

The Implementation of the caption package^{*}

Axel Sommerfeldt
`caption@sommerfee.de`

2010/01/09

Abstract

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

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

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

User manuals

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

<code>caption-eng.pdf</code>	The English documentation
<code>caption-rus.pdf</code>	The Russian documentation ¹
<code>caption-deu.pdf</code>	The German documentation

^{*}This package has version number v3.11, 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}[2010/01/09 v3.11 caption3 kernel (AR)]
```

1.2 Generic helpers

`\@nameundef` This is the opposite to `\@namedef` which is offered by the \LaTeX kernel. We use it to remove the definition of some commands and keyval options after `\begin{document}` (to save \TeX memory) and to remove caption options defined with `\captionsetup[⟨type⟩]`.

```
3 \providecommand*\@nameundef[1]{%
4   \expandafter\let\csname #1\endcsname\@undefined}
```

`\l@addto@macro` The $\text{\LaTeX 2}\epsilon$ kernel offers the internal helper macro `\g@addto@macro` which globally adds tokens to existing macros, like in `\AtBeginDocument`. This is the same but it works local, not global (using `\edef` instead of `\xdef`).

```
5 \providecommand\l@addto@macro[2]{%
6   \begingroup
7     \toks@\expandafter{#1#2}%
8     \edef\@tempa{\endgroup\def\noexpand#1{\the\toks@}}%
9   \@tempa}
```

`\bothIfFirst` `\bothIfFirst` tests if the first argument is not empty, `\bothIfSecond` tests if the second argument is not empty. If yes both arguments get typeset, otherwise none of them.

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

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

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

would expand to yes.

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

```

```

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

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

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

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

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

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

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

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

```

```

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

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

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

\caption@teststar \caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
\caption@teststar@{<cmd>}{<star arg>}{<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

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

```

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 L^AT_EX 2_ε.

```

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 L^AT_EX 2_ε), so the keyval package is used. The starred form makes the option available during the lifetime of the current package only, so they can be used with \usepackage, but *not* with \captionsetup later on.

```

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\csname 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\expandafter\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@addtooptlist{⟨type⟩}
caption@removefromoptlist \caption@removefromoptlist{⟨type⟩}

```

Adds or removes an `⟨type⟩` to the list of unused caption options. Note that the catcodes of `⟨type⟩` are sanitized here so removing `⟨type⟩` from the list do not fail when the float package is used (since `\float@getstyle` gives a result which tokens have catcode 12 = “other”).

```

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` and `\captionwidth` contain the extra margin resp. the total width used for captions. Please never set these values in a direct way, they are just accessible in user documents to provide compatibility to *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@ifoddpage` 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 $\text{\LaTeX}2\epsilon$. It will be written to the `.aux` file to pass label information from one run to another. (We use it inside `\caption@ifoddpage` and `\caption@ragged`.)

```

345 \newcommand*\caption@newlabel{\@newlabel\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{indention}[\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 L^AT_EX 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\@nil{%
506   \def\@tempa{#3}\ifx\@tempa\@empty \else
507     \caption@g@addto@list\caption@lsep@crlist{#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@setlabelseparator{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{<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` `\belowcaptionskip` Usually these skips are defined within the document class, but some document classes don't do so.

```

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\abovcaptionskip
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}
```

```

784 \fi
785 \vskip\belowcaptionskip}}

```

1.19.2 The $\mathcal{A}\mathcal{M}\mathcal{S}$ & 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@lsep@crlist\caption@lsep@name
919     {\caption@Error{%
920       The option ‘\labelsep=\caption@lsep@name’ 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 frenchb Babel option

```

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

```

```

975 \caption@CheckCommand\@makecaption{%
976   % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
977   % frenchb.ldf [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@lstfmt{\@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@ifoddp@ge{\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: \captionindent 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 indentation.
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}{\@%
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\addvspace{10\p@}}%
1377   \addtocontents{lot}{\protect\addvspace{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\addvspace{\@chapterlistsgap}}%
1400     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1401     \float@addtolists{\protect\addvspace{\@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\addvspace{\@chapterlistsgap}}%
1473 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1474 \float@addtolists{\protect\addvspace{\@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\addvspace{10\p@}}%
1498 \addtocontents{lot}{\protect\addvspace{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\addvspace{\@chapterlistsgap}}%
1521 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1522 \float@addtolists{\protect\addvspace{\@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\addvspace{10pt}}
1557 \addtocontents{lot}{\protect\addvspace{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 `ltcounts.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 \renewcommand*\sf@updatecaptionlist[4]{%
1705 \xdef\sfcaptionlist{%
1706 \sf@captionlist,%
1707 {\protect\numberline{\@subcaplabel}\noexpand\ignorespaces #2}}%
1708 {\@currentHref}}}%

1709 \fi
1710 \let\sfc@NEW@dottedxxxline\@undefined
1711 \let\sfc@NEW@subfloat\@undefined

1712 \def\sfc@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1713 \begingroup
1714 \caption@setfloattype{#1}%
1715 \caption@setoptions{subfloat}%
1716 \caption@setoptions{sub#1}%
1717 \ifnum #3>\@nameuse{c@#2depth}\else
1718 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1719 \fi
1720 \endgroup}

1721 \def\sfc@NEW@subfloat{%
1722 \begingroup
1723 \caption@setfloattype\@capttype
1724 \sf@ifpositiontop{%
1725 \maincaptiontoptrue
1726 }{%
1727 \maincaptiontopfalse
1728 }%
1729 \caption@setoptions{subfloat}%
1730 \caption@setoptions{sub\@capttype}%
1731 \let\sfc@oldlabel=\label
1732 \let\label=\subfloat@label
1733 \ifmaincaptiontop\else
1734 \advance\@nameuse{c@\@capttype}\@ne
1735 \fi
1736 \refstepcounter{sub\@capttype}%
1737 \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1738 \@ifnextchar [% %] match left bracket
1739 {\sf@@subfloat}%
1740 {\sf@@subfloat[\@empty]}}

```

2 Main package

2.1 Identification

```
1741 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1742 \ProvidesPackage{caption}[2010/01/09 v3.11 Customizing captions (AR)]
1743 %\@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.

1744 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1745 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1746 \RequirePackage{caption3}[2008/08/24] % needs v3.1j or newer
```

2.3 Check against incompatible document classes

```
1747 \caption@ifbool{documentclass}{}{%
1748   \caption@WarningNoLine{%
1749     Unsupported document class (or package) detected, \MessageBreak
1750     usage of the caption package is not recommended}%
1751   \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1752 }
```

2.4 Check against incompatible packages

```
1753 \@ifpackageloaded{caption2}{}%
1754   \caption@Error{%
1755     You can't use both, the (obsolete) caption2 *and* \MessageBreak
1756     the (current) caption package}%
1757   \endinput
1758 }{}

1759 \caption@AtBeginDocument{%
1760   \@ifpackageloaded{ftcap}{\caption@DisablePositionOption{ftcap}}{}%
1761   \@ifpackageloaded{nonfloat}{\caption@DisablePositionOption{nonfloat}}{}%
1762   \@ifpackageloaded{topcapt}{\caption@DisablePositionOption{topcapt}}{}%
1763   \caption@DisablePositionOption{<package>}
1764   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


```

1771 \DeclareCaptionOption{figureposition}{%
1772   \captionsetup*[figure]{position=#1}}
1773 \@onlypreamble@key{caption}{figureposition}

1774 \DeclareCaptionOption{tableposition}{%
1775   \captionsetup*[table]{position=#1}}
1776 \@onlypreamble@key{caption}{tableposition}

1777 \DeclareCaptionOption{figurename}{\caption@SetName{figure}{#1}}
1778 \DeclareCaptionOption{tablename}{\caption@SetName{table}{#1}}
1779 \DeclareCaptionOption{name}{\caption@setname\@capttype{#1}}

1780 \DeclareCaptionOption{listfigurename}{\caption@SetName{listfigure}{#1}}
1781 \DeclareCaptionOption{listtablename}{\caption@SetName{listtable}{#1}}

\caption@SetName \caption@SetName{<cmd>}{<value>}
1782 \newcommand*\caption@SetName[2]{%
1783   \caption@setname{#1}{#2}%
1784   \begingroup
1785     \ifundefined{language}{\language}{%
1786       \ifundefined{captions\language}{\language}{%
1787         \expandafter\g@addto@macro\csname captions\language\endcsname
1788           {\caption@setname{#1}{#2}}}%
1789     \endgroup}

1790 \newcommand*\caption@setname[2]{\@namedef{#1name}{#2}}
1791 \caption@AtBeginDocument{\let\caption@SetName\caption@setname}

\caption@DeclareWithinOption
1792 \newcommand*\caption@DeclareWithinOption[1]{%
1793   \DeclareCaptionOption{#1within}{\caption@Within{#1}{##1}}%
1794   \DeclareCaptionOption{#1without}{\caption@Within{#1}{none}}}

1795 \caption@DeclareWithinOption{figure}
1796 \caption@DeclareWithinOption{table}

1797 \DeclareCaptionOption{within}{%
1798   \ifundefined{c@figure}{\caption@Within{figure}{#1}}%
1799   \ifundefined{c@table}{\caption@Within{table}{#1}}%
1800   \caption@For{typelist}{\caption@Within{##1}{#1}}%
1801   \def\caption@within@default{#1}}
1802 \DeclareCaptionOption{without}{\KV@caption@within{none}}

\caption@within
1803 \newcommand*\caption@Within[1]{\def\caption@type{#1}\KV@caption@DCT@within}

```

2.5.2 Miscellaneous options

```

1804 \DeclareCaptionOption*{config}[caption]{%
1805   \InputIfFileExists{#1.cfg}%
1806   {\typeout{*** Local configuration file #1.cfg used ***}}%
1807   {\caption@Warning{Configuration file #1.cfg not found}}}

1808 \DeclareCaptionOption{@minipage}{%
1809   \caption@ifinlist{#1}{auto,default}%
1810   {\let\caption@if@minipage\@gobbles}%
1811   {\caption@set@bool\caption@if@minipage{#1}}}

```

```
1812 \captionsetup{@minipage=default}
```

2.5.3 caption v1.x compatibility options

```
1813 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1814 \@onlypreamble@key{caption}{compatibility}

1815 \DeclareCaptionOptionNoValue*{normal}{%
1816   \caption@setformat{plain}%
1817   \caption@setjustification{justified}}
1818 \DeclareCaptionOptionNoValue*{isu}{%
1819   \caption@setformat{hang}%
1820   \caption@setjustification{justified}}
1821 \DeclareCaptionOptionNoValue*{hang}{%
1822   \caption@setformat{hang}%
1823   \caption@setjustification{justified}}
1824 \DeclareCaptionOptionNoValue*{center}{%
1825   \caption@setformat{plain}%
1826   \caption@setjustification{centering}}
1827 \DeclareCaptionOptionNoValue*{anne}{%
1828   \caption@setformat{plain}%
1829   \caption@setjustification{centerlast}}
1830 \DeclareCaptionOptionNoValue*{centerlast}{%
1831   \caption@setformat{plain}%
1832   \caption@setjustification{centerlast}}

1833 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1834 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1835 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1836 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1837 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1838 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1839 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1840 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1841 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1842 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1843 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1844 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1845 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1846 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1847 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1848 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}

1849 \caption@setbool{ruled}{0}
1850 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}
```

2.5.4 caption2 v2.x compatibility options

```
1851 \DeclareCaptionOptionNoValue*{flushleft}{%
1852   \caption@setformat{plain}%
1853   \caption@setjustification{raggedright}}
1854 \DeclareCaptionOptionNoValue*{flushright}{%
1855   \caption@setformat{plain}%
1856   \caption@setjustification{raggedleft}}

1857 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}

1858 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
```

```
1859 \caption@WarningNoLine{Obsolete option 'ignoreLTcapwidth' ignored}}
```

2.5.5 Obsolete caption v3.0 options

```
1860 \DeclareCaptionOption*{caption}{%
1861   \caption@setbool{temp}{#1}%
1862   \caption@ifbool{temp}{}{%
1863     \caption@Error{%
1864       The package option 'caption=#1' is obsolete.\MessageBreak
1865       Please pass this option to the subfig package instead\MessageBreak
1866       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.

