1499     \if@twocolumn
1500         \@topnewpage[\@makechapterhead{##2}]%
1501     \else
1502         \@makechapterhead{##2}%
1503         \@afterheading
1504     \fi
1505 }{%
1506     \ifnum \c@secnumdepth >\m@ne
1507         \if@mainmatter
1508             \refstepcounter{chapter}%
1509             \typeout{\@chapapp\space\thechapter.}%
1510             \addcontentsline{toc}{chapter}%
1511                 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1512         \else
1513             \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1514         \fi
1515     \else
1516         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1517     \fi

```

```

1518 \chaptermark{##1}%
1519 \ifdim \@chapterlistsgap>\z@
1520 \addtocontents{lof}{\protect\advspace{\@chapterlistsgap}}%
1521 \addtocontents{lot}{\protect\advspace{\@chapterlistsgap}}%
1522 \float@addtolists{\protect\advspace{\@chapterlistsgap}}%
1523 \fi
1524 \if@twocolumn
1525 \@topnewpage[\@makechapterhead{##2}]%
1526 \else
1527 \@makechapterhead{##2}%
1528 \@afterheading
1529 \fi}%

1530 % thesis.cls [1996/25/01 1.0g LaTeX document class (wm).]
1531 \caption@patch@chapter{thesis}{%
1532 \ifnum \c@secnumdepth >\m@ne
1533 \if@mainmatter
1534 \refstepcounter{chapter}%
1535 \typeout{\chaptername\space\thechapter.}
1536 \if@thema
1537 \ifx\@shortauthor\@empty
1538 \addcontentsline{toc}{chapter}{%
1539 \protect\numberline{\thechapter.}##1}%
1540 \else
1541 \addcontentsline{toc}{chapter}{%
1542 \protect\numberline{\thechapter.}
1543 \@shortauthor\hfill\mbox{}}\vskip\normallineskip ##1}%
1544 \fi
1545 \else
1546 \addcontentsline{toc}{chapter}{%
1547 \protect\numberline{\thechapter.}##1}%
1548 \fi
1549 \else
1550 \addcontentsline{toc}{chapter}{##1}
1551 \fi
1552 \else
1553 \addcontentsline{toc}{chapter}{##1}
1554 \fi
1555 \chaptermark{##1}
1556 \addtocontents{lof}{\protect\advspace{10pt}}
1557 \addtocontents{lot}{\protect\advspace{10pt}}
1558 \if@twocolumn
1559 \@topnewpage[\@makechapterhead{##2}]
1560 \else
1561 \@makechapterhead{##2}
1562 \@afterheading
1563 \fi
1564 }{%
1565 \ifnum \c@secnumdepth >\m@ne
1566 \if@mainmatter
1567 \refstepcounter{chapter}%
1568 \typeout{\chaptername\space\thechapter.}%
1569 \if@thema
1570 \ifx\@shortauthor\@empty
1571 \addcontentsline{toc}{chapter}{%

```

```

1572         \protect\numberline{\thechapter.}##1}%
1573     \else
1574         \addcontentsline{toc}{chapter}{%
1575             \protect\numberline{\thechapter.}%
1576             \@shortauthor\hfill\mbox{} \vskip\normallineskip ##1}%
1577     \fi
1578     \else
1579         \addcontentsline{toc}{chapter}{%
1580             \protect\numberline{\thechapter.}##1}%
1581     \fi
1582     \else
1583         \addcontentsline{toc}{chapter}{##1}%
1584     \fi
1585     \else
1586         \addcontentsline{toc}{chapter}{##1}%
1587     \fi
1588     \chaptermark{##1}%
1589     \ifdim \@chapterlistsgap>\z@
1590         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1591         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1592         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1593     \fi
1594     \if@twocolumn
1595         \@topnewpage[\@makechapterhead{##2}]%
1596     \else
1597         \@makechapterhead{##2}%
1598         \@afterheading
1599     \fi}%
1600 \ifx\caption@patch@chapter\@gobblethree \else
1601     \caption@Debug{%
1602         Unsupported document class detected,\MessageBreak
1603         or \noexpand\@chapter was redefined by another package}%
1604 \fi
1605 \let\caption@PatchChapter\@undefined}
1606 %\@onlypreamble\caption@PatchChapter
1607 \newcommand\caption@patch@chapter[3]{%
1608     \begingroup
1609 %     \let\if@twocolumn\iffalse
1610     \let\if@mainmatter\iffalse
1611     \let\if@thema\iffalse
1612     \def\@tempa[##1]##2{#2}%
1613     \ifx\@tempa\@chapter
1614         \caption@Debug{document class `#1' detected}%
1615         \gdef\@chapter[##1]##2{#3}%
1616         \global\let\caption@patch@chapter\@gobblethree
1617     \fi
1618     \endgroup}
1619 %\@onlypreamble\caption@patch@chapter
1620 \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.)

```

1621 \newcommand*\caption@patch@stpelt{%

```

```

1622 \let\caption@stpelt\@stpelt
1623 \def\@stpelt##1{%
1624   \caption@stpelt{##1}%
1625   \begingroup
1626     \let\@elt\caption@stpelt
1627     \csname caption@cl@##1\endcsname
1628   \endgroup}%
1629 \let\caption@patch@stpelt\relax}
1630 \@onlypreamble\caption@patch@stpelt

```

\caption@addtoreset Like \@addtoreset from ltcounets.dtx

```

1631 \newcommand*\caption@addtoreset[2]{%
1632   \caption@patch@stpelt
1633   \@ifundefined{caption@cl@#2}{\@namedef{caption@cl@#2}}{}%
1634   \expandafter\@cons\csname caption@cl@#2\endcsname{##1}}
1635 \@onlypreamble\caption@addtoreset

```

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

```

1636 \newcommand*\caption@removefromreset[2]{%
1637   \begingroup
1638     \expandafter\let\csname c@#1\endcsname\caption@removefromreset
1639     \def\@elt##1{%
1640       \expandafter\ifx\csname c@##1\endcsname\caption@removefromreset
1641       \else
1642         \noexpand\@elt{##1}%
1643       \fi}%
1644     \expandafter\xdef\csname caption@cl@#2\endcsname{%
1645       \csname caption@cl@#2\endcsname}%
1646   \endgroup}
1647 \@onlypreamble\caption@removefromreset

```

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

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

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

```

1648 \newcommand*\DeclareCaptionSubType{%
1649   \caption@teststar\@DeclareCaptionSubType\@firstoftwo\@secondoftwo}
1650 \@onlypreamble\DeclareCaptionSubType

1651 \newcommand*\@DeclareCaptionSubType[1]{%
1652   \@testopt{\@@DeclareCaptionSubType{#1}}{alph}}
1653 \@onlypreamble\@DeclareCaptionSubType

1654 \def\@@DeclareCaptionSubType#1[#2]#3{%
1655   \@ifundefined{c@#3}%
1656     {\caption@Error{No float type '#3' defined}}%
1657     {\@ifundefined{c@sub#3}%
1658       {\caption@Debug{New subtype `sub#3'}}%
1659       \newcounter{sub#3}%
1660       \caption@addtoreset{sub#3}{#3}%
1661       \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1662       \@ifundefined{l@chapter}%
1663         {\edef\@tempa{\expandafter\expandafter\expandafter\noexpand
1664           \expandafter\@car\l@subsubsection\@nil}%

```

```

1665         \def\@tempb{\@dottedtocline}%
1666         \ifx\@tempa\@tempb % \l@subsubsection starts with \@dottedtocline
1667             \expandafter\edef\csname l@sub#3\endcsname{%
1668                 \noexpand\@dottedtocline{2}%
1669                 \expandafter\expandafter\expandafter\noexpand
1670                 \expandafter\@gobbletwo\l@subsubsection}%
1671         \else
1672             \@namedef{l@sub#3}{\@dottedtocline{2}{3.8em}{3.2em}}%
1673         \fi}%
1674         {\expandafter\let\csname l@sub#3\endcsname\l@subsection}%
1675         \@cons\caption@subtypelist{{#3}}}%
1676         {\caption@Debug{Modify caption `sub#3'}}}%
1677     \@namedef{sub#3name}{}%
1678     \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
1679     #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
1680     {\@namedef{p@sub#3}}}%
1681     \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1682     {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1683     \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1684     \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}%
1685     }}

```

```

1686 \@onlypreamble\@@DeclareCaptionSubType

```

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

```

1687 \newcommand*\caption@subtypelist{}

```

```

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

```

```

1688 \newcommand*\caption@For{\caption@withoptargs\caption@@For}

```

```

1689 %\@onlypreamble\caption@For

```

```

1690 \newcommand\caption@@For[3]{%
1691     \caption@AtBeginDocument#1{%
1692         \def\@elt##1{#3}%
1693         \@nameuse{caption@#2}%
1694         \let\@elt\relax}}%
1695 %\@onlypreamble\caption@@For

```

1.24 subfig package adaption

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

```

1696 \caption@AtBeginDocument{%
1697     \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1698     \ifx\subref\@tempa
1699         \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1700     \let\caption@setfloattype\@gobble
1701     \let\@dottedxxxline\sf@NEW@dottedxxxline
1702     \let\sf@subfloat\sf@NEW@subfloat

```

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

```

1703     \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%

```

This patch should provide better hyperref support:

```
1704 \ifpackageloaded{hyperref}{%
1705 \renewcommand*\sf@updatecaptionlist[4]{%
1706 \xdef\sf@captionlist{%
1707 \sf@captionlist,%
1708 {\protect\numberline{\@subcaplabel}\noexpand{\ignorespaces #2}}%
1709 {\@currentHref}}}%
1710 }{}%

1711 \fi
1712 \let\sf@NEW@dottedxxxline\@undefined
1713 \let\sf@NEW@subfloat\@undefined}

1714 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1715 \begingroup
1716 \caption@setfloattype{#1}%
1717 \caption@setoptions{subfloat}%
1718 \caption@setoptions{sub#1}%
1719 \ifnum #3>\@nameuse{c@#2depth}\else
1720 \@dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1721 \fi
1722 \endgroup}

1723 \def\sf@NEW@subfloat{%
1724 \begingroup
1725 \caption@setfloattype\@capttype
1726 \sf@ifpositiontop{%
1727 \maincaptiontoptrue
1728 }{%
1729 \maincaptiontopfalse
1730 }%
1731 \caption@setoptions{subfloat}%
1732 \caption@setoptions{sub\@capttype}%
1733 \let\sf@oldlabel=\label
1734 \let\label=\subfloat@label
1735 \ifmaincaptiontop\else
1736 \advance\@nameuse{c@\@capttype}\@ne
1737 \fi
1738 \refstepcounter{sub\@capttype}%
1739 \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1740 \@ifnextchar [% %] match left bracket
1741 {\sf@@subfloat}%
1742 {\sf@@subfloat[\@empty]}}
```

2 Main package

2.1 Identification

```
1743 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1744 \ProvidesPackage{caption}[2010/01/09 v3.1m Customizing captions (AR)]
1745 %\@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.