```
1867 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1868 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1869 \@onlypreamble@key{caption}{FPlist}
1870 \@onlypreamble@key{caption}{FPref}

1871 \newcommand*\caption@setFPoption[2]{%
1872   \edef\caption@tempa{\@car#2\@nil}%
1873   \caption@setbool{FP#1cap}{\if c\caption@tempa 1\else 0\fi}}
1874 \@onlypreamble\caption@setFPoption

1875 \captionsetup{FPlist=caption,FPref=figure}
```

2.5.7 hyperref package support options

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

```
1876 \DeclareCaptionOption{hypcap}[1]{\caption@setbool{hypcap}{#1}}
1877 \DeclareCaptionOption{hypcapSPACE}{\def\caption@hypcapSPACE{#1}}

1878 \captionsetup{hypcap=1,hypcapSPACE=.5\baselineskip}
```

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

```
1879 \caption@ifamsclass{%
1880   \caption@Info{AMS or SMF document class}%
1881   \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1882 }
```

2.7 KOMA-Script document classes support

```
1883 \caption@ifkomaclass{%
1884   \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 1885 \g@addto@macro\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1886 \g@addto@macro\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}
```

```

1887 \if@tablecaptionabove
1888 \@tablecaptionabovetrue
1889 \else
1890 \@tablecaptionabovefalse
1891 \fi

\onelinecaptionstrue
\onelinecaptionsfalse 1892 \g@addto@macro\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1893 \g@addto@macro\onelinecaptionsfalse{\let\caption@ifslc\@secondoftwo}

1894 \ifonelinecaptions
1895 \onelinecaptionstrue
1896 \else
1897 \onelinecaptionsfalse
1898 \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.

1899 \g@addto@macro\@captionabovetrue{\let\caption@position\@firstoftwo}
1900 \g@addto@macro\@captionabovefalse{\let\caption@position\@secondoftwo}

1901 \DeclareCaptionOption{figureposition}{%
1902 \caption@WarningNoLine{Option 'figureposition=#1' has no effect\MessageBreak
1903 when used with a KOMA script document class}}
1904 \DeclareCaptionOption{tableposition}{%
1905 \caption@WarningNoLine{Option 'tableposition=#1' has no effect\MessageBreak
1906 when used with a KOMA script document class}}

\setcapindent

1907 \let\caption@KOMA@setcapindent\@setcapindent
1908 \renewcommand*\@setcapindent[1]{%
1909 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

1910 \let\caption@KOMA@@setcapindent\@@setcapindent
1911 \renewcommand*\@@setcapindent[1]{%
1912 \caption@KOMA@@setcapindent{#1}\caption@setcapindent}

1913 \newcommand*\caption@setcapindent{%
1914 \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}

1915 \@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.)

1916 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1917 \csname\string\setcapwidth\endcsname
1918 \@namedef{\string\setcapwidth}[#1]#2{%
1919 \caption@KOMA@setcapwidth[#1]#2}\caption@setcapwidth{#1}}

1920 \newcommand*\caption@setcapwidth[1]{%
1921 \ifx\#1\\\else
1922 \@ifundefined{cap@margin}{}{%
1923 \def\@tempa{captionbeside}%
1924 \ifx\@tempa\@currenvir\else\caption@Warning{%
1925 Ignoring optional argument [#1] of \string\setcapwidth\MessageBreak}%

```

```

1926         \fi}%
1927     \fi
1928     \captionsetup{width=\cap@width}}

1929 \def\caption@tempa{\hsize}%
1930 \ifx\caption@tempa\cap@width \else
1931     \caption@setcapwidth{?}
1932 \fi

\setcapmargin

1933 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1934         \csname\string\@setcapmargin\endcsname
1935 \@namedef{\string\@setcapmargin}[#1]#2{%
1936     \caption@KOMA@setcapmargin[#1]{#2}\caption@setcapmargin}

1937 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1938         \csname\string\@@setcapmargin\endcsname
1939 \@namedef{\string\@@setcapmargin}[#1]#2{%
1940     \caption@KOMA@@setcapmargin[#1]{#2}\caption@setcapmargin}

1941 \newcommand*\caption@setcapmargin{%
1942     \begingroup
1943     \let\onelinecaptionsfalse\relax
1944     \def\@twoside{0}%
1945     \def\if@twoside{\def\@twoside{1}\iffalse}%
1946     \cap@margin
1947     \def\@tempa{\endgroup}%
1948     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1949         \def\hspace##1##{\@firstofone}%
1950         \edef\@tempa{\endgroup
1951             \noexpand\captionsetup{%
1952                 twoside=\@twoside,slc=0,%
1953                 margin={\cap@left,\cap@right}}}%
1954     \fi\fi
1955     \@tempa}

1956 \ifx\cap@margin\relax \else
1957     \caption@setcapmargin
1958 \fi

1959 }

```

2.8 Processing of options

```
1960 \caption@ProcessOptions*{caption}
```

2.9 \captionof and \captionlistentry

```

1961 \caption@AtBeginDocument{%
1962     \DeclareCaptionOption{type}{\caption@settype{#1}}%
1963     \DeclareCaptionOption{type*}{\caption@settype*{#1}}%
1964     \DeclareCaptionOption{subtype}[sub\@capytype]{\caption@setsubtype{#1}}%
1965     \DeclareCaptionOption{subtype*}[sub\@capytype]{\caption@setsubtype*{#1}}%
1966 }

```

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

```

\caption@settype \caption@settype*{<type>}
sets \@capttype and executes the options associated with it (using \caption@setoptions).
Furthermore we check \currentgrouplevel (if avail), redefine \@currentlabel
so a \label before \caption will result in a hint instead of a wrong reference, and
use the macro \caption@(sub)typehook (which will be used by our float package
support).
The non-starred version sets a hyperref anchor additionally (if hypcap=true and the
hypcap package is not loaded).
1967 \newcommand*\caption@settype{%
1968   \caption@@settype{}}
1969 \newcommand*\caption@setsubtype{%
1970   \caption@iftype
1971     {\caption@@settype{sub}}%
1972     {\caption@Error{Option 'subtype=' outside float}}}%
1973 \newcommand*\caption@@settype[1]{%
1974   \caption@teststar{\caption@@@settype{#1}}\@firstoftwo\@secondoftwo}
1975 \newcommand*\caption@@@settype[3]{%
1976   % #1 = "" or "sub"
1977   % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1978   % #3 = <type>, e.g. "figure" or "table"
1979   \@ifundefined{c@#3}%
1980     {\caption@Error{No float type '#3' defined}}%
1981     {\caption@Debug{#1type=#3}}%
1982     \caption@checkgrouplevel{#1}{%
1983       \captionsetup{#1type#2*\@empty=...}#2{ or
1984         \@backslashchar#1captionof}{}}%
1985     \edef\caption@tempa{#3}%
1986     \expandafter\ifx\csname @#1capttype\endcsname\caption@tempa \else
1987       \ifcaptionsetup@star\else\@nameuse{caption@#1type@warning}\fi
1988       \fi
1989     \expandafter\let\csname @#1capttype\endcsname\caption@tempa
1990     \@nameuse{caption@#1typehook}%
1991     \caption@setoptions{#3}%
1992     \ifx\caption@opt\relax
1993       \@nameundef{caption@#1type@warning}%
1994     \else
1995       \@namedef{caption@#1type@warning}{\caption@Warning{%
1996         The #1caption type was already set to
1997         '\csname @#1capttype\endcsname'\MessageBreak}}%
1998       \fi
1999       \let\caption@ifrefstepcounter\@secondoftwo
2000       #2{}%
2001       \let\@currentlabel\caption@undefinedlabel
2002       % \let\@currentHlabel\@undefined
2003       \ifx\caption@ORI@label\@undefined
2004         \let\caption@ORI@label\label
2005         \let\label\caption@xlabel

```

```

2006         \fi
2007         \caption@start}}}}

\caption@typehook Hook, will be extended later on, e.g. by our float package support.
2008 \newcommand*\caption@typehook{}

\caption@iftype Since we often need to check if \@cptype is defined (means: we are inside a floating
environment) this helper macro was introduced.
2009 \newcommand*\caption@iftype{%
2010   \@ifundefined{@cptype}{\let\@cptype\@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  $\epsilon$ -TeX)
2011 \begingroup\expandafter\expandafter\expandafter\endgroup
2012 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
2013   \caption@Debug{TeX engine: TeX}
2014   \let\caption@checkgrouplevel\@gobbletwo
2015 \else
2016   \caption@Debug{TeX engine: e-TeX}
2017   \newcommand*\caption@checkgrouplevel[2]{%
2018     \@ifundefined{#1caption@grouplevel}{\let\caption@grouplevel\z@}{}%
2019     \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}{}%
2020     \ifnum\currentgrouplevel>\caption@grouplevel\relax
2021       \expandafter\edef\csname #1caption@grouplevel\endcsname{%
2022         \the\currentgrouplevel}%
2023     \else
2024       \caption@Warning{\string#2\MessageBreak outside box or environment}%
2025     \fi
2026   }{}}
2027 \fi