```
1746 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1747 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1748 \RequirePackage{caption3}[2008/08/24] % needs v3.1j or newer
```

2.3 Check against incompatible document classes

```
1749 \caption@ifbool{documentclass}{}{%
1750   \caption@WarningNoLine{%
1751     Unsupported document class (or package) detected,\MessageBreak
1752     usage of the caption package is not recommended}%
1753   \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1754 }
```

2.4 Check against incompatible packages

```
1755 \@ifpackageloaded{caption2}{}%
1756   \caption@Error{%
1757     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1758     the (current) caption package}%
1759   \endinput
1760 }{}

1761 \caption@AtBeginDocument{%
1762   \@ifpackageloaded{ftcap}{\caption@DisablePositionOption{ftcap}}{}%
1763   \@ifpackageloaded{nonfloat}{\caption@DisablePositionOption{nonfloat}}{}%
1764   \@ifpackageloaded{topcapt}{\caption@DisablePositionOption{topcapt}}{}%
\caption@DisablePositionOption
```

`\caption@DisablePositionOption{package}`

disables the ‘position’ option.

```
1765 \newcommand*\caption@DisablePositionOption[1]{%
1766   \caption@Info{%
1767     '#1' package detected; setting 'position=b' for compatibility reasons}%
1768   \caption@setposition b%

1769   \DeclareCaptionOption{position}{}%
1770   \caption@Error{Usage of the 'position' option is incompatible\MessageBreak
1771     to the '#1' package}}

1772 \@onlypreamble\caption@DisablePositionOption
```

2.5 Declaration of options

2.5.1 Options for figure and table


```

1773 \DeclareCaptionOption{figureposition}{%
1774   \captionsetup*[figure]{position=#1}}
1775 \@onlypreamble@key{caption}{figureposition}

1776 \DeclareCaptionOption{tableposition}{%
1777   \captionsetup*[table]{position=#1}}
1778 \@onlypreamble@key{caption}{tableposition}

1779 \DeclareCaptionOption{figurename}{\caption@SetName{figure}{#1}}
1780 \DeclareCaptionOption{tablename}{\caption@SetName{table}{#1}}
1781 \DeclareCaptionOption{name}{\caption@setname\@capttype{#1}}

1782 \DeclareCaptionOption{listfigurename}{\caption@SetName{listfigure}{#1}}
1783 \DeclareCaptionOption{listtablename}{\caption@SetName{listtable}{#1}}

\caption@SetName \caption@SetName{<cmd>}{<value>}
1784 \newcommand*\caption@SetName[2]{%
1785   \caption@setname{#1}{#2}%
1786   \begingroup
1787     \@ifundefined{langaugename}{}{%
1788       \@ifundefined{captions\langaugename}{}{%
1789         \expandafter\g@addto@macro\csname captions\langaugename\endcsname
1790           {\caption@setname{#1}{#2}}}%
1791   \endgroup}

1792 \newcommand*\caption@setname[2]{\@namedef{#1name}{#2}}

1793 \caption@AtBeginDocument{\let\caption@SetName\caption@setname}

ption@DeclareWithinOption
1794 \newcommand*\caption@DeclareWithinOption[1]{%
1795   \DeclareCaptionOption{#1within}{\caption@Within{#1}{##1}}%
1796   \DeclareCaptionOption{#1without}{\caption@Within{#1}{none}}}

1797 \caption@DeclareWithinOption{figure}
1798 \caption@DeclareWithinOption{table}

1799 \DeclareCaptionOption{within}{%
1800   \@ifundefined{c@figure}{}{\caption@Within{figure}{#1}}%
1801   \@ifundefined{c@table}{}{\caption@Within{table}{#1}}%
1802   \caption@For{typelist}{\caption@Within{##1}{#1}}%
1803   \def\caption@within@default{#1}}
1804 \DeclareCaptionOption{without}{\KV@caption@within{none}}

\caption@within
1805 \newcommand*\caption@Within[1]{\def\caption@type{#1}\KV@caption@DCT@within}

```

2.5.2 Miscellaneous options

```

1806 \DeclareCaptionOption*{config}[caption]{%
1807   \InputIfFileExists{#1.cfg}%
1808   {\typeout{*** Local configuration file #1.cfg used ***}}%
1809   {\caption@Warning{Configuration file #1.cfg not found}}}

1810 \DeclareCaptionOption{@minipage}{%
1811   \caption@ifinlist{#1}{auto,default}%
1812   {\let\caption@if@minipage\@gobbletwo}%
1813   {\caption@set@bool\caption@if@minipage{#1}}}

```

```
1814 \captionsetup{@minipage=default}
```

2.5.3 caption v1.x compatibility options

```
1815 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1816 \@onlypreamble@key{caption}{compatibility}

1817 \DeclareCaptionOptionNoValue*{normal}{%
1818   \caption@setformat{plain}%
1819   \caption@setjustification{justified}}
1820 \DeclareCaptionOptionNoValue*{isu}{%
1821   \caption@setformat{hang}%
1822   \caption@setjustification{justified}}
1823 \DeclareCaptionOptionNoValue*{hang}{%
1824   \caption@setformat{hang}%
1825   \caption@setjustification{justified}}
1826 \DeclareCaptionOptionNoValue*{center}{%
1827   \caption@setformat{plain}%
1828   \caption@setjustification{centering}}
1829 \DeclareCaptionOptionNoValue*{anne}{%
1830   \caption@setformat{plain}%
1831   \caption@setjustification{centerlast}}
1832 \DeclareCaptionOptionNoValue*{centerlast}{%
1833   \caption@setformat{plain}%
1834   \caption@setjustification{centerlast}}

1835 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1836 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1837 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1838 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1839 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1840 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1841 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1842 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1843 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1844 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1845 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1846 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1847 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1848 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1849 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1850 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}

1851 \caption@setbool{ruled}{0}
1852 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}
```

2.5.4 caption2 v2.x compatibility options

```
1853 \DeclareCaptionOptionNoValue*{flushleft}{%
1854   \caption@setformat{plain}%
1855   \caption@setjustification{raggedright}}
1856 \DeclareCaptionOptionNoValue*{flushright}{%
1857   \caption@setformat{plain}%
1858   \caption@setjustification{raggedleft}}

1859 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}

1860 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
```

```
1861 \caption@WarningNoLine{Obsolete option 'ignoreLTcapwidth' ignored}}
```

2.5.5 Obsolete caption v3.0 options

```
1862 \DeclareCaptionOption*{caption}{%
1863 \caption@setbool{temp}{#1}%
1864 \caption@ifbool{temp}{}{%
1865 \caption@Error{%
1866 The package option 'caption=#1' is obsolete.\MessageBreak
1867 Please pass this option to the subfig package instead\MessageBreak
1868 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.

```
1869 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1870 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1871 \@onlypreamble@key{caption}{FPlist}
1872 \@onlypreamble@key{caption}{FPref}

1873 \newcommand*\caption@setFPoption[2]{%
1874 \edef\caption@tempa{\@car#2\@nil}%
1875 \caption@setbool{FP#1cap}{\if c\caption@tempa 1\else 0\fi}}
1876 \@onlypreamble\caption@setFPoption

1877 \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).

```
1878 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1879 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}
1880 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

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

```
1881 \caption@ifamsclass{%
1882 \caption@Info{AMS or SMF document class}%
1883 \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1884 }
```

2.7 KOMA-Script document classes support

```
1885 \caption@ifkomaclass{%
1886 \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 1887 \g@addto@macro\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1888 \g@addto@macro\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}
```

```

1889 \if@tablecaptionabove
1890 \@tablecaptionabovetrue
1891 \else
1892 \@tablecaptionabovefalse
1893 \fi

\onelinecaptionstrue
\onelinecaptionsfalse 1894 \g@addto@macro\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1895 \g@addto@macro\onelinecaptionsfalse{\let\caption@ifslc\@secondoftwo}

1896 \ifonelinecaptions
1897 \onelinecaptionstrue
1898 \else
1899 \onelinecaptionsfalse
1900 \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.

1901 \g@addto@macro\@captionabovetrue{\let\caption@position\@firstoftwo}
1902 \g@addto@macro\@captionabovefalse{\let\caption@position\@secondoftwo}

1903 \DeclareCaptionOption{figureposition}{%
1904 \caption@WarningNoLine{Option 'figureposition=#1' has no effect\MessageBreak
1905 when used with a KOMA script document class}}
1906 \DeclareCaptionOption{tableposition}{%
1907 \caption@WarningNoLine{Option 'tableposition=#1' has no effect\MessageBreak
1908 when used with a KOMA script document class}}

\setcapindent

1909 \let\caption@KOMA@setcapindent\@setcapindent
1910 \renewcommand*\@setcapindent[1]{%
1911 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

1912 \let\caption@KOMA@@setcapindent\@@setcapindent
1913 \renewcommand*\@@setcapindent[1]{%
1914 \caption@KOMA@@setcapindent{#1}\caption@setcapindent}

1915 \newcommand*\caption@setcapindent{%
1916 \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}

1917 \@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.)

1918 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1919 \csname\string\setcapwidth\endcsname
1920 \@namedef{\string\setcapwidth}[#1]#2{%
1921 \caption@KOMA@setcapwidth[#1]#2}\caption@setcapwidth{#1}}

1922 \newcommand*\caption@setcapwidth[1]{%
1923 \ifx\#1\else
1924 \ifundefined{cap@margin}{}{%
1925 \def\@tempa{captionbeside}%
1926 \ifx\@tempa\@currenvir\else\caption@Warning{%
1927 Ignoring optional argument [#1] of \string\setcapwidth\MessageBreak}%

```

```

1928         \fi}%
1929     \fi
1930     \captionsetup{width=\cap@width}

1931 \def\caption@tempa{\hspace}%
1932 \ifx\caption@tempa\cap@width \else
1933     \caption@setcapwidth{?}
1934 \fi

\setcapmargin

1935 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1936         \csname\string\@setcapmargin\endcsname
1937 \@namedef{\string\@setcapmargin}[#1]#2{%
1938     \caption@KOMA@setcapmargin[#1]{#2}\caption@setcapmargin}

1939 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1940         \csname\string\@@setcapmargin\endcsname
1941 \@namedef{\string\@@setcapmargin}[#1]#2{%
1942     \caption@KOMA@@setcapmargin[#1]{#2}\caption@setcapmargin}

1943 \newcommand*\caption@setcapmargin{%
1944     \begingroup
1945     \let\onelinecaptionsfalse\relax
1946     \def\@twoside{0}%
1947     \def\if@twoside{\def\@twoside{1}\iffalse}%
1948     \cap@margin
1949     \def\@tempa{\endgroup}%
1950     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1951         \def\hspace##1##{\@firstofone}%
1952         \edef\@tempa{\endgroup
1953             \noexpand\captionsetup{%
1954                 twoside=\@twoside,slc=0,%
1955                 margin={\cap@left,\cap@right}}}%
1956     \fi\fi
1957     \@tempa}

1958 \ifx\cap@margin\relax \else
1959     \caption@setcapmargin
1960 \fi

1961 }

```

2.8 Processing of options

```
1962 \caption@ProcessOptions*{caption}
```

2.9 \captionof and \captionlistentry

```

1963 \caption@AtBeginDocument{%

1964     \DeclareCaptionOption{type}{\caption@settype{#1}}%
1965     \DeclareCaptionOption{type*}{\caption@settype*{#1}}%

1966     \DeclareCaptionOption{subtype}[sub\@capttype]{\caption@setsubtype{#1}}%
1967     \DeclareCaptionOption{subtype*}[sub\@capttype]{\caption@setsubtype*{#1}}%

1968 }