\caption@undefinedlabel This label will be used for \currentlabel inside (floating) environments as default.
(see above)
2028 \newcommand*\caption@undefinedlabel{%
2029   \protect\caption@xref{\caption@labelname}{\on@line}}
2030 \DeclareRobustCommand*\caption@xref[2]{%
2031   \caption@WarningNoLine{\noexpand\label before \string\caption#2}%
2032   \@setref\relax\@undefined{#1}}
2033 \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.
2034 \newcommand*\caption@xlabel[1]{%
2035   \caption@@xlabel
2036   \def\caption@labelname{#1}%
2037   \caption@ORI@label{#1}}
2038 \newcommand*\caption@@xlabel{%
2039   \global\let\caption@@xlabel\@empty
2040   \@bsphack
2041   \protected@write\@auxout{}%
2042     {\string\providecommand*\string\caption@xref[2]{%

```

```

2043         \string\@setref\string\relax\string\@undefined{\string##1}}}%
2044     \@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)

2045 \caption@AtBeginDocument{%
2046     \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}
2047 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

\captionlistentry \captionlistentry[<float type>]{<list entry>}
\captionlistentry* [<float type>]{<list entry>}
2048 \newcommand*\captionlistentry{%
2049     \caption@teststar\captionlistentry\@firstoftwo\@secondoftwo}

2050 \newcommand*\@captionlistentry[1]{%
2051     \@testopt{\caption@listentry{#1}}{\@capttype}

2052 \def\caption@listentry#1[#2]#3{%
2053     \@bsphack
2054     #1{\caption@getttitle{#3}}%
2055     {\caption@refstepcounter{#2}%
2056     \caption@makecurrent{#2}{#3}}%
2057     \caption@addcontentsline{#2}{#3}%
2058     \@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

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

will be used instead of

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

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

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

```

2059 \def\ContinuedFloat{%
2060     \ifnextchar[\@Continued@Float\@ContinuedFloat}

```



```

2061 \def \@Continued@Float[#1]{\addtocounter{#1}\m@ne}
2062 \def \@Continued@Float{%
2063   \caption@iftype
2064     {\addtocounter\@c@type\m@ne
2065     \caption@Continued@Float\@c@type}%
2066     {\caption@Error{\noexpand\Continued@Float outside float}}}}
2067 \def \caption@Continued@Float#1{%
2068   \ifstar{\caption@Continued@Float@{#1}}{\caption@Continued@Float{#1}}}
2069 \def \caption@Continued@Float@{%
2070   \addtocounter\@c@type\@ne
2071   \@stpelt{Continued@Float}\stepcounter{Continued@Float}%
2072   \def \caption@resetContinued@Float##1{\xdef \caption@CFtype{##1}}%
2073   \caption@@Continued@Float}
2074 \def \caption@Continued@Float#1{%
2075   \edef \caption@tempa{#1}%
2076   \ifx \caption@tempa \caption@CFtype
2077     \stepcounter{Continued@Float}%
2078     \let \caption@resetContinued@Float \gobble
2079     \caption@@Continued@Float{#1}%
2080     \sf@Continued@Float{#1}%
2081   \else
2082     \caption@Error{Continued `#1' after `\'caption@CFtype'}%
2083   \fi}
2084 \def \caption@@Continued@Float#1{%
2085   \expandafter \l@addto@macro \csname the#1\endcsname \theContinued@Float
2086   \@ifundefined{theH#1}{}{}%
2087   \expandafter \l@addto@macro \csname theH#1\endcsname {%
2088     \@alph@c@Continued@Float}%
2089   \caption@setoptions{Continued@Float}%
2090   \caption@setoptions{continued#1}}
2091 \providecommand*\sf@Continued@Float[1]{}
2092 \newcommand*\caption@CFtype{??}

```

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

```

2093 \newcounter{Continued@Float}
2094 \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`.)

```

2095 \newcommand*\caption@resetContinued@Float[1]{%
2096   \@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.

```

2097 \newcommand*\caption@refstepcounter[1]{%
2098   \caption@resetContinued@Float{#1}%

```

```

2099 \caption@@refstepcounter{#1}%
2100 \let\caption@ifrefstepcounter\@firstoftwo}

2101 \newcommand*\caption@@refstepcounter{\refstepcounter}
2102 \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[]{}).

2103 \@ifundefined{kernel@ifnextchar}%
2104 {\newcommand\caption@dblarg[1]{\@ifnextchar[{\#1}{\caption@xdblarg{\#1}}}%
2105 {\newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{\#1}}}%
2106 \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)

2107 \newcommand*\caption@begin[1]{%
2108   \begingroup
2109   \caption@setfnum{\#1}%
2110   \caption@fixposition
2111   \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>.

2112 \newcommand\caption@beginex[3]{%
2113   \caption@begin{\#1}%
2114   \caption@addcontentsline{\#1}{\#2}%
2115   \caption@ifempty{\#3}{}

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

2116 \newcommand*\caption@end{%
2117   \endgroup
2118   \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.

2119 \newcommand*\caption@setfnum[1]{%
2120   \@ifundefined{fnum{\#1}}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
2121   \@namedef{fnum{\#1}}{\caption@fnum{\#1}}%
2122   \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`.

```
2123 \newcommand*\caption@boxrestore{%
2124   \let\if@nobreak\iffalse
2125   \let\if@noskipsec\iffalse
2126   \let\par\@par
2127 % \let\-\@dischyph
2128 % \let'\@acci\let'\@accii\let\=\@acciii
2129   \parindent\z@ \parskip\z@skip
2130   \everypar{}%
2131 % \linewidth\hsize
2132 % \@totalleftmargin\z@
2133   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
2134   \parfillskip\@flushglue \lineskip\normallineskip
2135   \baselineskip\normalbaselineskip
2136   \sloppy
2137   \let\\\@normalcr
2138 }
```

`\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).

```
2139 \newcommand*\caption@normalsize{%
2140   \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.**

```
2141 \let\caption@setfloatcapt\@firstofone
```

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

```
\caption@makeanchor 2142 \newcommand*\caption@makecurrent[2]{}
\caption@start      2143 \let\caption@makeanchor\@firstofone
\caption@@start
\caption@freezeHref
\caption@defrostHref
```

```

2144 \let\caption@start\relax
2145 \let\caption@@start\relax
2146 \let\caption@freezeHref\relax
2147 \let\caption@defrostHref\relax

```

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

```

2148 \newcommand\caption@getttitle[1]{%
2149   \@ifundefined{NR@getttitle}%
2150     {\def\@currentlabelname{#1}}%
2151     {\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.

```

2152 \def\caption@caption{%
2153   \caption@iftype
2154     {\caption@checkgrouplevel\@empty\caption
2155      \caption@star
2156       {\caption@refstepcounter\@capttype}%
2157       {\caption@dblarg{\@caption\@capttype}}}%
2158     {\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.

```

2159 \newcommand*\caption@star[2]{%
2160   \@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](#).)

```

2161 \long\def\caption@@caption#1[#2]#3{%
2162   \ifcaption@star \else
2163     \caption@prepareanchor{#1}{#2}%
2164     \fi
2165   \par
2166   \caption@beginex{#1}{#2}{#3}%
2167   \caption@setfloatcapt{%
2168     \caption@boxrestore
2169     \if@minipage
2170       \@setminipage
2171     \fi
2172     \caption@normalsize
2173     \ifcaption@star
2174       \let\caption@makeanchor\@firstofone
2175     \fi

```

```

2176 \makecaption{\csname fnum@#1\endcsname}%
2177 {\ignorespaces\caption@makeanchor{#3}}\par
2178 \caption@if@minipage{\@minipagetrue\@minipagefalse}%
2179 \caption@end}%

```

`\caption@prepareanchor`

```

2180 \newcommand*\caption@prepareanchor[2]{%
2181 \caption@makecurrent{#1}{#2}%
2182 \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.

```

2183 \long\def\caption@makecaption#1#2{%
2184 \caption@iftop
2185 {\vskip\belowcaptionskip}%
2186 {\caption@rule\vskip\abovecaptionskip}%
2187 \caption@@make{#1}{#2}%
2188 \caption@iftop
2189 {\vskip\abovecaptionskip\caption@rule}%
2190 {\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.

```

2191 \newcommand*\caption@redefine{}
2192 \g@addto@macro\caption@redefine{%
2193 \caption@setbool{incompatible}{0}%
2194 \caption@CheckCommand\caption{%
2195 % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2196 \def\caption{%
2197 \ifx\@capttype\@undefined
2198 \@latex@error{\noexpand\caption outside float}\@ehd
2199 \expandafter\@gobble
2200 \else
2201 \refstepcounter\@capttype
2202 \expandafter\@firstofone
2203 \fi
2204 {\@dblarg{\@caption\@capttype}}%
2205 }}%
2206 \caption@CheckCommand\caption{%
2207 % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
2208 \def\caption{
2209 \ifx\@capttype\@undefined
2210 \@latex@error{\noexpand\caption outside figure or table}\@ehd
2211 \expandafter\@gobble
2212 \else
2213 \refstepcounter\@capttype
2214 \expandafter\@firstofone

```

```

2215     \fi
2216     {\@dblarg{\@caption\@capttype}}%
2217   }}%

2218 \caption@CheckCommand\caption{%
2219   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2220   \renewcommand\caption{%
2221     \ifx\@capttype\@undefined
2222       \@latex@error{\noexpand\caption outside float}\@ehd
2223       \expandafter\@gobble
2224     \else
2225       \refstepcounter\@capttype
2226       \let\@tempf\@caption
2227       \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
2228         \expandafter\expandafter\let
2229         \expandafter\@tempf\csname @float@c@\@capttype\endcsname
2230       \fi
2231       \@dblarg{\@tempf\@capttype}}}%

2233 \caption@CheckCommand\caption{%
2234   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2235   % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2236   % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2237   \def\caption{%
2238     \ifx\@capttype\@undefined
2239       \@latex@error{\noexpand\caption outside float}\@ehd
2240       \expandafter\@gobble
2241     \else
2242       \H@refstepcounter\@capttype
2243       \@ifundefined{fst@\@capttype}{%
2244         \let\Hy@tempa\@caption
2245       }{%
2246         \let\Hy@tempa\Hy@float@caption
2247       }%
2248       \expandafter\@firstofone
2249       \fi
2250       {\@dblarg{\Hy@tempa\@capttype}}%
2251   }}%

2252 \caption@CheckCommand\caption{%
2253   % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2254   \def\caption{%
2255     \ifx\@capttype\@undefined
2256       \@latex@error{\noexpand\caption outside float}\@ehd
2257       \expandafter\@gobble
2258     \else
2259       \H@refstepcounter\@capttype
2260       \let\Hy@tempa\@caption
2261       \@ifundefined{float@caption}{%
2262       }{%
2263         \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
2264         \let\Hy@tempa\Hy@float@caption
2265       \fi
2266       }%
2267       \expandafter\@firstofone

```

```

2268     \fi
2269     {\@dblarg{\Hy@tempa\@captive}}}%
2270   } }%

2271 \caption@ifcheckcommand{}{%
2272   \caption@info{%
2273     Incompatible package detected (regarding \string\caption).\MessageBreak
2274     \string\caption\space=\space\meaning\caption}%
2275   \caption@setbool{incompatible}{1}}%

2276 \caption@checkcommand\@caption{%
2277   % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2278   \long\def\@caption#1[#2]#3{%
2279     \par
2280     \addcontentsline{\csname ext@#1\endcsname}{#1}%
2281     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
2282     \begingroup
2283       \@parboxrestore
2284       \if@minipage
2285         \@setminipage
2286       \fi
2287       \normalsize
2288       \makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2289     \endgroup}}%

2290 \caption@checkcommand\@caption{%
2291   % beamerbaselocalstructure.sty, v 1.53 2007/01/28 20:48:21 tantau
2292   \long\def\@caption#1[#2]#3{% second argument ignored
2293     \par\nobreak
2294     \begingroup
2295       \@parboxrestore
2296       \if@minipage
2297         \@setminipage
2298       \fi
2299       \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
2300     \endgroup}}%

2301 % \caption@checkcommand\float@caption{%
2302 %   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2303 %   \long\def\float@caption#1[#2]#3{%
2304 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%
2305 %     {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
2306 %     \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
2307 %     \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
2308 %     \ifnextchar[{\float@ccon}{\egroup}}%
2309 %   \long\def\float@ccon[#1]{#1\par\egroup}}%

2310 \caption@checkcommand\@caption{%
2311   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2312   \long\def\@caption#1[#2]#3{%
2313     \hyper@makecurrent{\@captive}%
2314     \def\@currentlabelname{#2}%
2315     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2316       \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
2317     }%
2318     \begingroup
2319       \@parboxrestore

```

```

2320         \if@minipage
2321             \@setminipage
2322         \fi
2323         \normalsize
2324         \@makecaption{\csname fnum@#1\endcsname}{%
2325             \ignorespaces
2326             \ifHy@nesting
2327                 \hyper@@anchor{\@currentHref}{#3}%
2328             \else
2329                 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2330             \fi
2331         }%
2332         \par
2333     \endgroup
2334 }}%

2335 \caption@CheckCommand\@caption{%
2336     % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2337     % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2338     % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2339     \long\def\@caption#1[#2]#3{%
2340         \expandafter\ifx\csname if@capstart\expandafter\endcsname
2341             \csname iftrue\endcsname
2342         \global\let\@currentHref\hc@currentHref
2343     \else
2344         \hyper@makecurrent{\@captype}%
2345     \fi
2346     \def\@currentlabelname{#2}%
2347     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2348         \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2349     }%
2350     \begingroup
2351         \@parboxrestore
2352         \if@minipage
2353             \@setminipage
2354         \fi
2355         \normalsize
2356         \expandafter\ifx\csname if@capstart\expandafter\endcsname
2357             \csname iftrue\endcsname
2358         \global\@capstartfalse
2359         \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2360     \else
2361         \@makecaption{\csname fnum@#1\endcsname}{%
2362             \ignorespaces
2363             \ifHy@nesting
2364                 \hyper@@anchor{\@currentHref}{#3}%
2365             \else
2366                 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2367             \fi
2368         }%
2369     \fi
2370     \par
2371 \endgroup
2372 }}%

```



```

2373 \caption@CheckCommand\@caption{%
2374   % hyperref.sty [2009/11/27 v6.79k Hypertext links for LaTeX]
2375   \long\def\@caption#1[#2]#3{%
2376     \expandafter\ifx\csname if@capstart\expandafter\endcsname
2377       \csname iftrue\endcsname
2378     \global\let\@currentHref\hc@currentHref
2379   \else
2380     \hyper@makecurrent{\@capttype}%
2381   \fi
2382   \def\@currentlabelname{#2}%
2383   \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2384     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2385   }%
2386   \begingroup
2387     \@parboxrestore
2388     \if@minipage
2389       \@setminipage
2390     \fi
2391     \normalsize
2392     \expandafter\ifx\csname if@capstart\expandafter\endcsname
2393       \csname iftrue\endcsname
2394     \global\@capstartfalse
2395     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2396   \else
2397     \@makecaption{\csname fnum@#1\endcsname}{%
2398       \ignorespaces
2399       \ifHy@nesting
2400         \expandafter\hyper@@anchor\expandafter{\@currentHref}{#3}%
2401       \else
2402         \Hy@raisedlink{%
2403           \expandafter\hyper@@anchor\expandafter{\@currentHref}{\relax}%
2404         }%
2405       #3%
2406     \fi
2407   }%
2408   \fi
2409   \par
2410 \endgroup
2411 }}%

2412 \caption@CheckCommand\@caption{%
2413   % hyperref.sty [2009/12/09 v6.79m Hypertext links for LaTeX]
2414   % hyperref.sty [2009/12/28 v6.79z Hypertext links for LaTeX]
2415   \long\def\@caption#1[#2]#3{%
2416     \expandafter\ifx\csname if@capstart\expandafter\endcsname
2417       \csname iftrue\endcsname
2418     \global\let\@currentHref\hc@currentHref
2419   \else
2420     \hyper@makecurrent{\@capttype}%
2421   \fi
2422   \@ifundefined{NR@getttitle}{%
2423     \def\@currentlabelname{#2}%
2424   }{%
2425     \NR@getttitle{#2}%
2426   }%

```

```

2427 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2428 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2429 }%
2430 \begingroup
2431 \let\parboxrestore
2432 \if@minipage
2433 \let\setminipage
2434 \fi
2435 \normalsize
2436 \expandafter\ifx\csname if@capstart\expandafter\endcsname
2437 \csname iftrue\endcsname
2438 \global\@capstartfalse
2439 \let\makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2440 \else
2441 \let\makecaption{\csname fnum@#1\endcsname}{%
2442 \ignorespaces
2443 \ifHy@nesting
2444 \expandafter\hyper@@anchor\expandafter{\@currentHref}{#3}%
2445 \else
2446 \Hy@raisedlink{%
2447 \expandafter\hyper@@anchor\expandafter{%
2448 \@currentHref
2449 }{\relax}%
2450 }%
2451 #3%
2452 \fi
2453 }%
2454 \fi
2455 \par
2456 \endgroup
2457 }}%

2458 \caption@CheckCommand\@caption{%
2459 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
2460 \long\def\@caption#1[#2]{%
2461 \def\@currentlabelname{#2}%
2462 \NR@@caption{#1}[{#2}]%
2463 }}%

2464 \caption@CheckCommand\@caption{%
2465 % nameref.sty [2009/11/27 v2.32 Cross-referencing by name of section]
2466 \long\def\@caption#1[#2]{%
2467 \NR@getttitle{#2}%
2468 \NR@@caption{#1}[{#2}]%
2469 }}%

2470 \caption@CheckCommand\@caption{%
2471 % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
2472 \long\def\@caption#1[#2]#3{%
2473 \ifundefined{if#1topcap}%
2474 {\subfig@oldcaption{#1}[{#2}]{#3}}%
2475 {\@nameuse{if#1topcap}%
2476 \@listsubcaptions{#1}%
2477 \subfig@oldcaption{#1}[{#2}]{#3}%
2478 \else
2479 \subfig@oldcaption{#1}[{#2}]{#3}%

```

```

2480         \@listsubcaptions{#1}%
2481         \fi}}}%

2482 \caption@CheckCommand\@caption{%
2483 % subfig.sty [2005/06/28 ver: 1.3 subfig package]
2484 \def\@caption{\caption@}%
2485 % \long\def\caption@#1[#2]#3{%
2486 % \ifundefined{caption@setfloattype}%
2487 % \caption@settype
2488 % \caption@setfloattype
2489 % \@captype
2490 % \sf@ifpositiontop{%
2491 % \@listsubcaptions{#1}%
2492 % \sf@old@caption{#1}[#{2}]{#3}%
2493 % }{%
2494 % \sf@old@caption{#1}[#{2}]{#3}%
2495 % \@listsubcaptions{#1}%
2496 % }}%
2497 }%

2498 \caption@IfCheckCommand{}{%
2499 \caption@Info{%
2500 Incompatible package detected (regarding \string\@caption).\MessageBreak
2501 \string\@caption\space=\space\meaning\@caption}%
2502 \caption@setbool{incompatible}{1}}%

The option compatibility= will override the compatibility mode.

2503 \ifundefined{caption@ifcompatibility}%
2504 {\let\caption@ifcompatibility\caption@ifincompatible
2505 \let\caption@tempa\caption@WarningNoLine}%
2506 {\let\caption@tempa\@gobble}% suppress warning

2507 \caption@ifcompatibility{%
2508 \caption@tempa{%
2509 \noexpand\caption will not be redefined since it's already\MessageBreak
2510 redefined by a document class or package which is\MessageBreak
2511 unknown to the caption package}%
2512 \renewcommand*\caption@redefine{}}%

\ContinuedFloat is not supported in compatibility mode.

2513 \renewcommand*\caption@ContinuedFloat[1]{%
2514 \caption@Error{Not available in compatibility mode}}%

\caption@start is not supported in compatibility mode.

2515 \caption@AtBeginDocument*{%
2516 \let\caption@start\relax
2517 \ifundefined{caption@ORI@capstart}}{%
2518 \caption@Debug{%
2519 Restore hypcap definition of \string\capstart\@gobble}%
2520 \let\capstart\caption@ORI@capstart}%
2521 \ifundefined{caption@ORI@float@makebox}}{%
2522 \caption@Debug{%
2523 Restore hyperref redefinition of \string\float@makebox\@gobble}%
2524 \let\float@makebox\caption@ORI@float@makebox}%
2525 }%

```

```

\caption@star We redefine \caption@star here so it does not make any harm.
2526     \renewcommand*\caption@star[2]{#1#2}%

2527 }{%
2528     \caption@ifincompatible{%
2529         \caption@WarningNoLine{%
2530             Forced redefinition of \noexpand\caption since the\MessageBreak
2531             unsupported(!) package option 'compatibility=false'\MessageBreak
2532             was given}%
2533     }{}%

\caption
\@caption 2534     \renewcommand*\caption@redefine{%
2535         \let\caption\caption@caption
2536         \let\@caption\caption@@caption}%
2537     \caption@redefine

2538 }%
2539 \caption@AtBeginDocument*{%
2540     \let\caption@ORI@capstart\@undefined
2541     \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.
2542 \let\caption@ORI@xfloat\@xfloat
2543 \def\@xfloat#1[#2]{%
2544     \caption@ORI@xfloat{#1}[#2]%
2545     \caption@settype{#1}}%

2546 }

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.
2547 \caption@AtBeginDocument{\caption@redefine}

\@makecaption
2548 \let\@makecaption\caption@makecaption

```

2.13 Support for sub-captions