```

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

```

\caption@settype \caption@settype*{<type>}
sets \@capytype and executes the options associated with it (using \caption@setoptions).
Furthermore we check \currentgrouplevel (if avail), redefine \@currentlabel
so a \label before \caption will result in a hint instead of a wrong reference, and
use the macro \caption@(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).
1969 \newcommand*\caption@settype{%
1970 \caption@@settype{}}

1971 \newcommand*\caption@setsubtype{%
1972 \caption@iftype
1973 {\caption@@settype{sub}}%
1974 {\caption@Error{Option `subtype=' outside float}}}%

1975 \newcommand*\caption@@settype[1]{%
1976 \caption@teststar{\caption@@@settype{#1}}\@firstoftwo\@secondoftwo}

1977 \newcommand*\caption@@@settype[3]{%
1978 % #1 = "" or "sub"
1979 % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1980 % #3 = <type>, e.g. "figure" or "table"
1981 \@ifundefined{c@#3}%
1982 {\caption@Error{No float type '#3' defined}}%
1983 {\caption@Debug{#1type=#3}}%
1984 \caption@checkgrouplevel{#1}{%
1985 \captionsetup{#1type#2*\@empty=...}#2{ or
1986 \backslashchar#1captionof}}}%

1987 \edef\caption@tempa{#3}%
1988 \expandafter\ifx\cename @#1capytype\endcsname\caption@tempa \else
1989 \ifcaptionsetup@star\else\@nameuse{caption@#1type@warning}\fi
1990 \fi
1991 \expandafter\let\cename @#1capytype\endcsname\caption@tempa
1992 \@nameuse{caption@#1typehook}%

1993 \caption@setoptions{#3}%
1994 \ifx\caption@opt\relax
1995 \@nameundef{caption@#1type@warning}%
1996 \else
1997 \@namedef{caption@#1type@warning}{\caption@Warning{%
1998 The #1caption type was already set to
1999 '\cename @#1capytype\endcsname'\MessageBreak}}%
2000 \fi

2001 \let\caption@ifrefstepcounter\@secondoftwo
2002 #2{}{%
2003 \let\@currentlabel\caption@undefinedlabel
2004 % \let\@currentHlabel\@undefined
2005 \ifx\caption@ORI@label\@undefined
2006 \let\caption@ORI@label\label
2007 \let\label\caption@xlabel

```

```
2008     \fi
2009     \caption@start}}}
```

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

```
2010 \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.

```
2011 \newcommand*\caption@iftype{%
2012   \@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)

```
2013 \begingroup\expandafter\expandafter\expandafter\endgroup
2014 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
2015   \caption@Debug{TeX engine: TeX}
2016   \let\caption@checkgrouplevel\@gobbletwo
2017 \else
2018   \caption@Debug{TeX engine: e-TeX}
2019   \newcommand*\caption@checkgrouplevel[2]{%
2020     \@ifundefined{#1caption@grouplevel}{%
2021       \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}{}%
2022       \ifnum\currentgrouplevel>\caption@grouplevel\relax
2023         \expandafter\edef\csname #1caption@grouplevel\endcsname{%
2024           \the\currentgrouplevel}%
2025       \else
2026         \caption@Warning{\string#2\MessageBreak outside box or environment}%
2027       \fi
2028     }{}}
2029 \fi
```

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

```
2030 \newcommand*\caption@undefinedlabel{%
2031   \protect\caption@xref{\caption@labelname}{\on@line}}
2032 \DeclareRobustCommand*\caption@xref[2]{%
2033   \caption@WarningNoLine{\noexpand\label before \string\caption#2}%
2034   \@setref\relax\@undefined{#1}}
2035 \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.

```
2036 \newcommand*\caption@xlabel[1]{%
2037   \caption@@xlabel
2038   \def\caption@labelname{#1}%
2039   \caption@ORI@label{#1}}
2040 \newcommand*\caption@@xlabel{%
2041   \global\let\caption@@xlabel\@empty
2042   \@bsphack
2043   \protected@write\@auxout{}%
2044     {\string\providecommand*\string\caption@xref[2]{%
```

```

2045         \string\@setref\string\relax\string\@undefined{\string##1}}}%
2046     \@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)

2047 \caption@AtBeginDocument{%
2048     \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}
2049 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

\captionlistentry \captionlistentry[<float type>]{<list entry>}
\captionlistentry* [<float type>]{<list entry>}
2050 \newcommand*\captionlistentry{%
2051     \caption@teststar\@captionlistentry\@firstoftwo\@secondoftwo}

2052 \newcommand*\@captionlistentry[1]{%
2053     \@testopt{\caption@listentry{#1}}\@capttype}

2054 \def\caption@listentry#1[#2]#3{%
2055     \@bsphack
2056     #1{\caption@getttitle{#3}}%
2057     {\caption@refstepcounter{#2}%
2058     \caption@makecurrent{#2}{#3}}%
2059     \caption@addcontentsline{#2}{#3}%
2060     \@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.

```

2061 \def\ContinuedFloat{%
2062     \@ifnextchar[\@Continued@Float\@ContinuedFloat}

```



```

2063 \def \@Continued@Float[#1]{\addtocounter{#1}\m@ne}
2064 \def \@Continued@Float{%
2065   \caption@iftype
2066     {\addtocounter\@capttype\m@ne
2067     \caption@Continued@Float\@capttype}%
2068     {\caption@Error{\noexpand\Continued@Float outside float}}}
2069 \def \caption@Continued@Float#1{%
2070   \ifstar{\caption@Continued@Float@{#1}}{\caption@Continued@Float{#1}}}
2071 \def \caption@Continued@Float@{%
2072   \addtocounter\@capttype\@ne
2073   \@stpelt{Continued@Float}\stepcounter{Continued@Float}%
2074   \def \caption@resetContinued@Float#1{\xdef \caption@CFtype{##1}}%
2075   \caption@@Continued@Float}
2076 \def \caption@Continued@Float#1{%
2077   \edef \caption@tempa{#1}%
2078   \ifx \caption@tempa \caption@CFtype
2079     \stepcounter{Continued@Float}%
2080     \let \caption@resetContinued@Float \gobble
2081     \caption@@Continued@Float{#1}%
2082     \sf@Continued@Float{#1}%
2083   \else
2084     \caption@Error{Continued `#1' after \caption@CFtype}%
2085   \fi}
2086 \def \caption@@Continued@Float#1{%
2087   \expandafter\l@addto@macro\c@name the#1\endcsname\theContinued@Float
2088   \@ifundefined{theH#1}{}{%
2089     \expandafter\l@addto@macro\c@name theH#1\endcsname{%
2090       \@alph\c@Continued@Float}}%
2091   \caption@setoptions{Continued@Float}%
2092   \caption@setoptions{continued#1}}
2093 \providecommand*\sf@Continued@Float[1]{}
2094 \newcommand*\caption@CFtype{??}

```

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

```

2095 \newcounter{Continued@Float}
2096 \let \theContinued@Float \@empty

```

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

```

2097 \newcommand*\caption@resetContinued@Float[1]{%
2098   \@stpelt{Continued@Float}\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.

```

2099 \newcommand*\caption@refstepcounter[1]{%
2100   \caption@resetContinued@Float{#1}%

```

```

2101 \caption@@refstepcounter{#1}%
2102 \let\caption@ifrefstepcounter\@firstoftwo}

2103 \newcommand*\caption@@refstepcounter{\refstepcounter}
2104 \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[ ]{}).

2105 \@ifundefined{kernel@ifnextchar}%
2106 {\newcommand\caption@dblarg[1]{\@ifnextchar[{\#1}{\caption@xdblarg{\#1}}]}%
2107 {\newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{\#1}}]}%
2108 \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)

2109 \newcommand*\caption@begin[1]{%
2110 \begingroup
2111 \caption@setfnum{\#1}%
2112 \caption@fixposition
2113 \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>.

2114 \newcommand\caption@beginex[3]{%
2115 \caption@begin{\#1}%
2116 \caption@addcontentsline{\#1}{\#2}%
2117 \caption@ifempty{\#3}{}

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

2118 \newcommand*\caption@end{%
2119 \endgroup
2120 \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.

2121 \newcommand*\caption@setfnum[1]{%
2122 \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
2123 \@namedef{fnum@#1}{\caption@fnum{\#1}}%
2124 \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`.

```
2125 \newcommand*\caption@boxrestore{%
2126   \let\if@nobreak\iffalse
2127   \let\if@noskipsec\iffalse
2128   \let\par\@@par
2129 % \let-\@dischyph
2130 % \let'\@acci\let\'\@accii\let\=\@acciii
2131   \parindent\z@ \parskip\z@skip
2132   \everypar{}%
2133 % \linewidth\hsize
2134 % \@totalleftmargin\z@
2135   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
2136   \parfillskip\@flushglue \lineskip\normallineskip
2137   \baselineskip\normalbaselineskip
2138   \sloppy
2139   \let\\\@normalcr
2140 }
```

`\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).

```
2141 \newcommand*\caption@normalsize{%
2142   \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.**

```
2143 \let\caption@setfloatcapt\@firstofone
```

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

```
\caption@makeanchor 2144 \newcommand*\caption@makecurrent[2]{}
\caption@start      2145 \let\caption@makeanchor\@firstofone
\caption@@start
\caption@freezeHref
\caption@defrostHref
```

```

2146 \let\caption@start\relax
2147 \let\caption@@start\relax
2148 \let\caption@freezeHref\relax
2149 \let\caption@defrostHref\relax

```

`\caption@getttitle` This one is needed for support of the `nameref` package.

```

2150 \newcommand\caption@getttitle[1]{%
2151   \@ifundefined{NR@getttitle}%
2152     {\def\@currentlabelname{#1}}%
2153     {\NR@getttitle{#1}}}

```

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.

```

2154 \def\caption@caption{%
2155   \caption@iftype
2156     {\caption@checkgrouplevel\@empty\caption
2157      \caption@star
2158       {\caption@refstepcounter\@capttype}%
2159       {\caption@dblarg{\@caption\@capttype}}}%
2160     {\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.

```

2161 \newcommand*\caption@star[2]{%
2162   \@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](#).)

```

2163 \long\def\caption@@caption#1[#2]#3{%
2164   \ifcaption@star \else
2165     \caption@prepareanchor{#1}{#2}%
2166     \fi
2167   \par
2168   \caption@beginex{#1}{#2}{#3}%
2169   \caption@setfloatcapt{%
2170     \caption@boxrestore
2171     \if@minipage
2172       \@setminipage
2173     \fi
2174     \caption@normalsize
2175     \ifcaption@star
2176       \let\caption@makeanchor\@firstofone
2177     \fi

```

```

2178     \@makecaption{\csname fnum@#1\endcsname}%
2179             {\ignorespaces\caption@makeanchor{#3}}\par
2180     \caption@if@minipage\@minipagetrue\@minipagefalse}%
2181     \caption@end}%

```

`\caption@prepareanchor`

```

2182 \newcommand*\caption@prepareanchor[2]{%
2183   \caption@makecurrent{#1}{#2}%
2184   \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.

```

2185 \long\def\caption@makecaption#1#2{%
2186   \caption@iftop
2187     {\vskip\belowcaptionskip}%
2188     {\caption@rule\vskip\abovecaptionskip}%
2189   \caption@@make{#1}{#2}%
2190   \caption@iftop
2191     {\vskip\abovecaptionskip\caption@rule}%
2192     {\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.