```

\caption@DeclareSubType \caption@DeclareSub initializes the usage of \caption in sub-floats.
2549 \def\caption@DeclareSubType sub#1\@nil{%
2550     \caption@Debug{Initializing subtype for `#1'\@gobble}%
2551     \@namedef{caption@c@#1}{0}%
2552     \@namedef{caption@beginsub#1}{\caption@beginsubfloat{#1}}
2553 \@onlypreamble\caption@DeclareSubType

Initialize the sub-captions defined with \DeclareCaptionSubType...
2554 \caption@For*{subtypelist}{\caption@DeclareSubType sub#1\@nil}

```

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

```
2555 \caption@AtBeginDocument*{%
2556   \@ifundefined{sf@counterlist}{}{}%
2557   \@for\sftemp:=\sf@counterlist\do{%
2558     \expandafter\caption@DeclareSubType\sftemp\@nil}}}
```

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

```
2559 \newcommand*\caption@subtyphook{%
2560   \ifx\caption\caption@subcaption \else
2561     \caption@ifrefstepcounter{}{}%
2562     % no \caption or \subcaption in this (floating) environment yet
2563     \caption@Debug{Increment \@c@type\ counter =\the\value\@c@type}%
2564     \caption@l@stepcounter\@c@type
2565     \let\addcontentsline\caption@addsubcontentsline}%
2566   \ifnum\csname caption@c@\@c@type\endcsname=\value\@c@type \else
2567     \caption@Debug{Reset sub\@c@type\ counter}%
2568     \expandafter\xdef\csname caption@c@\@c@type\endcsname{%
2569       \the\value\@c@type}%
2570     \@stpel\@subcaption
2571   \fi
2572   \c@ContinuedFloat=0\relax
2573   \let\caption@resetContinuedFloat\@gobble
2574   \let\caption@addcontentsline\caption@kernel@addcontentsline
2575   \let\caption@setfloatcapt\@firstofone
2576   \caption@clearmargin
2577   \caption@iflist{}{}{\let\caption@setlist\@gobble}%
2578   \caption@setoptions{sub}%
2579   \caption@setoptions{subfloat}% for subfig-package compatibility
2580   \let\caption\caption@subcaption
2581   \let\@makecaption\caption@makecaption
2582 \fi}%
```

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

```
2583 \newcommand*\caption@subcaption{%
2584   \caption@iftype
2585     {\caption@checkgrouplevel{sub}\subcaption
2586     \caption@star
2587     {\caption@refstepcounter\@subcaption}%
2588     {\caption@dblarg{\@caption\@subcaption}}}%
2589     {\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.

```
2590 \let\caption@kernel@addcontentsline\caption@addcontentsline
2591 \renewcommand*\caption@addcontentsline[2]{%
2592   \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
2593   \caption@kernel@addcontentsline{#1}{#2}%
2594   \sf@ifpositiontop{}{\@listsubcaptions{#1}}%
2595   \caption@addsubcontentslines{#1}}
```

```

2596 \newcommand*\caption@addsubcontentslines[1]{%
2597   \begingroup
2598     \caption@subcontentslines
2599   \endgroup
2600   \caption@clearsubcontentslines}%

2601 \caption@AtBeginDocument*{%
2602   \@ifundefined{sf@ifpositiontop}{\let\sf@ifpositiontop@gobbletwo}{}%
2603   \caption@clearsubcontentslines
2604   \g@addto@macro\caption@typehook{\caption@checksubcontentslines}%
2605   \AtEndDocument{\caption@checksubcontentslines}}%

caption@addsubcontentsline Add a pending sub-caption list entry.
2606 \newcommand*\caption@addsubcontentsline[3]{%
2607   \begingroup
2608   \let\label\@gobble \let\index\@gobble \let\glossary\@gobble
2609   \protected@edef\@tempa{\endgroup
2610     \noexpand\g@addto@macro\noexpand\caption@subcontentslines{%
2611       \noexpand\@namedef{the#2}{\csname the#2\endcsname}%
2612       \ifx\@currentHref\@undefined \else
2613         \noexpand\def\noexpand\@currentHref{\@currentHref}%
2614       \fi
2615       \protect\addcontentsline{#1}{#2}{#3}}}%
2616   \@tempa}

ion@checksubcontentslines Checks if the list of pending sub-captions is empty, if not, a warning will be issued.
2617 \newcommand*\caption@checksubcontentslines{%
2618   \ifx\caption@subcontentslines\@empty \else
2619     \caption@Error{%
2620       Something's wrong--perhaps a missing \protect\caption\MessageBreak
2621       in the last figure or table}%
2622     \caption@clearsubcontentslines
2623   \fi}

ion@clearsubcontentslines Clear pending sub-caption list entries.
2624 \newcommand*\caption@clearsubcontentslines{%
2625   \global\let\caption@subcontentslines\@empty}

```

2.14 Document class & Babel package support

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

```
2626 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}
```

2.14.2 The beamer class

```
2627 \@ifclassloaded{beamer}{%
2628   \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.

```

2629   \captionsetup{list=false}
2630   \DeclareCaptionOption{list}[1]{}
2631   \DeclareCaptionOption{listof}[1]{}

```

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

```

2632 \expandafter\let\expandafter\caption@ORI@figure
2633 \csname\string\figure\endcsname
2634 \@namedef{\string\figure}[#1]{%
2635 \caption@ORI@figure[#1]%
2636 \caption@settype{figure}}
2637 \expandafter\let\expandafter\caption@ORI@table
2638 \csname\string\table\endcsname
2639 \@namedef{\string\table}[#1]{%
2640 \caption@ORI@table[#1]%
2641 \caption@settype{table}}
2642 {}

```

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.

```

2643 \@ifundefined{scr@caption}{}{%
2644 \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)

```

2645 \@nameuse{caption@frenchb}\@nameundef{caption@frenchb}

```

2.14.5 The frenchle/pro package

```

2646 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
2647 \caption@Info{frenchle/pro package is loaded}%
2648 \let\captionfont@ORI\captionfont
2649 \let\captionlabelfont@ORI\captionlabelfont
2650 \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`.

```

2651 \@ifundefined{GOfrench}%
2652 {\let\caption@tempa\@firstofone}%
2653 {\def\caption@tempa{\g@addto@macro\GOfrench}}%
2654 \caption@tempa{%
2655 \let\captionfont\captionfont@ORI
2656 \let\captionfont@ORI\@undefined
2657 \let\captionlabelfont\captionlabelfont@ORI
2658 \let\captionlabelfont@ORI\@undefined
2659 \let\@makecaption\@makecaption@ORI
2660 \let\@makecaption@ORI\@undefined

```

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

```

2661 \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.

2662 \let\caption@tcORI\@tablescaption
2663 \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.

2664 \let\@eatDP\@undefined
2665 \let\caption@tempa\@empty
2666 \ifx\ffrench\figure
2667 \l@addto@macro\caption@tempa{\let\fnum@figure\ffrench}%
2668 \fi
2669 \ifx\tfrench\table
2670 \l@addto@macro\caption@tempa{\let\fnum@table\tfrench}%
2671 \fi
2672 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
2673 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
2674 \caption@tempa

2675 }%
2676 }

```

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.

2677 \newcommand\caption@ifpackageloaded[1]{%
2678 \testopt{\caption@ifpackageloaded{#1}}{}%
2679 \@onlypreamble\caption@ifpackageloaded

2680 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
2681 \ifpackageloaded{#1}\@firstofone{%
2682 \caption@Debug{#1 package is not loaded (yet)}\@gobble}%
2683 \caption@AtBeginDocument}{%
2684 \caption@ifpackageloaded{#1}[#2]{#3}{#4}}%
2685 \@onlypreamble\caption@ifpackageloaded

2686 \newcommand\caption@ifpackageloaded[1]{%
2687 \testopt{\caption@ifpackageloaded{#1}}{}%
2688 \@onlypreamble\caption@ifpackageloaded

2689 \long\def\caption@ifpackageloaded#1[#2]{%
2690 \ifpackageloaded{#1}{%
2691 \caption@Info{#1 package is loaded}%
2692 \ifpackagelater{#1}[#2]\@firstoftwo{%
2693 \caption@Error{%
2694 For a successful cooperation we need at least version\MessageBreak
2695 `#2' of package #1,\MessageBreak
2696 but only version\MessageBreak

```



```

2697         '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
2698         is available}%
2699         \@secondoftwo}%
2700     }\@secondoftwo}}
2701 \@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.

```

2702 \newcommand*\caption@clearmargin{%
2703     \setcaptionmargin\z@
2704     \let\caption@minmargin\@undefined}

2705 \caption@setbool{needfreeze}{0}
2706 \caption@AtBeginDocument*{%
2707     \caption@ifneedfreeze{%

```

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

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

```

2708 \newcommand*\caption@freeze{%
2709     \caption@teststar\caption@@freeze\@gobble\@firstofone}%

2710 \newcommand*\caption@@freeze[1]{%
2711     \global\let\caption@SCcontinued\relax
2712     \global\let\caption@SCsetup\@undefined
2713     \global\let\caption@SClentry\@undefined
2714     \global\let\caption@SCtext\@undefined
2715     \global\let\caption@SClabel\@undefined

2716     \let\caption@ORI@ContinuedFloat\ContinuedFloat
2717     \def\ContinuedFloat{%
2718         \caption@withoptargs\caption@SC@ContinuedFloat}%
2719     \def\caption@SC@ContinuedFloat##1{%
2720         \let\caption@ORI@setcounter\setcounter
2721         \let\caption@ORI@addtocounter\addtocounter
2722         \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
2723         \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
2724         \caption@ORI@ContinuedFloat##1%
2725         \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
2726         \let\setcounter\caption@ORI@setcounter
2727         \let\addtocounter\caption@ORI@addtocounter}%
2728     \let\caption@ORI@setup\captionsetup
2729     \def\captionsetup{%
2730         \caption@withoptargs\caption@SC@setup}%
2731     \def\caption@SC@setup##1##2{%
2732         \caption@g@addto@list\caption@SCsetup{##2}%
2733         \caption@ORI@setup##1{##2}}%
2734     \let\caption@ORI\caption
2735     \def\caption{%
2736         \def\caption{\caption@Error{%
2737             Only one \noexpand\caption can be placed in this environment}}%
2738         \let\captionsetup\caption@setup
2739         \let\caption@@refstepcounter\caption@l@stepcounter
2740         \caption@ORI}%
2741     \long\def\@caption##1[##2]##3{%
2742         \@bsphack

```

```

2743         \gdef\caption@SClentry{##2}%
2744         \gdef\caption@SCTest{##3}%
2745         \@esphack}%
2746     #1{% is \@gobble in star form, and \@firstofone otherwise
2747         \def\label##1{\@bsphack\gdef\caption@SCLabel{##1}\@esphack}}%
2748 }%

\caption@defrost \caption@defrost
2749 \newcommand*\caption@defrost{%
2750     \ifx\caption@ORI@ContinuedFloat\@undefined
2751         \caption@defrost@setup
2752     \ifx\caption@SCTest\@undefined \else
2753         \expandafter\expandafter\expandafter\caption
2754         \expandafter\expandafter\expandafter[%
2755         \expandafter\expandafter\expandafter{%
2756         \expandafter\caption@SClentry\expandafter}\expandafter]%
2757         \expandafter{\caption@SCTest}%
2758     \fi
2759     \ifx\caption@SCLabel\@undefined \else
2760         \expandafter\label\expandafter{\caption@SCLabel}%
2761     \fi
2762     \else
2763         \caption@Error{Internal Error:\MessageBreak
2764             \noexpand\caption@defrost in same group as \string\caption@freeze}%
2765     \fi}%

2766 \newcommand*\caption@defrost@setup{%
2767     \caption@SCcontinued
2768     \ifx\caption@SCsetup\@undefined \else
2769         \expandafter\captionsetup\expandafter{\caption@SCsetup}%
2770     \fi}%

2771 }{}%
2772 \caption@undefbool{needfreeze}}

```

2.15.1 The float package

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

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

- The caption will be typeset inside a `\savebox` called `\@floatcapt` so it can be placed above or below the float contents afterwards.
- `\@makecaption` will not be used to finally typeset the caption. Instead `\@fs@capt` will be used which definition is part of the float style. (Note that `\@fs@capt` will not typeset any vertical space above or below the caption; instead this space will be typeset by the float style code itself.)

```

2773 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
2774     \@ifpackageloaded{floatrow}{%
2775         \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
2776     }{}%

```

`\@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>]`.

```
2777 \let\caption@ORI@float@setevery\@float@setevery
2778 \def\@float@setevery#1{%
2779     \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.

```
2780     \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.

```
2781     \renewcommand\caption@setfloatcapt[1]{%
2782         \let\@makecaption\caption@@make
2783         \global\setbox\@floatcapt\vbox{%
2784             \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.

```
2785     \float@getstyle\float@style{#1}%
2786     \caption@setstyle*\float@style
2787     \caption@setoptions\float@style
2788 }{}%
2789 \caption@freezeHref % will be defrosted in \float@makebox
2790 \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.

```
2791 \g@addto@macro\caption@typehook{%
2792     \expandafter\ifx\csname #1name\endcsname\relax
2793         \expandafter\let\csname #1name\expandafter\endcsname
2794             \csname fname@#1\endcsname
2795     \fi}%
```

`\fs@plaintop` 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.

```
2796 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2797 \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`).

```
2798 \providecommand*\float@ifstyle[1]{%
2799   \expandafter\ifx\csname fst@#1\endcsname\relax
2800     \expandafter\@secondoftwo
2801   \else
2802     \expandafter\@firstoftwo
2803   \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.

```
2804 \providecommand*\float@getstyle[2]{%
2805   \edef#1{%
2806     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2807     \expandafter\expandafter\expandafter\noexpand
2808     \csname fst@#2\endcsname}%
2809   \edef#1{#1}%
2810   \caption@Debug{floatstyle{#2} = '#1'}}%
```

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

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

```
2811 \providecommand*\float@setstyle[2]{%
2812   \expandafter\edef\csname fst@#1\endcsname{%
2813     \expandafter\noexpand\csname fs@#2\endcsname}}%
```

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

```
2814 \providecommand*\float@dostyle[1]{%
2815   \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.)

```

2816 \providecommand*\float@ifcaption[1]{%
2817   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2818   \expandafter\@firstoftwo
2819   \else
2820   \expandafter\@secondoftwo
2821   \fi}%

2822 }}{%
2823 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2824 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2825 % \clearcaptionsetup{boxed}% used by the floatrow package?
2826 }

```

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

```
2827 \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’.

```

2828 \caption@ifbool{ruled}{%
2829   \captionsetup[ruled]{margin=0pt,minmargin=0,slc=0}%
2830 }{%
2831   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2832 }
2833 \caption@undefbool{ruled}

```

2.15.2 The floatflt package

```
2834 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

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

```

2835 \let\caption@ORI@floatingfigure\floatingfigure
2836 \def\floatingfigure{%
2837   \caption@floatflt{figure}%
2838   \caption@ORI@floatingfigure}%

```

\floatingtable Same with \floatingtable...

```

2839 \let\caption@ORI@floatingtable\floatingtable
2840 \def\floatingtable{%
2841   \caption@floatflt{table}%
2842 %   \caption@setautoposition b%
2843   \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.

```

2844 \newcommand*\caption@floatflt[1]{%
2845   \caption@settype{#1}%
2846   \caption@clearmargin
2847   \caption@setoptions{floating#1}%
2848   \let\caption@boxrestore\@parboxrestore}%
2849 }{}

```

2.15.3 The fltpage package

```

2850 \caption@ifpackageloaded{fltpage}[1998/10/29 v.0.3]{%
2851   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

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

2852 \renewcommand\FP@helpNote[2]{%
2853   \begingroup % save \caption@thepage
2854     \caption@pageref{#2}%
2855     \typeout{FP#1 is inserted on page \caption@thepage!}%
2856   \endgroup}%

```

`\FP@floatBegin` Original code:

```

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

2857 \renewcommand*\FP@floatBegin[1]{%
2858   \def\@captype{#1}%
2859   \let\FP@LabelText\@empty
2860   \begin{lrbox}{\FP@floatCorpusBOX}%
2861     \caption@ifFPrefcap
2862       {\caption@freeze\relax}%
2863       {\def\label##1{\@bsphack\gdef\FP@LabelText{##1}\@esphack}%
2864         \caption@freeze*}%
2865     \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}}%
  }%

2866 \renewcommand*\FP@floatEnd{%
2867   \end{lrbox}%
2868   \stepcounter{FP@\@capttype C}%
2869   \caption@label\FP@positionLabel
2870   \FP@helpNote\@capttype\FP@positionLabel
2871   \edef\FP@RestoreCounter{%
2872     \noexpand\setcounter{\@capttype}{\the\value\@capttype}%
2873     \noexpand\setcounter{ContinuedFloat}{\the\value{ContinuedFloat}}}%
2874   \FP@float
2875     {\FP@positionLabel}% location label test
2876     {\begin\@capttype[p!]}%
2877       \usebox\FP@floatCorpusBOX
2878       \caption@defrost@setup
2879       \caption@ifFPlistcap
2880         {\caption@refstepcounter\@capttype
2881           \expandafter\caption@makecurrent\expandafter\@capttype
2882             \expandafter{\caption@SCLentry}}}%
2883         {\expandafter\captionlistentry\expandafter{\caption@SCLentry}}}%
2884     \caption@makeanchor\relax
2885     \ifx\FP@LabelText\@empty \else
2886       \expandafter\label\expandafter{\FP@LabelText}%

```

```

2887         \fi
2888         \end\@capttype}%
2889     {\FP@RestoreCounter
2890     \@ifundefined{theH\@capttype}}{%
2891         \expandafter\l@addto@macro\csname theH\@capttype\endcsname{.FP}}}%
2892     {\begin\@capttype[b!]}%
2893         \let\FP@savedSetfnumCommand\caption@setfnum
2894         \def\caption@setfnum##1{%
2895             \FP@savedSetfnumCommand{##1}%
2896             \ifx\FP@guide\empty \else
2897                 \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}}%
2898         \fi}%
2899         \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2900         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2901         \caption@setoptions{FP\@capttype}%
2902         \FP@separatorCaption
2903         \caption@ifFPlistcap{}{\let\caption@addcontentsline\@gobbletwo}%
2904         \caption@defrost
2905     \end\@capttype}%
2906 }%

2907 \caption@For{typelist}{%
2908     \newcounter{FP@#1C}%
2909     \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2910 }{%
2911     \let\caption@ifFPlistcap\@undefined
2912     \let\caption@ifFPrefcap\@undefined
2913 }

```

2.15.4 The hyperref package

```

2914 \caption@IfPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2915     \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2916         \caption@WarningNoLine{%
2917             Hyperref support is turned off\MessageBreak
2918             because hyperref has stopped early}%
2919     }{%
2920         \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`.

```

2921     \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`.

```

2922     \renewcommand*\caption@makecurrent[2]{%
2923         \caption@makecurrentHref{#1}%
2924         \caption@Debug{hyperref current=\@currentHref}%
2925         \caption@getttitle{#2}}%
2926     \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`.

```

2927 \renewcommand\caption@makeanchor[1]{%
2928 \caption@Debug{hyperref anchor: \@currentHref}%
2929 % If we cannot have nesting, the anchor is empty.
2930 \ifHy@nesting
2931 \expandafter\hyper@@anchor\expandafter{\@currentHref}{#1}%
2932 \else
2933 \Hy@raisedlink{%
2934 \expandafter\hyper@@anchor\expandafter{\@currentHref}{\relax}%
2935 }#1%
2936 \fi}%
2937 \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.

```

2938 \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`.

```

2939 \def\caption@start{\caption@ifhycap\caption@start@relax}%
2940 \def\caption@start@{%

```

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

```

2941 \caption@makestart\@captype
2942 \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.)

```

2943 \global\@capstarttrue
2944 \let\hc@currentHref\@currentHref
2945 \def\caption@makecurrentHref##1{%
2946 \global\@capstartfalse
2947 \global\let\@currentHref\hc@currentHref}%

```

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

```

2948 \let\caption@makeanchor\@firstofone
2949 }%

```

`\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.

```
2950 \newcommand*\caption@makestart[1]{%
2951 \begingroup
2952 \Hy@hypertextnamesfalse
2953 % \gdef\@currentHlabel{}%
2954 \hyper@makecurrent{#1.caption}%
2955 \endgroup
2956 \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.

```
2957 \newcommand*\caption@startanchor[1]{%
2958 \ifvmode\begingroup
2959 \caption@Debug{hycap anchor: #1 (vertical mode)}%
2960 \@tempdima\prevdepth
2961 \nointerlineskip
2962 \vspace*{-\caption@hycapSPACE}%
2963 \caption@anchor{#1}%
2964 \vspace*{\caption@hycapSPACE}%
2965 \prevdepth\@tempdima
2966 \endgroup\else
2967 \caption@Debug{hycap anchor: #1 (horizontal mode)}%
2968 \caption@anchor{#1}%
2969 \fi}%

```

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

```
2970 \newcommand*\caption@anchor[1]{%
2971 \ifmeasuring@ \else
2972 \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2973 \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.)