```

2193 \newcommand*\caption@redefine{}
2194 \g@addto@macro\caption@redefine{%
2195   \caption@setbool{incompatible}{0}%
2196   \caption@CheckCommand\caption{%
2197     % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2198     \def\caption{%
2199       \ifx\@capttype\@undefined
2200         \@latex@error{\noexpand\caption outside float}\@ehd
2201         \expandafter\@gobble
2202       \else
2203         \refstepcounter\@capttype
2204         \expandafter\@firstofone
2205       \fi
2206       {\@dblarg{\@caption\@capttype}}%
2207     }}%
2208   \caption@CheckCommand\caption{%
2209     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
2210     \def\caption{
2211       \ifx\@capttype\@undefined
2212         \@latex@error{\noexpand\caption outside figure or table}\@ehd
2213         \expandafter\@gobble
2214       \else
2215         \refstepcounter\@capttype
2216         \expandafter\@firstofone

```

```

2217     \fi
2218     {\@dblarg{\@caption\@capttype}}%
2219   }}%

2220 \caption@CheckCommand\caption{%
2221   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2222   \renewcommand\caption{%
2223     \ifx\@capttype\@undefined
2224       \@latex@error{\noexpand\caption outside float}\@ehd
2225       \expandafter\@gobble
2226     \else
2227       \refstepcounter\@capttype
2228       \let\@tempf\@caption
2229       \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
2230         \expandafter\expandafter\let
2231           \expandafter\@tempf\csname @float@c@\@capttype\endcsname
2232       \fi
2233     \fi
2234     \@dblarg{\@tempf\@capttype}}}%

2235 \caption@CheckCommand\caption{%
2236   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2237   % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2238   % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2239   \def\caption{%
2240     \ifx\@capttype\@undefined
2241       \@latex@error{\noexpand\caption outside float}\@ehd
2242       \expandafter\@gobble
2243     \else
2244       \H@refstepcounter\@capttype
2245       \@ifundefined{fst@\@capttype}{%
2246         \let\Hy@tempa\@caption
2247       }{%
2248         \let\Hy@tempa\Hy@float@caption
2249       }%
2250       \expandafter\@firstofone
2251     \fi
2252     {\@dblarg{\Hy@tempa\@capttype}}%
2253   }}%

2254 \caption@CheckCommand\caption{%
2255   % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2256   \def\caption{%
2257     \ifx\@capttype\@undefined
2258       \@latex@error{\noexpand\caption outside float}\@ehd
2259       \expandafter\@gobble
2260     \else
2261       \H@refstepcounter\@capttype
2262       \let\Hy@tempa\@caption
2263       \@ifundefined{float@caption}{%
2264       }{%
2265         \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
2266           \let\Hy@tempa\Hy@float@caption
2267         \fi
2268       }%
2269       \expandafter\@firstofone

```

```

2270     \fi
2271     {\@dblarg{\Hy@tempa\@capttype}}%
2272   }}%

2273 \caption@ifcheckcommand{}{%
2274   \caption@info{%
2275     Incompatible package detected (regarding \string\caption).\MessageBreak
2276     \string\caption\space=\space\meaning\caption}%
2277   \caption@setbool{incompatible}{1}}%

2278 \caption@checkcommand\@caption{%
2279   % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2280   \long\def\@caption#1[#2]#3{%
2281     \par
2282     \addcontentsline{\csname ext@#1\endcsname}{#1}%
2283     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
2284     \begingroup
2285       \@parboxrestore
2286       \if@minipage
2287         \@setminipage
2288       \fi
2289       \normalsize
2290       \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2291     \endgroup}}%

2292 \caption@checkcommand\@caption{%
2293   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
2294   \long\def\@caption#1[#2]#3{% second argument ignored
2295     \par\nobreak
2296     \begingroup
2297       \@parboxrestore
2298       \if@minipage
2299         \@setminipage
2300       \fi
2301       \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
2302     \endgroup}}%

2303 % \caption@checkcommand\float@caption{%
2304 %   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2305 %   \long\def\float@caption#1[#2]#3{%
2306 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%
2307 %     {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
2308 %     \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
2309 %     \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
2310 %     \@ifnextchar[{\float@ccon}{\egroup}}%
2311 %   \long\def\float@ccon[#1]{#1\par\egroup}}%

2312 \caption@checkcommand\@caption{%
2313   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2314   \long\def\@caption#1[#2]#3{%
2315     \hyper@makecurrent{\@capttype}%
2316     \def\@currentlabelname{#2}%
2317     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2318       \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
2319     }%
2320   \begingroup
2321     \@parboxrestore

```

```

2322     \if@minipage
2323         \@setminipage
2324     \fi
2325     \normalsize
2326     \@makecaption{\csname fnum@#1\endcsname}{%
2327         \ignorespaces
2328         \ifHy@nesting
2329             \hyper@@anchor{\@currentHref}{#3}%
2330         \else
2331             \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2332         \fi
2333     }%
2334     \par
2335 \endgroup
2336 }}%

2337 \caption@CheckCommand\@caption{%
2338     % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2339     % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2340     % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2341     \long\def\@caption#1[#2]#3{%
2342         \expandafter\ifx\csname if@capstart\expandafter\endcsname
2343             \csname iftrue\endcsname
2344             \global\let\@currentHref\hc@currentHref
2345         \else
2346             \hyper@makecurrent{\@captype}%
2347         \fi
2348         \def\@currentlabelname{#2}%
2349         \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2350             \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2351         }%
2352     \begingroup
2353         \@parboxrestore
2354         \if@minipage
2355             \@setminipage
2356         \fi
2357         \normalsize
2358         \expandafter\ifx\csname if@capstart\expandafter\endcsname
2359             \csname iftrue\endcsname
2360             \global\@capstartfalse
2361             \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2362         \else
2363             \@makecaption{\csname fnum@#1\endcsname}{%
2364                 \ignorespaces
2365                 \ifHy@nesting
2366                     \hyper@@anchor{\@currentHref}{#3}%
2367                 \else
2368                     \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2369                 \fi
2370             }%
2371         \fi
2372     \par
2373 \endgroup
2374 }}%

```



```

2375 \caption@CheckCommand\@caption{%
2376 % hyperref.sty [2009/11/27 v6.79k Hypertext links for LaTeX]
2377 \long\def\@caption#1[#2]#3{%
2378   \expandafter\ifx\csname if@capstart\expandafter\endcsname
2379     \csname iftrue\endcsname
2380     \global\let\@currentHref\hc@currentHref
2381   \else
2382     \hyper@makecurrent{\@capytype}%
2383   \fi
2384   \def\@currentlabelname{#2}%
2385   \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2386     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2387   }%
2388   \begingroup
2389     \@parboxrestore
2390     \if@minipage
2391       \setminipage
2392     \fi
2393     \normalsize
2394     \expandafter\ifx\csname if@capstart\expandafter\endcsname
2395       \csname iftrue\endcsname
2396       \global\@capstartfalse
2397       \makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2398     \else
2399       \makecaption{\csname fnum@#1\endcsname}{%
2400         \ignorespaces
2401         \ifHy@nesting
2402           \expandafter\hyper@@anchor\expandafter{\@currentHref}{#3}%
2403         \else
2404           \Hy@raisedlink{%
2405             \expandafter\hyper@@anchor\expandafter{\@currentHref}{\relax}%
2406           }%
2407         #3%
2408       \fi
2409     }%
2410   \fi
2411   \par
2412 \endgroup
2413 }}%

2414 \caption@CheckCommand\@caption{%
2415 % hyperref.sty [2009/12/09 v6.79m Hypertext links for LaTeX]
2416 % hyperref.sty [2009/12/28 v6.79z Hypertext links for LaTeX]
2417 \long\def\@caption#1[#2]#3{%
2418   \expandafter\ifx\csname if@capstart\expandafter\endcsname
2419     \csname iftrue\endcsname
2420     \global\let\@currentHref\hc@currentHref
2421   \else
2422     \hyper@makecurrent{\@capytype}%
2423   \fi
2424   \ifundefined{NR@getttitle}{%
2425     \def\@currentlabelname{#2}%
2426   }{%
2427     \NR@getttitle{#2}%
2428   }%

```

```

2429 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2430 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2431 }%
2432 \begingroup
2433 \let\parboxrestore
2434 \if@minipage
2435 \let\setminipage
2436 \fi
2437 \normalsize
2438 \expandafter\ifx\csname if@capstart\endcsname
2439 \csname iftrue\endcsname
2440 \global\@capstartfalse
2441 \let\makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2442 \else
2443 \let\makecaption{\csname fnum@#1\endcsname}{%
2444 \ignorespaces
2445 \ifHy@nesting
2446 \expandafter\hyper@@anchor\expandafter{\@currentHref}{#3}%
2447 \else
2448 \Hy@raisedlink{%
2449 \expandafter\hyper@@anchor\expandafter{%
2450 \@currentHref
2451 }{\relax}%
2452 }%
2453 #3%
2454 \fi
2455 }%
2456 \fi
2457 \par
2458 \endgroup
2459 }}%

2460 \caption@CheckCommand\@caption{%
2461 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
2462 \long\def\@caption#1[#2]{%
2463 \def\@currentlabelname{#2}%
2464 \NR@@caption{#1}[{#2}]%
2465 }}%

2466 \caption@CheckCommand\@caption{%
2467 % nameref.sty [2009/11/27 v2.32 Cross-referencing by name of section]
2468 \long\def\@caption#1[#2]{%
2469 \NR@getttitle{#2}%
2470 \NR@@caption{#1}[{#2}]%
2471 }}%

2472 \caption@CheckCommand\@caption{%
2473 % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
2474 \long\def\@caption#1[#2]#3{%
2475 \@ifundefined{if#1topcap}%
2476 {\subfig@oldcaption{#1}[{#2}]{#3}}%
2477 {\@nameuse{if#1topcap}%
2478 \@listsubcaptions{#1}%
2479 \subfig@oldcaption{#1}[{#2}]{#3}%
2480 \else
2481 \subfig@oldcaption{#1}[{#2}]{#3}%

```

```

2482         \@listsubcaptions{#1}%
2483         \fi}}}%

2484 \caption@CheckCommand\@caption{%
2485 % subfig.sty [2005/06/28 ver: 1.3 subfig package]
2486 \def\@caption{\caption@}%
2487 % \long\def\caption@#1[#2]#3{%
2488 % \ifundefined{caption@setfloattype}%
2489 % \caption@settype
2490 % \caption@setfloattype
2491 % \@capttype
2492 % \sf@ifpositiontop{%
2493 % \@listsubcaptions{#1}%
2494 % \sf@old@caption{#1}[#{2}]{#3}%
2495 % }{%
2496 % \sf@old@caption{#1}[#{2}]{#3}%
2497 % \@listsubcaptions{#1}%
2498 % }}%
2499 %}%

2500 \caption@IfCheckCommand{}{%
2501 \caption@Info{%
2502 Incompatible package detected (regarding \string\@caption).\MessageBreak
2503 \string\@caption\space=\space\meaning\@caption}%
2504 \caption@setbool{incompatible}{1}}%

The option compatibility= will override the compatibility mode.