```
2974 \ifx\HyperRaiseLinkLength\@tempdima
2975 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2976 \else
2977 \let\caption@raisedlink\Hy@raisedlink
2978 \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.

```
2979 \def\caption@@start{%

```

```

2980 \ifundefined{hc@currentHref}{%
2981 \caption@Warning{%
2982 The option 'hycap=true' will be ignored for this\MessageBreak
2983 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`.

```

2984 \def\caption@freezeHref{%
2985 \let\caption@ORI@start\caption@start
2986 \def\caption@start{\let\caption@start\caption@ORI@start}%
2987 %
2988 % \l@addto@macro\caption@subtypehook{%
2989 % \let\caption@@start\caption@ORI@@start}%
2990 \global\let\caption@currentHref\@undefined
2991 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2992 \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2993 \renewcommand*\caption@setfloatcapt{%
2994 \ifx\caption@currentHref\@undefined \else
2995 \let\caption@makeanchor\@firstofone
2996 \fi
2997 \caption@ORI@setfloatcapt}}%

```

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

```

2998 \def\caption@defrostHref{%
2999 \ifx\caption@currentHref\@undefined \else
3000 \caption@startanchor\caption@currentHref
3001 \global\let\caption@currentHref\@undefined
3002 \fi}%

```

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

```

3003 \ifundefined{HyOrg@float@makebox}{}{%
3004 \caption@Debug{%
3005 Redefining \noexpand\float@makebox (again)\@gobble}%
3006 \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
3007 \renewcommand\float@makebox[1]{%
3008 \HyOrg@float@makebox{#1\relax \caption@defrostHref}}%
3009 }%
3010 }{}

```

2.15.5 The hycap package

```

3011 \caption@IfPackageLoaded{hycap}{% v1.0
3012 \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.

```
3013 \let\caption@ORI@capstart\capstart % save for compatibility mode
3014 \@ifundefined{capstarttrue}% check for v1.10 of hypcap package
3015 {\def\capstart{\caption@start@}}%
3016 {\def\capstart{\ifcapstart\caption@start@\fi}}%
3017 \let\caption@start\relax
3018 \let\caption@@start\relax
```

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

```
3019 \caption@set@bool\caption@ifhypcap 1%
3020 \renewcommand*\caption@hypcapSPACE{\hypcapSPACE}%
3021 \fi}}
```

2.15.6 The listings package

```
3022 \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.

```
3023 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
3024 \def\lst@MakeCaption#1{% #1 is ‘t’ or ‘b’
3025 \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`.)

```
3026 \caption@setposition{#1}%
3027 \caption@iftop{%
3028 \@tempdima\belowcaptionskip
3029 \belowcaptionskip\abovecaptionskip
3030 \abovecaptionskip\@tempdima}{}}%
```

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

```
3031 \caption@setup{rule=0}%
```

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

```
3032 \caption@setoptions{lstlisting}%
```

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

```
3033 \caption@setautoposition{#1}%
```

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

```
3034 \caption@begin{lstlisting}%
3035 \caption@ORI@lst@MakeCaption{#1}%
3036 \caption@end
3037 \endgroup}%
```

```

\lst@makecaption  Wrapper macros for typesetting the caption= resp. title= value.
\lst@maketitle   3038 \def\lst@makecaption{\caption@starfalse\@makecaption}%
                 3039 \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.
                 3040 \providecommand*\ext@lstlisting{lol}%
                 3041 {}{}

```

2.15.7 The longtable package

```

\LTcapttype  \LTcapttype is preset to table.
              3042 \providecommand*\LTcapttype{table}

              3043 \caption@ifpackageloaded{longtable}[1995/05/24 v3.14]{%
              3044 \RequirePackage{ltcaption}[2007/09/01]%
              3045 \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.

            3046 \caption@AtBeginDocument{%
            3047 \let\caption@ORI@LT@array\LT@array
            3048 \renewcommand*\LT@array{%

\captionsetup for longtable:
            3049 \global\let\caption@opt@@longtable\@undefined
            3050 \def\captionsetup{%
            3051 \noalign\bgroup
            3052 \@ifstar\@captionsetup\@captionsetup}% gobble *
            3053 \def\@captionsetup##1{\LT@captionsetup{##1}\egroup}%
            3054 \def\LT@captionsetup##1{%
            3055 \captionsetup@startrue\caption@setup@options[@longtable]{##1}%
            3056 \global\let\caption@opt@@longtable\caption@opt@@longtable}%

\captionabove & \captionbelow for longtable: (KOMA-Script document class)
            3057 \def\@captionabovetrue{\LT@captionsetup{position=t}}%
            3058 \def\@captionabovetruefalse{\LT@captionsetup{position=b}}%

\captionlistentry for longtable:
            3059 \def\captionlistentry{%
            3060 \noalign\bgroup
            3061 \@ifstar{\egroup\LT@captionlistentry}% gobble *
            3062 {\egroup\LT@captionlistentry}}%
            3063 \def\LT@captionlistentry##1{%
            3064 \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
            3065 % \caption@ifhyperc{}

```

```

3066 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
3067 % \def\hyper@makecurrent##1{%
3068 % \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
3069 % \caption@makestart{##1}%
3070 %% \let\Hy@LT@currentHlabel\@currentHlabel
3071 % \let\Hy@LT@currentHref\@currentHref
3072 % \def\hyper@makecurrent####1{%
3073 % \let\@currentHlabel\Hy@LT@currentHlabel
3074 % \let\@currentHref\Hy@LT@currentHref}}%
3075 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
3076 % \def\ContinuedFloat{\noalign{%
3077 % \gdef\caption@setContinuedFloat{%
3078 % \let\caption@resetContinuedFloat\@gobble}%
3079 % \def\caption@setoptions####1{%
3080 % \g@addto\macro\caption@setContinuedFloat{%
3081 % \caption@setoptions{####1}}}%
3082 % \let\@capytype\LTcapytype
3083 % \caption@ORI@ContinuedFloat}}%
3084 % }{%
3085 % \def\ContinuedFloat{\noalign{%
3086 % \caption@Error{%
3087 % \noexpand\ContinuedFloat inside longtables\MessageBreak
3088 % is only available with 'hycap=true'}}}%
3089 % }%
3090 % \global\let\caption@setContinuedFloat\@empty
3091 % \def\ContinuedFloat{\noalign{%
3092 % \caption@Error{\noexpand\ContinuedFloat outside float}}}%
3093 % \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 `\LTcapytype` instead of `{table}`:

```

3094 \long\def\LT@c@ption#1[#2]#3{%
3095   \LT@makecaption#1{\csname fnun@\LTcapytype\endcsname}{#3}%
3096   \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>\hsize

```

```

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

```

Our definition:

```

3097 \renewcommand\LT@makecaption[3]{%
3098 \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`.)

```

3099 \caption@settype*\LTcaptype
3100 \ifdim\LTcapwidth=4in \else
3101 \setcaptionwidth\LTcapwidth
3102 \fi
3103 \caption@setoptions{longtable}%
3104 % \caption@setContinuedFloat
3105 \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!

```

3106 \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%

```

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

```

3107 \caption@startrue#1\caption@starfalse
3108 \caption@resetContinuedFloat\LTcaptype
3109 \caption@begin\LTcaptype
3110 \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...)

```

3111 \vskip-\ht\strutbox

```

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

```

3112 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
3113 \caption@@make{#2}{#3}\endgraf
3114 \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
3115 \caption@end}}%

```

```

3116 {} {}

```

2.15.8 The picinpar package

```
3117 \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.

```
3118 \long\def\figwindow[#1,#2,#3,#4] {%
3119   \caption@window{figure}%
3120   \caption@setoptions{figwindow}%
3121   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
3122 \long\def\tabwindow[#1,#2,#3,#4] {%
3123   \caption@window{table}%
3124   \caption@setoptions{tabwindow}%
3125   \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.

```
3126 \newcommand*\caption@window[1]{%
3127   \let\caption@boxrestore\@parboxrestore
3128   \let\@makecaption\caption@@make
3129   \caption@setautoposition b%
3130   \caption@settype{#1}%
3131   \caption@clearmargin}%

```

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

```
3132 \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`.

```
3133   \ifdim\picwd=\z@
3134     \let\caption@makecurrent\@gobbletwo
3135     \let\caption@@start\relax
3136     \caption@prepareslc
3137   \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.

```
3138   \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
3139   \if\@tempa*%
3140     \let\@tempa\@firstofone
3141   \else\if\@tempa[%
3142     \let\@tempa\@firstofone
3143   \else
3144     \let\@tempa\@empty
3145   \fi\fi
3146   \expandafter\caption\@tempa{#1}}%
```

```
3147 }{ }
```


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.

```
3148 \newcommand*\piccaptiontype[1]{\def\@piccapttype{#1}}
```

```
3149 \caption@ifpackageloaded{picins}{%
```

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

```
3150 \@ifundefined{piccapttype}{%
```

```
3151 \caption@iftype{%
```

```
3152 \let\@piccapttype\@capttype
```

```
3153 }{%
```

```
3154 \def\@piccapttype{figure}%
```

```
3155 }%
```

```
3156 }{}%
```

```
3157 \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.

```
3158 \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.

```
3159 \let\caption@ORI\make@piccaption\make@piccaption
```

```
3160 \def\make@piccaption{%
```

```
3161 \let\caption@ORI\caption
```

```

3162 \long\def\caption[##1]##2{%
3163 \caption@freezeHref % will be defrosted in \ivparpic
3164 \caption@settype\@piccaptiontype
3165 % \ifnum\c@piccaptionpos>2\relax
3166 \caption@clearmargin
3167 % \else
3168 % \captionwidth\z@ % do not use "width=" setting
3169 % \fi
3170 \caption@setoptions{parpic}%
3171 \let\caption@boxrestore\@parboxrestore
3172 \caption@setautoposition b%
3173 \expandafter\expandafter\expandafter\caption@ORI
3174 \expandafter\expandafter\expandafter[%
3175 \expandafter\expandafter\expandafter{%
3176 \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
3177 \caption@ORI@make@piccaption
3178 \let\caption\caption@ORI}%

```

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

```

3179 \let\caption@ORI@ivparpic\ivparpic
3180 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
3181 \let\caption@ORI@noindent\noindent
3182 \def\noindent{%
3183 \caption@defrostHref
3184 \let\noindent\caption@ORI@noindent
3185 \noindent}%
3186 \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
3187 \let\noindent\caption@ORI@noindent}%
3188 }{%
3189 \let\piccaptiontype\@undefined
3190 }

```

2.15.10 The rotating package

```

3191 \caption@ifPackageLoaded{rotating}[1995/08/22 v2.10]{%

```

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

```

3192 \def\rotcaption{\let\makecaption\@makerotcaption\caption}%
3193 % \let\@rotcaption\@undefined

```

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

```

3194 \def\rotcaptionof{%
3195 \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

```

`\@makerotcaption` Original (bugfixed) code:

```
\long\def\@makerotcaption#1#2{%
  \setbox\@tempboxa\hbox{#1: #2}%
  \ifdim \wd\@tempboxa > .8\vsiz
    \rotatebox{90}{%
      \begin{minipage}{.8\textheight}#1: #2\end{minipage}%
    }%\par    % <== \par removed (AR)
  \else%
    \rotatebox{90}{\box\@tempboxa}%
  \fi
  \nobreak\hspace{12pt}% <== \nobreak added (AR)
}
```

Our version emulates this behavior, but if `width=` is set, the rotated caption is always typeset as minipage. (Note that `margin=` is not supported here.)

```
3196 \long\def\@makerotcaption#1#2{%
3197   \ifdim\captionwidth=\z@
3198     \setcaptionwidth{.8\textheight}%
3199     \caption@slc{#1}{#2}{.8\vsiz}{%
3200       \let\caption@makerot\caption@@make
3201       \caption@clearmargin
3202 %       \long\def\caption@parbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
3203 %       (not needed because \rotatebox uses an \hbox anyway)
3204       \let\caption@parbox\@secondoftwo}%
3205       \caption@set@bool\caption@ifslc0% been there, done that
3206     \fi
3207     \rotatebox{90}{\caption@makerot{#1}{#2}}%
3208     \nobreak\hspace{12pt}}%
3209 \newcommand\caption@makerot[2]{%
3210   \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
3211 \caption@For{typelist}{%
3212   \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
3213   \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
3214 }
```

2.15.11 The sidecap package

```
3215 \caption@ifpackageloaded{sidecap}[1999/05/11 v1.4d]{%
3216   \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.)