2505 \ifundefined{caption@ifcompatibility}%
2506 {\let\caption@ifcompatibility\caption@ifincompatible
2507 \let\caption@tempa\caption@WarningNoLine}%
2508 {\let\caption@tempa\@gobble}% suppress warning

2509 \caption@ifcompatibility{%
2510 \caption@tempa{%
2511 \noexpand\caption will not be redefined since it's already\MessageBreak
2512 redefined by a document class or package which is\MessageBreak
2513 unknown to the caption package}%
2514 \renewcommand*\caption@redefine{}}%

\ContinuedFloat is not supported in compatibility mode.

2515 \renewcommand*\caption@ContinuedFloat[1]{%
2516 \caption@Error{Not available in compatibility mode}}%

\caption@start is not supported in compatibility mode.

2517 \caption@AtBeginDocument*{%
2518 \let\caption@start\relax
2519 \ifundefined{caption@ORI@capstart}}{%
2520 \caption@Debug{%
2521 Restore hycap definition of \string\capstart\@gobble}%
2522 \let\capstart\caption@ORI@capstart}%
2523 \ifundefined{caption@ORI@float@makebox}}{%
2524 \caption@Debug{%
2525 Restore hyperref redefinition of \string\float@makebox\@gobble}%
2526 \let\float@makebox\caption@ORI@float@makebox}%
2527 }%

```

```

\caption@star We redefine \caption@star here so it does not make any harm.
2528   \renewcommand*\caption@star[2]{#1#2}%
2529   }{%
2530   \caption@ifincompatible{%
2531     \caption@WarningNoLine{%
2532       Forced redefinition of \noexpand\caption since the\MessageBreak
2533       unsupported(!) package option 'compatibility=false'\MessageBreak
2534       was given}%
2535   }}%

```

```

\caption
\@caption 2536   \renewcommand*\caption@redefine{%
2537     \let\caption\caption@caption
2538     \let\@caption\caption@@caption}%
2539   \caption@redefine
2540   }%
2541   \caption@AtBeginDocument*{%
2542     \let\caption@ORI@capstart\@undefined
2543     \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.**

```

2544   \let\caption@ORI@xfloat\@xfloat
2545   \def\@xfloat#1[#2]{%
2546     \caption@ORI@xfloat{#1}[#2]%
2547     \caption@settype{#1}}%
2548 }

```

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.

```
2549 \caption@AtBeginDocument{\caption@redefine}
```

`\@makecaption`

```
2550 \let\@makecaption\caption@makecaption
```

2.13 Support for sub-captions

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

```

2551 \def\caption@DeclareSubType sub#1\@nil{%
2552   \caption@Debug{Initializing subtype for `#1'\@gobble}%
2553   \@namedef{caption@c@#1}{0}%
2554   \@namedef{caption@beginsub#1}{\caption@beginsubfloat{#1}}
2555   \@onlypreamble\caption@DeclareSubType

```

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

```
2556 \caption@For*{subtypelist}{\caption@DeclareSubType sub#1\@nil}
```

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

```
2557 \caption@AtBeginDocument*{%
2558   \@ifundefined{sf@counterlist}{}{%
2559     \@for\sf@temp:=\sf@counterlist\do{%
2560       \expandafter\caption@DeclareSubType\sf@temp\@nil}}
```

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

```
2561 \newcommand*\caption@subtypehook{%
2562   \ifx\caption\caption@subcaption \else
2563     \caption@ifrefstepcounter{}{%
2564       % no \caption or \subcaption in this (floating) environment yet
2565       \caption@Debug{Increment \@c@type\ counter =\the\value\@c@type}%
2566       \caption@l@stepcounter\@c@type
2567       \let\addcontentsline\caption@addsubcontentsline}%
2568     \ifnum\csname caption@c@\@c@type\endcsname=\value\@c@type \else
2569       \caption@Debug{Reset sub\@c@type\ counter}%
2570       \expandafter\xdef\csname caption@c@\@c@type\endcsname{%
2571         \the\value\@c@type}%
2572       \@stpel\@subc@type
2573     \fi
2574     \c@ContinuedFloat=0\relax
2575     \let\caption@resetContinuedFloat\@gobble
2576     \let\caption@addcontentsline\caption@kernel@addcontentsline
2577     \let\caption@setfloatcapt\@firstofone
2578     \caption@clearmargin
2579     \caption@iflist{}{\let\caption@setlist\@gobble}%
2580     \caption@setoptions{sub}%
2581     \caption@setoptions{subfloat}% for subfig-package compatibility
2582     \let\caption\caption@subcaption
2583     \let\@makecaption\caption@makecaption
2584   \fi}%
```

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

```
2585 \newcommand*\caption@subcaption{%
2586   \caption@iftype
2587   {\caption@checkgrouplevel{sub}\subcaption
2588     \caption@star
2589     {\caption@refstepcounter\@subc@type}%
2590     {\caption@dblarg{\@caption\@subc@type}}}%
2591   {\caption@Error{\noexpand\subcaption outside float}}}
```

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

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

```
2592 \let\caption@kernel@addcontentsline\caption@addcontentsline
2593 \renewcommand*\caption@addcontentsline[2]{%
2594   \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
2595   \caption@kernel@addcontentsline{#1}{#2}%
2596   \sf@ifpositiontop{}{\@listsubcaptions{#1}}%
2597   \caption@addsubcontentslines{#1}}
```

```

2598 \newcommand*\caption@addsubcontentslines[1]{%
2599   \begingroup
2600     \caption@subcontentslines
2601   \endgroup
2602   \caption@clearsubcontentslines}%

2603 \caption@AtBeginDocument*{%
2604   \@ifundefined{sf@ifpositiontop}{\let\sf@ifpositiontop\@gobbletwo}{}%
2605   \caption@clearsubcontentslines
2606   \g@addto@macro\caption@typehook{\caption@checksubcontentslines}%
2607   \AtEndDocument{\caption@checksubcontentslines}}%

```

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

```

2608 \newcommand*\caption@addsubcontentsline[3]{%
2609   \begingroup
2610   \let\label\@gobble \let\index\@gobble \let\glossary\@gobble
2611   \protected@edef\@tempa{\endgroup
2612     \noexpand\g@addto@macro\noexpand\caption@subcontentslines{%
2613       \noexpand\@namedef{the#2}{\csname the#2\endcsname}%
2614       \ifx\@currentHref\@undefined \else
2615         \noexpand\def\noexpand\@currentHref{\@currentHref}%
2616       \fi
2617       \protect\addcontentsline{#1}{#2}{#3}}}%
2618   \@tempa}

```

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

```

2619 \newcommand*\caption@checksubcontentslines{%
2620   \ifx\caption@subcontentslines\@empty \else
2621     \caption@Error{%
2622       Something's wrong--perhaps a missing \protect\caption\MessageBreak
2623       in the last figure or table}%
2624     \caption@clearsubcontentslines
2625   \fi}

```

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

```

2626 \newcommand*\caption@clearsubcontentslines{%
2627   \global\let\caption@subcontentslines\@empty}

```

2.14 Document class & Babel package support

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

```

2628 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}

```

2.14.2 The beamer class

```

2629 \@ifclassloaded{beamer}{%
2630   \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.

```

2631   \captionsetup{list=false}
2632   \DeclareCaptionOption{list}[1]{}
2633   \DeclareCaptionOption{listof}[1]{}

```

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

```

\table
2634 \expandafter\let\expandafter\caption@ORI@figure
2635 \csname\string\figure\endcsname
2636 \@namedef{\string\figure} [#1] {%
2637 \caption@ORI@figure [#1] %
2638 \caption@settype{figure}}
2639 \expandafter\let\expandafter\caption@ORI@table
2640 \csname\string\table\endcsname
2641 \@namedef{\string\table} [#1] {%
2642 \caption@ORI@table [#1] %
2643 \caption@settype{table}}
2644 {}

```

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.

```

2645 \@ifundefined{scr@caption} {} {%
2646 \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)

```

2647 \@nameuse{caption@frenchb}\@nameundef{caption@frenchb}

```

2.14.5 The frenchle/pro package

```

2648 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods} {} {%
2649 \caption@Info{frenchle/pro package is loaded}%
2650 \let\captionfont@ORI\captionfont
2651 \let\captionlabelfont@ORI\captionlabelfont
2652 \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`.

```

2653 \@ifundefined{GOfrench}%
2654 {\let\caption@tempa\@firstofone}%
2655 {\def\caption@tempa{\g@addto@macro\GOfrench}}%
2656 \caption@tempa{%
2657 \let\captionfont\captionfont@ORI
2658 \let\captionfont@ORI\undefined
2659 \let\captionlabelfont\captionlabelfont@ORI
2660 \let\captionlabelfont@ORI\undefined
2661 \let\@makecaption\@makecaption@ORI
2662 \let\@makecaption@ORI\undefined

```

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

```

2663 \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.
2664 \let\caption@tcORI\@tablescaption
2665 \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.

2666 \let\@eatDP\@undefined
2667 \let\caption@tempa\@empty
2668 \ifx\ffrench\ffrench\ifnum@figure
2669 \l@addto@macro\caption@tempa{\let\fnum@figure\ffrench}%
2670 \fi
2671 \ifx\tfrench\tfrench\ifnum@table
2672 \l@addto@macro\caption@tempa{\let\fnum@table\tfrench}%
2673 \fi
2674 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
2675 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
2676 \caption@tempa

2677 }%
2678 }}

```

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.

2679 \newcommand\caption@IfPackageLoaded[1]{%
2680 \testopt{\caption@IfPackageLoaded{#1}}{}}
2681 \@onlypreamble\caption@IfPackageLoaded

2682 \long\def\caption@@IfPackageLoaded#1[#2]#3#4{%
2683 \@ifpackageloaded{#1}\@firstofone{%
2684 \caption@Debug{#1 package is not loaded (yet)}\@gobble}%
2685 \caption@AtBeginDocument}{%
2686 \caption@@ifpackageloaded{#1}[#2]{#3}{#4}}
2687 \@onlypreamble\caption@@IfPackageLoaded

2688 \newcommand\caption@ifpackageloaded[1]{%
2689 \testopt{\caption@ifpackageloaded{#1}}{}}
2690 \@onlypreamble\caption@ifpackageloaded

2691 \long\def\caption@@ifpackageloaded#1[#2]{%
2692 \@ifpackageloaded{#1}{%
2693 \caption@Info{#1 package is loaded}%
2694 \@ifpackagelater{#1}{#2}\@firstoftwo{%
2695 \caption@Error{%
2696 For a successful cooperation we need at least version\MessageBreak
2697 '#2' of package #1,\MessageBreak
2698 but only version\MessageBreak

```



```

2699         '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
2700         is available}%
2701     \@secondoftwo}%
2702 }{\@secondoftwo}}
2703 \@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.