```
3217   \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`.

```
3218   \let\caption@ORI@SC@zfloat\SC@zfloat
3219   \def\SC@zfloat#1#2#3[#4]{%
```

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

```
3220 \let\caption@ORI\caption
3221 \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
3222 \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.

```
3223 \caption@settype*{#2}%
3224 \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.

```
3225 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
3226 \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.

```
3227 \let\caption@ORI@settype\caption@settype
3228 \def\caption@settype##1{% will be done in \xfloat
3229 \caption@ORI@settype*{##1}% do not change \@currentlabel
3230 \caption@setSC@justify
3231 %%% \caption@setoptions{SCfloat}%
3232 \caption@setoptions{SC\@capttype}%
3233 \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

```
3234 \def\caption@setSC@justify{%
3235 \caption@clearmargin
3236 \@ifundefined{SC@justify}{}{%
3237 \ifx\SC@justify\@empty \else
3238 \let\caption@hj\SC@justify
3239 \let\SC@justify\@empty
3240 \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!

```
3241 \let\caption\SC@orig@caption
3242 \def\SC@orig@caption[##1]##2{\caption@defrost}%
```

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

```
3243 \caption@setSC@justify % for compatibility mode
3244 \caption@ORI@endSC@FLOAT}%
```

```

3245 \newcommand*\caption@For@SC[2]{%
3246   \def#1{b}% = \sidecaptionvpos{#2}{b} (v1.6)
3247   \newenvironment{SC#2}%
3248     {\SC@float[#1]{#2}}{\endSC@float}%
3249   \newenvironment{SC#2*}%
3250     {\SC@dblfloat[#1]{#2}}{\endSC@dblfloat}}%
3251 \@onlypreamble\caption@For@SC
3252 \caption@For{typelist}{%
3253   \expandafter\caption@For@SC\csname SC@#1@vpos\endcsname{#1}}%
3254 }{}

```

2.15.12 The subfigure package

```

3255 \caption@IfPackageLoaded{subfigure}[2002/01/23 v2.1]{%
\sif@ifpositiontop If the subfigure package is loaded, we map \sf@ifpositiontop to \iffiguresettopcap
resp. \iftabletopcap, so the subfigure v2.1 options figbotcap etc. will still work.
3256 \def\sif@ifpositiontop{%
3257   \ifx\@capttype\@undefined
3258     \expandafter\@gobbletwo
3259   \else\ifx\@capttype\relax
3260     \expandafter\expandafter\expandafter\@gobbletwo
3261   \else
3262     \expandafter\expandafter\expandafter\sif@if@position@top
3263   \fi\fi}
3264 \def\sif@if@position@top{%
3265   \@ifundefined{if\@capttype topcap}%
3266     {\@gobbletwo}%
3267     {\@nameuse{if\@capttype topcap}}%
3268     \expandafter\@firstoftwo
3269   \else
3270     \expandafter\@secondoftwo
3271   \fi}}
3272 }{}

```

2.15.13 The supertabular and xtab packages

```

3273 \caption@IfPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.
3274 \renewcommand*\tablecaption{%
3275   \caption@star
3276   {\refstepcounter{table}}}%
3277   {\caption@dblarg{\@xtablecaption}}}%
\@xtablecaption Make \nameref and \autoref work.
3278 \let\caption@ORI@xtablecaption\@xtablecaption
3279 \long\def\@xtablecaption[#1]#2{%
3280   \caption@getttitle{#2}%
3281   \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}

```

```

3282 \long\def\ST@caption#1[#2]#3{\par%
3283   \caption@settype*{#1}%
3284   \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.

```

3285   \def\caption@fixposition{%
3286     \caption@setposition{\if@topcaption t\else b\fi}}%
3287   \caption@beginex{#1}{#2}{#3}%
3288   \caption@boxrestore
3289   \caption@normalsize
3290   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3291   \caption@end}%

```

```

3292 }{}

```

```

3293 \caption@ifpackageloaded{xtab}[2000/04/09 v2.3]{%

```

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

```

3294   \renewcommand*\tablecaption{%
3295     \caption@star
3296     {\refstepcounter{table}}}%
3297     {\caption@dblarg{\@xtablecaption}}}%

```

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

```

3298   \let\caption@ORI@xtablecaption\@xtablecaption
3299   \long\def\@xtablecaption[#1]#2{%
3300     \caption@getttitle{#2}%
3301     \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}}

3302 \long\def\ST@caption#1[#2]#3{\par%
3303 \caption@settype*{#1}%
3304 \caption@setoptions{xtabular}%
3305 \def\caption@fixposition{%
3306 \caption@setposition{\if@topcaption t\else b\fi}}%
3307 \@initisotab
3308 \caption@beginex{#1}{#2}{#3}%
3309 \caption@boxrestore
3310 \caption@normalsize
3311 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3312 \caption@end
3313 \global\advance\ST@pageleft -\PWSTcapht
3314 \ST@trace\tw@{Added caption. Space left for xtabular: \the\ST@pageleft}}%

3315 }{}

```

2.15.14 The threeparttable package

```

3316 \caption@ifPackageLoaded{threeparttable}[2003/06/13 v3.0]{%
\threeparttable Unfortunately \@capytype is not set when \TPT@common will be used, so we have to
redefine \threeparttable and \measuredfigure instead.
3317 \let\caption@ORI@threeparttable\threeparttable
3318 \renewcommand*\threeparttable{%
3319 \caption@settype{table}%
3320 \caption@setposition a% ?
3321 \caption@clearmargin
3322 \caption@setoptions{threeparttable}%
3323 \caption@ORI@threeparttable}%

\measuredfigure Same here...
3324 \let\caption@ORI@measuredfigure\measuredfigure
3325 \renewcommand*\measuredfigure{%
3326 \caption@settype{figure}%
3327 \caption@setposition a% ?
3328 \caption@clearmargin
3329 \caption@setoptions{measuredfigure}%
3330 \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}

```

```

3331 \def\TPT@caption#1[#2]#3{%
3332   \gdef\TPT@docapt{%
3333     \global\let\TPT@docapt\@undefined
3334     \caption@setautoposition\caption@TPT@position
3335     \TPT@LA@caption{#1}[#2][#3]}%
3336   \ifx\TPT@hsize\@empty
3337     \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
3338     \gdef\caption@TPT@position{t}%
3339     \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
3340   \else
3341     \def\caption@TPT@position{b}%
3342     \TPT@docapt
3343   \fi
3344   \ignorespaces}%

3345 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
3346 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%

3347 {}

```

2.15.15 The wrapfig package

```

3348 \caption@ifPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)
\float@ifstyle \float@ifstyle{<type>}{<if-clause>}{<else-clause>}
(see float package support for details)
3349 \providecommand*\float@ifstyle[1]{%
3350   \expandafter\ifx\csname fst@#1\endcsname\relax
3351     \expandafter\@secondoftwo
3352   \else
3353     \expandafter\@firstoftwo
3354   \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.

```

3355 \newcommand*\caption@restylewrapfloat[1]{%
3356   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
3357     \csname wrap#1\endcsname
3358   \caption@Error{%
3359     For a successful cooperation of the 'wrapfig' package\MessageBreak
3360     with the 'float' package you should load the 'wrapfig'\MessageBreak
3361     package *after* the 'float' package}%
3362   \else
3363     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
3364       \csname wrap#1\endcsname
3365     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
3366     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
3367       \csname wrap#1\endcsname
3368   \fi}%

```


`\caption@wrapfloat`

```
3369 \newcommand*\caption@wrapfloat[1]{%
3370   \caption@settype*{#1}%
3371   \float@ifstyle{#1}{%
3372     \ifx\WF@floatstyhook\@undefined
3373       \caption@Error{%
3374         For a successful cooperation of the 'wrapfig' package\MessageBreak
3375         with the 'float' package you should use at least\MessageBreak
3376         'wrapfig' version 3.6}%
3377     \else
3378       \float@dostyle{#1}%
3379     \fi}{}%
3380   \caption@clearmargin
3381   %%% \caption@setoptions{wrapfloat}%
3382   \caption@setoptions{wrap#1}%
3383   \@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`.)

```
3384 \caption@restylewrapfloat{figure}%
3385 \caption@restylewrapfloat{table}%
3386 \caption@For{typelist}{%
3387   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
3388   \caption@restylewrapfloat{#1}}%
3389 \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.

```
3390 \ifundefined{float@restyle}{}{%
3391   \toks@=\expandafter{\float@restyle{#1}% (env may or may not be defined)
3392   \caption@restylewrapfloat{#1}}%
3393   \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
3394   \@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`.

```
3395 \let\caption@ORI@wrapfloat\wrapfloat
3396 \def\wrapfloat#1{%
3397   \float@ifstyle{#1}{%
3398     \caption@Error{%
3399       For a successful cooperation of the 'wrapfig' package\MessageBreak
3400       with the 'float' package you should load the 'wrapfig'\MessageBreak
3401       package *right after* the 'float' package}}}%
3402   \caption@ORI@wrapfloat{#1}%
3403   \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:

```
3404 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
3405   \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
3406   \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
3407   \caption@start
3408   \ifdim\hsize>\z@ \@parboxrestore \else
3409   \setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
3410   \ignorespaces \fi}%
3411 }{ }
```

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