```

2704 \newcommand*\caption@clearmargin{%
2705   \setcaptionmargin\z@
2706   \let\caption@minmargin\undefined}

2707 \caption@setbool{needfreeze}{0}
2708 \caption@AtBeginDocument*{%
2709   \caption@ifneedfreeze{%

```

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

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

```

2710 \newcommand*\caption@freeze{%
2711   \caption@teststar\caption@@freeze\@gobble\@firstofone}%

2712 \newcommand*\caption@@freeze[1]{%
2713   \global\let\caption@SCcontinued\relax
2714   \global\let\caption@SCsetup\@undefined
2715   \global\let\caption@SClentry\@undefined
2716   \global\let\caption@SCtext\@undefined
2717   \global\let\caption@SClabel\@undefined

2718   \let\caption@ORI@ContinuedFloat\ContinuedFloat
2719   \def\ContinuedFloat{%
2720     \caption@withoptargs\caption@SC@ContinuedFloat}%
2721   \def\caption@SC@ContinuedFloat##1{%
2722     \let\caption@ORI@setcounter\setcounter
2723     \let\caption@ORI@addtocounter\addtocounter
2724     \def\setcounter####1####2{\csname c@###1\endcsname####2\relax}%
2725     \def\addtocounter####1####2{\advance\csname c@###1\endcsname ####2\relax}%
2726     \caption@ORI@ContinuedFloat##1%
2727     \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
2728     \let\setcounter\caption@ORI@setcounter
2729     \let\addtocounter\caption@ORI@addtocounter}%
2730   \let\caption@ORI@setup\captionsetup
2731   \def\captionsetup{%
2732     \caption@withoptargs\caption@SC@setup}%
2733   \def\caption@SC@setup##1##2{%
2734     \caption@g@addto@list\caption@SCsetup{##2}%
2735     \caption@ORI@setup##1{##2}}%
2736   \let\caption@ORI\caption
2737   \def\caption{%
2738     \def\caption{\caption@Error{%
2739       Only one \noexpand\caption can be placed in this environment}}%
2740     \let\captionsetup\caption@setup
2741     \let\caption@@refstepcounter\caption@l@stepcounter
2742     \caption@ORI}%
2743   \long\def\@caption##1[##2]##3{%
2744     \@bsphack

```

```

2745         \gdef\caption@SClentry{##2}%
2746         \gdef\caption@SCText{##3}%
2747         \@esphack}%
2748     #1{% is \@gobble in star form, and \@firstofone otherwise
2749         \def\label##1{\@bsphack\gdef\caption@SClabel{##1}\@esphack}}%
2750 }%

\caption@defrost \caption@defrost
2751 \newcommand*\caption@defrost{%
2752     \ifx\caption@ORI@ContinuedFloat\@undefined
2753         \caption@defrost@setup
2754         \ifx\caption@SCText\@undefined \else
2755             \expandafter\expandafter\expandafter\caption
2756                 \expandafter\expandafter\expandafter[%
2757                 \expandafter\expandafter\expandafter{%
2758                 \expandafter\caption@SClentry\expandafter}\expandafter]%
2759             \expandafter{\caption@SCText}%
2760         \fi
2761         \ifx\caption@SClabel\@undefined \else
2762             \expandafter\label\expandafter{\caption@SClabel}%
2763         \fi
2764     \else
2765         \caption@Error{Internal Error:MessageBreak
2766             \noexpand\caption@defrost in same group as \string\caption@freeze}%
2767     \fi}%

2768 \newcommand*\caption@defrost@setup{%
2769     \caption@SCcontinued
2770     \ifx\caption@SCsetup\@undefined \else
2771         \expandafter\captionsetup\expandafter{\caption@SCsetup}%
2772     \fi}%

2773 }{}%
2774 \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 `\float@capt` 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.)

```

2775 \caption@IfPackageLoaded{float}[2001/11/08 v1.3d]{%
2776 \@ifpackageloaded{floatrow}{%
2777 \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]}{}%
2778 }%

```

`\@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]`.

```
2779 \let\caption@ORI@float@setevery\@float@setevery
2780 \def\@float@setevery#1{%
2781   \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.

```
2782   \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.

```
2783   \renewcommand\caption@setfloatcapt[1]{%
2784     \let\@makecaption\caption@make
2785     \global\setbox\@floatcapt\vbox{%
2786       \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.

```
2787   \float@getstyle\float@style{#1}%
2788   \caption@setstyle*\float@style
2789   \caption@setoptions\float@style
2790   }{}%
2791   \caption@freezeHref % will be defrosted in \float@makebox
2792   \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.

```
2793 \g@addto@macro\caption@typehook{%
2794   \expandafter\ifx\csname #1name\endcsname\relax
2795   \expandafter\let\csname #1name\expandafter\endcsname
2796   \csname fname@#1\endcsname
2797   \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.

```
2798 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2799 \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`).

```
2800 \providecommand*\float@ifstyle[1]{%
2801   \expandafter\ifx\csname fst@#1\endcsname\relax
2802     \expandafter\@secondoftwo
2803   \else
2804     \expandafter\@firstoftwo
2805   \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.

```
2806 \providecommand*\float@getstyle[2]{%
2807   \edef#1{%
2808     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2809     \expandafter\expandafter\expandafter\noexpand
2810     \csname fst@#2\endcsname}%
2811   \edef#1{#1}%
2812   \caption@Debug{floatstyle{#2} = '#1'}%}
```

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

```
2813 \providecommand*\float@setstyle[2]{%
2814   \expandafter\edef\csname fst@#1\endcsname{%
2815     \expandafter\noexpand\csname fs@#2\endcsname}}%
```

`\float@dostyle` `\float@dostyle{<type>}`
2816 `\providecommand*\float@dostyle[1]{%`
2817 `\@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.)

```

2818 \providecommand*\float@ifcaption[1]{%
2819   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2820   \expandafter\@firstoftwo
2821   \else
2822   \expandafter\@secondoftwo
2823   \fi}%

2824 }}{%
2825 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2826 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2827 % \clearcaptionsetup{boxed}% used by the floatrow package?
2828 }

```

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

```
2829 \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’.

```

2830 \caption@ifbool{ruled}{%
2831   \captionsetup[ruled]{margin=0pt,minmargin=0,slc=0}%
2832 }{%
2833   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2834 }
2835 \caption@undefbool{ruled}

```

2.15.2 The floatflt package

```
2836 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

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

```

2837 \let\caption@ORI@floatingfigure\floatingfigure
2838 \def\floatingfigure{%
2839   \caption@floatflt{figure}%
2840   \caption@ORI@floatingfigure}%

```

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

```

2841 \let\caption@ORI@floatingtable\floatingtable
2842 \def\floatingtable{%
2843   \caption@floatflt{table}%
2844 % \caption@setautoposition b%
2845   \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.

```

2846 \newcommand*\caption@floatflt[1]{%
2847   \caption@settype{#1}%
2848   \caption@clearmargin
2849   \caption@setoptions{floating#1}%
2850   \let\caption@boxrestore\@parboxrestore}%
2851 }{}

```

2.15.3 The fltpage package

```

2852 \caption@ifPackageLoaded{fltpage}[1998/10/29 v.0.3]{%
2853   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

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

2854 \renewcommand\FP@helpNote[2]{%
2855   \begingroup % save \caption@thepage
2856     \caption@pageref{#2}%
2857     \typeout{FP#1 is inserted on page \caption@thepage!}%
2858   \endgroup}%

```

`\FP@floatBegin` Original code:

```

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

2859 \renewcommand*\FP@floatBegin[1]{%
2860   \def\@capttype{#1}%
2861   \let\FP@LabelText\@empty
2862   \begin{lrbox}{\FP@floatCorpusBOX}%
2863     \caption@ifFPrefcap
2864       {\caption@freeze\relax}%
2865       {\def\label##1{\@bsphack\gdef\FP@LabelText{##1}\@esphack}%
2866         \caption@freeze*}%
2867     \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}}%
}%

2868 \renewcommand*\FP@floatEnd{%
2869   \end{lrbox}%
2870   \stepcounter{FP@\@capttype C}%
2871   \caption@label\FP@positionLabel
2872   \FP@helpNote\@capttype\FP@positionLabel
2873   \edef\FP@RestoreCounter{%
2874     \noexpand\setcounter{\@capttype}{\the\value\@capttype}%
2875     \noexpand\setcounter{ContinuedFloat}{\the\value{ContinuedFloat}}}%
2876   \FP@float
2877   {\FP@positionLabel}% location label test
2878   {\begin\@capttype[p!]}%
2879     \usebox\FP@floatCorpusBOX
2880     \caption@defrost@setup
2881     \caption@ifFPlistcap
2882     {\caption@refstepcounter\@capttype
2883       \expandafter\caption@makecurrent\expandafter\@capttype
2884         \expandafter{\caption@SCLentry}}%
2885     {\expandafter\caption@listentry\expandafter{\caption@SCLentry}}%
2886     \caption@makeanchor\relax
2887     \ifx\FP@LabelText\@empty \else
2888       \expandafter\label\expandafter{\FP@LabelText}%
```

```

2889         \fi
2890     \end\@capttype}%
2891     {\FP@RestoreCounter
2892     \@ifundefined{theH\@capttype}}{%
2893         \expandafter\l@addto@macro\csname theH\@capttype\endcsname{.FP}}}%
2894     {\begin\@capttype[b!]}%
2895         \let\FP@savedSetfnumCommand\caption@setfnum
2896         \def\caption@setfnum##1{%
2897             \FP@savedSetfnumCommand{##1}%
2898             \ifx\FP@guide\@empty \else
2899                 \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}}%
2900         \fi}%
2901         \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2902         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2903         \caption@setoptions{FP\@capttype}%
2904         \FP@separatorCaption
2905         \caption@ifFPlistcap{}{\let\caption@addcontentsline\@gobbletwo}%
2906         \caption@defrost
2907     \end\@capttype}%
2908 }%

2909 \caption@For{typelist}{%
2910     \newcounter{FP@#1C}%
2911     \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2912 }%
2913 \let\caption@ifFPlistcap\@undefined
2914 \let\caption@ifFPrefcap\@undefined
2915 }

```

2.15.4 The hyperref package

```

2916 \caption@IfPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2917     \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2918         \caption@WarningNoLine{%
2919             Hyperref support is turned off\MessageBreak
2920             because hyperref has stopped early}}%
2921     }{%
2922         \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`.**

```

2923     \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`.

```

2924     \renewcommand*\caption@makecurrent[2]{%
2925         \caption@makecurrentHref{#1}%
2926         \caption@Debug{hyperref current=\@currentHref}%
2927         \caption@getttitle{#2}}%
2928     \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`.

```
2929 \renewcommand\caption@makeanchor[1]{%
2930 \caption@Debug{hyperref anchor: \@currentHref}%
2931 % If we cannot have nesting, the anchor is empty.
2932 \ifHy@nesting
2933 \expandafter\hyper@@anchor\expandafter{\@currentHref}{#1}%
2934 \else
2935 \Hy@raisedlink{%
2936 \expandafter\hyper@@anchor\expandafter{\@currentHref}{\relax}%
2937 }#1%
2938 \fi}%
2939 \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.

```
2940 \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`.

```
2941 \def\caption@start{\caption@ifhycap\caption@start@relax}%
2942 \def\caption@start@{%
```

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

```
2943 \caption@makestart\@captype
2944 \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.)

```
2945 \global\@capstarttrue
2946 \let\hc@currentHref\@currentHref
2947 \def\caption@makecurrentHref##1{%
2948 \global\@capstartfalse
2949 \global\let\@currentHref\hc@currentHref}%
```

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

```
2950 \let\caption@makeanchor\@firstofone
2951 }%
```

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

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

-or-

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

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

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

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

```
2952 \newcommand*\caption@makestart[1]{%
2953 \begingroup
2954 \Hy@hypertextnamesfalse
2955 % \gdef\@currentHlabel{}%
2956 \hyper@makecurrent{#1.caption}%
2957 \endgroup
2958 \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.

```
2959 \newcommand*\caption@startanchor[1]{%
2960 \ifvmode\begingroup
2961 \caption@Debug{hycap anchor: #1 (vertical mode)}%
2962 \@tempdima\prevdepth
2963 \nointerlineskip
2964 \vspace*{-\caption@hycap@space}%
2965 \caption@anchor{#1}%
2966 \vspace*{\caption@hycap@space}%
2967 \prevdepth\@tempdima
2968 \endgroup\else
2969 \caption@Debug{hycap anchor: #1 (horizontal mode)}%
2970 \caption@anchor{#1}%
2971 \fi}%
```

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

```
2972 \newcommand*\caption@anchor[1]{%
2973 \ifmeasuring@ \else
2974 \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2975 \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`.)

```
2976 \ifx\HyperRaiseLinkLength\@tempdima
2977 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2978 \else
2979 \let\caption@raisedlink\Hy@raisedlink
2980 \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.

```
2981 \def\caption@@start{%
```

```

2982     \@ifundefined{hc@currentHref}{%
2983         \caption@Warning{%
2984             The option 'hycap=true' will be ignored for this\MessageBreak
2985             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`.

```

2986     \def\caption@freezeHref{%
2987         \let\caption@ORI@start\caption@start
2988         \def\caption@start{\let\caption@start\caption@ORI@start}%
2989 %
2990 %
2991 %
2992         \let\caption@ORI@@start\caption@@start
2993         \l@addto@macro\caption@subtypehook{%
2994             \let\caption@@start\caption@ORI@@start}%
2995 %
2996 %
2997         \global\let\caption@currentHref\@undefined
2998         \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2999         \let\caption@ORI@setfloatcapt\caption@setfloatcapt
3000         \renewcommand*\caption@setfloatcapt{%
3001             \ifx\caption@currentHref\@undefined \else
3002                 \let\caption@makeanchor\@firstofone
3003             \fi
3004             \caption@ORI@setfloatcapt}}%

```

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

```

3000     \def\caption@defrostHref{%
3001         \ifx\caption@currentHref\@undefined \else
3002             \caption@startanchor\caption@currentHref
3003             \global\let\caption@currentHref\@undefined
3004         \fi}%

```

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

```

3005     \@ifundefined{HyOrg@float@makebox}{}{%
3006         \caption@Debug{%
3007             Redefining \noexpand\float@makebox (again)\@gobble}%
3008         \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
3009         \renewcommand\float@makebox[1]{%
3010             \HyOrg@float@makebox{#1\relax \caption@defrostHref}}%
3011     }%
3012 }}{}

```

2.15.5 The `hycap` package

```

3013 \caption@IfPackageLoaded{hycap}{% v1.0
3014     \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* `\captions` in the same environment, at least unless a new `\capstart` will be placed.

```

3015 \let\caption@ORI@capstart\capstart % save for compatibility mode
3016 \@ifundefined{capstarttrue}% check for v1.10 of hypcap package
3017   {\def\capstart{\caption@start@}}%
3018   {\def\capstart{\ifcapstart\caption@start@\fi}}%
3019 \let\caption@start\relax
3020 \let\caption@@start\relax

```

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

```

3021 \caption@set@bool\caption@ifhypcap 1%
3022 \renewcommand*\caption@hypcapspace{\hypcapspace}%
3023 \fi}}

```

2.15.6 The listings package

```

3024 \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.

```

3025 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
3026 \def\lst@MakeCaption#1{% #1 is ‘t’ or ‘b’
3027   \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`.)

```

3028   \caption@setposition{#1}%
3029   \caption@iftop{%
3030     \@tempdima\belowcaptionskip
3031     \belowcaptionskip\abovecaptionskip
3032     \abovecaptionskip\@tempdima}}%

```

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

```

3033   \caption@setup{rule=0}%

```

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

```

3034   \caption@setoptions{lstlisting}%

```

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

```

3035   \caption@setautoposition{#1}%

```

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

```

3036   \caption@begin{lstlisting}%
3037   \caption@ORI@lst@MakeCaption{#1}%
3038   \caption@end
3039 \endgroup}%

```

```

\lst@makecaption  Wrapper macros for typesetting the caption= resp. title= value.
\lst@maketitle   3040 \def\lst@makecaption{\caption@starfalse\@makecaption}%
                 3041 \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.
                 3042 \providecommand*\ext@lstlisting{lol}%
                 3043 {}

```

2.15.7 The longtable package

```

\LTcapttype     \LTcapttype is preset to table.
                 3044 \providecommand*\LTcapttype{table}

                 3045 \caption@IfPackageLoaded{longtable}[1995/05/24 v3.14]{%
                 3046 \RequirePackage{ltcaption}[2007/09/01]%
                 3047 \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.

                 3048 \caption@AtBeginDocument{%
                 3049 \let\caption@ORI@LT@array\LT@array
                 3050 \renewcommand*\LT@array{%

\captionsetup for longtable:
                 3051 \global\let\caption@opt@@longtable\@undefined
                 3052 \def\captionsetup{%
                 3053 \noalign\bgroup
                 3054 \@ifstar\@captionsetup\@captionsetup}% gobble *
                 3055 \def\@captionsetup##1{\LT@captionsetup{##1}\egroup}%
                 3056 \def\LT@captionsetup##1{%
                 3057 \captionsetup@startrue\caption@setup@options[@longtable]{##1}%
                 3058 \global\let\caption@opt@@longtable\caption@opt@@longtable}%

\captionabove & \captionbelow for longtable: (KOMA-Script document class)
                 3059 \def\@captionabovetrue{\LT@captionsetup{position=t}}%
                 3060 \def\@captionabovefalse{\LT@captionsetup{position=b}}%

\captionlistentry for longtable:
                 3061 \def\captionlistentry{%
                 3062 \noalign\bgroup
                 3063 \@ifstar{\egroup\LT@captionlistentry}% gobble *
                 3064 {\egroup\LT@captionlistentry}}%
                 3065 \def\LT@captionlistentry##1{%
                 3066 \caption@listentry\@firstoftwo[\LTcapttype]{##1}}%

\ContinuedFloat for longtable:
(Commented out, since it's not deeply tested and quite useless anyway)
Note: hyperref versions < v6.76j uses 2x \hyper@makecurrent
                 3067 % \caption@ifhycap{%

```

```

3068 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
3069 % \def\hyper@makecurrent##1{%
3070 % \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
3071 % \caption@makestart{##1}%
3072 %% \let\Hy@LT@currentHlabel\@currentHlabel
3073 % \let\Hy@LT@currentHref\@currentHref
3074 % \def\hyper@makecurrent###1{%
3075 %% \let\@currentHlabel\Hy@LT@currentHlabel
3076 % \let\@currentHref\Hy@LT@currentHref}}%
3077 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
3078 % \def\ContinuedFloat{\noalign{%
3079 % \gdef\caption@setContinuedFloat{%
3080 % \let\caption@resetContinuedFloat\@gobble}%
3081 % \caption@setoptions###1{%
3082 % \g@addto@macro\caption@setContinuedFloat{%
3083 % \caption@setoptions{###1}}}%
3084 % \let\@capttype\LTcapttype
3085 % \caption@ORI@ContinuedFloat}}%
3086 % }{%
3087 % \def\ContinuedFloat{\noalign{%
3088 % \caption@Error{%
3089 % \noexpand\ContinuedFloat inside longtables\MessageBreak
3090 % is only available with 'hycap=true'}}}%
3091 % }%
3092 % \global\let\caption@setContinuedFloat\@empty
3093 % \def\ContinuedFloat{\noalign{%
3094 % \caption@Error{\noexpand\ContinuedFloat outside float}}}%
3095 % \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}`:

```

3096 \long\def\LT@c@ption#1[#2]#3{%
3097 \LT@makecaption#1{\csname fnun@\LTcapttype\endcsname}{#3}%
3098 \LT@captionlistentry{#2}}%

```

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

The original definition:

```

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

```

```

#1{#2: }#3%
\else
\hbox to\hsize{\hfil\box\@tempboxa\hfil}%
\fi
\endgraf\vskip\baselineskip}%
\hss}}

```

Our definition:

```

3099 \renewcommand\LT@makecaption[3]{%
3100 \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`.)

```

3101 \caption@settype*\LTcaptype
3102 \ifdim\LTcapwidth=4in \else
3103 \setcaptionwidth\LTcapwidth
3104 \fi
3105 \caption@setoptions{longtable}%
3106 % \caption@setContinuedFloat
3107 \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!

```

3108 \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%

```

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

```

3109 \caption@startrue#1\caption@starfalse
3110 \caption@resetContinuedFloat\LTcaptype
3111 \caption@begin\LTcaptype
3112 \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...)

```

3113 \vskip-\ht\strutbox

```

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

```

3114 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
3115 \caption@@make{#2}{#3}\endgraf
3116 \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
3117 \caption@end}}%

```

```

3118 }{}

```

2.15.8 The picinpar package

```
3119 \caption@ifpackageloaded{picinpar}{%
```

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

```
3120 \long\def\figwindow[#1,#2,#3,#4] {%  
3121   \caption@window{figure}%  
3122   \caption@setoptions{figwindow}%  
3123   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%  
  
3124 \long\def\tabwindow[#1,#2,#3,#4] {%  
3125   \caption@window{table}%  
3126   \caption@setoptions{tabwindow}%  
3127   \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.
```

```
3128 \newcommand*\caption@window[1]{%  
3129   \let\caption@boxrestore\@parboxrestore  
3130   \let\@makecaption\caption@@make  
3131   \caption@setautoposition b%  
3132   \caption@settype{#1}%  
3133   \caption@clearmargin}%
```

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

```
3134 \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.
```

```
3135   \ifdim\picwd=\z@  
3136     \let\caption@makecurrent\@gobbletwo  
3137     \let\caption@@start\relax  
3138     \caption@prepareslc  
3139   \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.

```
3140   \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%  
3141   \if\@tempa*%  
3142     \let\@tempa\@firstofone  
3143   \else\if\@tempa[%]  
3144     \let\@tempa\@firstofone  
3145   \else  
3146     \let\@tempa\@empty  
3147   \fi\fi  
3148   \expandafter\caption\@tempa{#1}}%
```

```
3149 }{ }
```


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.

```
3150 \newcommand*\piccaptiontype[1]{\def\@piccapttype{#1}}
```

```
3151 \caption@ifpackageloaded{picins}{%
```

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

```
3152 \@ifundefined{@piccapttype}{%
```

```
3153   \caption@iftype{%
```

```
3154     \let\@piccapttype\@capttype
```

```
3155   }{%
```

```
3156     \def\@piccapttype{figure}%
```

```
3157   }{%
```

```
3158 }{ }%
```

```
3159 \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.

```
3160 \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.

```
3161 \let\caption@ORI@make@piccaption\make@piccaption
```

```
3162 \def\make@piccaption{%
```

```
3163   \let\caption@ORI\caption
```

```

3164 \long\def\caption[##1]##2{%
3165 \caption@freezeHref % will be defrosted in \ivparpic
3166 \caption@settype\@piccaptiontype
3167 % \ifnum\c@piccaptionpos>2\relax
3168 \caption@clearmargin
3169 % \else
3170 % \captionwidth\z@ % do not use "width=" setting
3171 % \fi
3172 \caption@setoptions{parpic}%
3173 \let\caption@boxrestore\@parboxrestore
3174 \caption@setautoposition b%
3175 \expandafter\expandafter\expandafter\caption@ORI
3176 \expandafter\expandafter\expandafter[%
3177 \expandafter\expandafter\expandafter{%
3178 \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
3179 \caption@ORI@make@piccaption
3180 \let\caption\caption@ORI%

```

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

```

3181 \let\caption@ORI@ivparpic\ivparpic
3182 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
3183 \let\caption@ORI@noindent\noindent
3184 \def\noindent{%
3185 \caption@defrostHref
3186 \let\noindent\caption@ORI@noindent
3187 \noindent}%
3188 \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
3189 \let\noindent\caption@ORI@noindent}%
3190 }{%
3191 \let\piccaptiontype\@undefined
3192 }

```

2.15.10 The rotating package

```
3193 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%
```

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

```
3194 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
3195 % \let\@rotcaption\@undefined

```

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

```
3196 \def\rotcaptionof{%
3197 \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

```

`\@makerotcaption` Original (bugfixed) code:

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

```
3198 \long\def\@makerotcaption#1#2{%
3199   \ifdim\captionwidth=\z@
3200     \setcaptionwidth{.8\textheight}%
3201     \caption@slc{#1}{#2}{.8\vszise}{%
3202       \let\caption@makerot\caption@@make
3203       \caption@clearmargin
3204 %       \long\def\caption@parbox##1##2{\hbox{\hszise=.8\textheight\relax##2}}%
3205 %       (not needed because \rotatebox uses an \hbox anyway)
3206       \let\caption@parbox\@secondoftwo}%
3207       \caption@set@bool\caption@ifslc0% been there, done that
3208     \fi
3209     \rotatebox{90}{\caption@makerot{#1}{#2}}%
3210     \nobreak\hspace{12pt}}%
3211 \newcommand\caption@makerot[2]{%
3212   \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
3213 \caption@For{typelist}{%
3214   \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
3215   \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
3216 {} }
```

2.15.11 The sidecap package

```
3217 \caption@IfPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
3218   \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.)

```
3219   \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`.

```
3220   \let\caption@ORI@SC@zfloat\SC@zfloat
3221   \def\SC@zfloat#1#2#3[#4]{%
```

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

```
3222 \let\caption@ORI\caption
3223 \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
3224 \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.

```
3225 \caption@settype*{#2}%
3226 \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.

```
3227 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
3228 \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.

```
3229 \let\caption@ORI@settype\caption@settype
3230 \def\caption@settype##1{% will be done in \@xfloat
3231 \caption@ORI@settype*{##1}% do not change \@currentlabel
3232 \caption@setSC@justify
3233 %%% \caption@setoptions{SCfloat}%
3234 \caption@setoptions{SC\@capttype}%
3235 \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

```
3236 \def\caption@setSC@justify{%
3237 \caption@clearmargin
3238 \@ifundefined{SC@justify}{}{%
3239 \ifx\SC@justify\@empty \else
3240 \let\caption@hj\SC@justify
3241 \let\SC@justify\@empty
3242 \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!

```
3243 \let\caption\SC@orig@caption
3244 \def\SC@orig@caption[##1]##2{\caption@defrost}%
```

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

```
3245 \caption@setSC@justify % for compatibility mode
3246 \caption@ORI@endSC@FLOAT}%
```

```

3247 \newcommand*\caption@For@SC[2]{%
3248   \def#1{b}% = \sidecaptionvpos{#2}{b} (v1.6)
3249   \newenvironment{SC#2}%
3250     {\SC@float[#1]{#2}}{\endSC@float}%
3251   \newenvironment{SC#2*}%
3252     {\SC@dblfloat[#1]{#2}}{\endSC@dblfloat}}%
3253 \@onlypreamble\caption@For@SC
3254 \caption@For{typelist}{%
3255   \expandafter\caption@For@SC\csname SC@#1@vpos\endcsname{#1}}%
3256 {}

```

2.15.12 The subfigure package

```

3257 \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.

```

3258 \def\sf@ifpositiontop{%
3259   \ifx\@capttype\@undefined
3260     \expandafter\@gobbletwo
3261   \else\ifx\@capttype\relax
3262     \expandafter\expandafter\expandafter\@gobbletwo
3263   \else
3264     \expandafter\expandafter\expandafter\sf@if@position@top
3265   \fi\fi}
3266 \def\sf@if@position@top{%
3267   \@ifundefined{if\@capttype topcap}%
3268     {\@gobbletwo}%
3269     {\@nameuse{if\@capttype topcap}}%
3270     \expandafter\@firstoftwo
3271   \else
3272     \expandafter\@secondoftwo
3273   \fi}}
3274 {}

```

2.15.13 The supertabular and xtab packages

```

3275 \caption@IfPackageLoaded{supertabular}[2002/07/19 v4.1e]{%

```

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

```

3276 \renewcommand*\tablecaption{%
3277   \caption@star
3278   {\refstepcounter{table}}%
3279   {\caption@dblarg{\@xtablecaption}}}%

```

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

```

3280 \let\caption@ORI@xtablecaption\@xtablecaption
3281 \long\def\@xtablecaption[#1]#2{%
3282   \caption@getttitle{#2}%
3283   \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}
```

```
3284 \long\def\ST@caption#1[#2]#3{\par%
3285   \caption@settype*{#1}%
3286   \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.

```
3287   \def\caption@fixposition{%
3288     \caption@setposition{\if@topcaption t\else b\fi}}%
3289   \caption@beginex{#1}{#2}{#3}%
3290   \caption@boxrestore
3291   \caption@normalsize
3292   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3293   \caption@end}%
```

```
3294 }{ }
```

```
3295 \caption@IfPackageLoaded{xtab}[2000/04/09 v2.3]{%
```

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

```
3296 \renewcommand*\tablecaption{%
3297   \caption@star
3298   {\refstepcounter{table}}%
3299   {\caption@dblarg{\@tablecaption}}}%
```

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

```
3300 \let\caption@ORI@xtablecaption\@xtablecaption
3301 \long\def\@xtablecaption[#1]#2{%
3302   \caption@getttitle{#2}%
3303   \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}}

3304 \long\def\ST@caption#1[#2]#3{\par%
3305   \caption@settype*{#1}%
3306   \caption@setoptions{xtabular}%

3307   \def\caption@fixposition{%
3308     \caption@setposition{\if@topcaption t\else b\fi}}%

3309   \@initisotab
3310   \caption@beginex{#1}{#2}{#3}%
3311   \caption@boxrestore
3312   \caption@normalsize
3313   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3314   \caption@end
3315   \global\advance\ST@pageleft -\PWSTcapht
3316   \ST@trace\tw@{Added caption. Space left for xtabular: \the\ST@pageleft}}%

3317 {}

```

2.15.14 The threeparttable package

```

3318 \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.

```

```

3319 \let\caption@ORI@threeparttable\threeparttable
3320 \renewcommand*\threeparttable{%
3321   \caption@settype{table}%
3322   \caption@setposition a% ?
3323   \caption@clearmargin
3324   \caption@setoptions{threeparttable}%
3325   \caption@ORI@threeparttable}%

```

\measuredfigure Same here...

```

3326 \let\caption@ORI@measuredfigure\measuredfigure
3327 \renewcommand*\measuredfigure{%
3328   \caption@settype{figure}%
3329   \caption@setposition a% ?
3330   \caption@clearmargin
3331   \caption@setoptions{measuredfigure}%
3332   \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}

```

```

3333 \def\TPT@caption#1[#2]#3{%
3334   \gdef\TPT@docapt{%
3335     \global\let\TPT@docapt\@undefined
3336     \caption@setautoposition\caption@TPT@position
3337     \TPT@LA@caption{#1}[#2][#3]}%
3338   \ifx\TPT@hsize\@empty
3339     \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
3340     \gdef\caption@TPT@position{t}%
3341     \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
3342   \else
3343     \def\caption@TPT@position{b}%
3344     \TPT@docapt
3345   \fi
3346   \ignorespaces}%

3347 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
3348 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%

3349 }{}

```

2.15.15 The wrapfig package

```

3350 \caption@IfPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)

```

```

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

```

```

3351 \providecommand*\float@ifstyle[1]{%
3352   \expandafter\ifx\csname fst@#1\endcsname\relax
3353     \expandafter\@secondoftwo
3354   \else
3355     \expandafter\@firstoftwo
3356   \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.

```

3357 \newcommand*\caption@restylewrapfloat[1]{%
3358   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
3359     \csname wrap#1\endcsname
3360   \caption@Error{%
3361     For a successful cooperation of the 'wrapfig' package\MessageBreak
3362     with the 'float' package you should load the 'wrapfig'\MessageBreak
3363     package *after* the 'float' package}%
3364   \else
3365     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
3366       \csname wrap#1\endcsname
3367     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
3368     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
3369       \csname wrap#1\endcsname
3370   \fi}%

```


`\caption@wrapfloat`

```
3371 \newcommand*\caption@wrapfloat[1]{%
3372   \caption@settype*{#1}%
3373   \float@ifstyle{#1}{%
3374     \ifx\WF@floatstyhook\@undefined
3375       \caption@Error{%
3376         For a successful cooperation of the 'wrapfig' package\MessageBreak
3377         with the 'float' package you should use at least\MessageBreak
3378         'wrapfig' version 3.6}%
3379     \else
3380       \float@dostyle{#1}%
3381     \fi}{}%
3382   \caption@clearmargin
3383 %%% \caption@setoptions{wrapfloat}%
3384   \caption@setoptions{wrap#1}%
3385   \@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`.)

```
3386 \caption@restylewrapfloat{figure}%
3387 \caption@restylewrapfloat{table}%
3388 \caption@For{typelist}{%
3389   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
3390   \caption@restylewrapfloat{#1}}%
3391 \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.

```
3392 \@ifundefined{float@restyle}{}{%
3393   \toks@=\expandafter{\float@restyle{#1}} (env may or may not be defined)
3394   \caption@restylewrapfloat{#1}}%
3395   \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
3396   \@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`.

```
3397 \let\caption@ORI@wrapfloat\wrapfloat
3398 \def\wrapfloat#1{%
3399   \float@ifstyle{#1}{%
3400     \caption@Error{%
3401       For a successful cooperation of the 'wrapfig' package\MessageBreak
3402       with the 'float' package you should load the 'wrapfig'\MessageBreak
3403       package *right after* the 'float' package}}}%
3404   \caption@ORI@wrapfloat{#1}}%
3405 \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:

```
3406 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
3407   \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
3408   \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
3409   \caption@start
3410   \ifdim\hsize>\z@ \@parboxrestore \else
3411   \setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
3412   \ignorespaces \fi}%
3413 }
```

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