

The Implementation of the caption package^{*}

Axel Sommerfeldt
`caption@sommerfee.de`

2008/08/24

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 v3.1j, last revised 2008/08/24.

¹Thanks a lot to Olga Lapko for this translation

Contents

1	Kernel	4
1.1	Identification	4
1.2	Generic helpers	4
1.3	Errors	7
1.4	Using the keyval package	8
1.5	Margin resp. width	12
1.6	Indentions	14
1.7	Styles	15
1.8	Formats	16
1.9	Label formats	17
1.10	Label separators	18
1.11	Text formats	19
1.12	Fonts	19
1.13	Justifications	21
1.13.1	The ragged2e package	22
1.14	Vertical spaces before and after captions	23
1.15	Positioning	23
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	26
1.19.3	The beamer class	27
1.19.4	The KOMA-Script classes	28
1.19.5	The NTG Dutch classes	29
1.19.6	The thesis class	29
1.19.7	The frenchb Babel option	30
1.19.8	The frenchle/pro package	30
1.20	Execution of options	31
1.21	Making an ‘List of’ entry	31
1.22	Typesetting the caption	31
1.23	Types & sub-types	35
1.24	subfig package adaption	45
2	Main package	47
2.1	Identification	47
2.2	Loading the kernel	47
2.3	Check against incompatible packages	47

2.4	Check document class	47
2.5	Adaption to the \mathcal{AMS} & SMF document classes	47
2.6	Emulation of the KOMA-Script commands	48
2.7	Declaration of options	49
2.7.1	Options for figure and table	49
2.7.2	Miscellaneous options	50
2.7.3	caption v1.x compatibility options	51
2.7.4	caption2 v2.x compatibility options	51
2.7.5	Obsolete caption v3.0 options	52
2.7.6	ftpage package support options	52
2.7.7	hyperref package support options	52
2.8	Processing of options	52
2.9	<code>\captionof</code> and <code>\captionlistentry</code>	52
2.10	<code>\ContinuedFloat</code>	55
2.11	Internal helpers	56
2.12	<code>\caption</code> , <code>\@caption</code> , and <code>\@makecaption</code>	58
2.13	Support for sub-captions	65
2.14	Document class & Babel package support	67
2.14.1	The \mathcal{AMS} & SMF classes	67
2.14.2	The beamer class	67
2.14.3	The KOMA-Script classes	67
2.14.4	The frenchb Babel option	68
2.14.5	The frenchle/pro package	68
2.15	Package support	69
2.15.1	The float package	71
2.15.2	The floatflt package	74
2.15.3	The ftpage package	74
2.15.4	The hyperref package	77
2.15.5	The hypcap package	80
2.15.6	The listings package	80
2.15.7	The longtable package	81
2.15.8	The picinpar package	84
2.15.9	The picins package	85
2.15.10	The rotating package	87
2.15.11	The sidecap package	88
2.15.12	The subfigure package	89
2.15.13	The supertabular and xtab packages	90
2.15.14	The threeparttable package	91
2.15.15	The wrapfig package	92

1 Kernel

1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2008/08/24 v3.1j 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{\<cmd>}{\<list entry>}{\<yes>}{\<no>}
32 \newcommand*\caption@ifin@list[2]{%
33 \caption@ifempty@list#1%
34 {\@secondoftwo}%
35 {\@expandtwoargs\caption@@ifinlist{#2}{#1}}}

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

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

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

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

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

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

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

```

```

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

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

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

\caption@teststar \caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
\caption@teststar@{<cmd>}{<star arg>}{<non-star arg>}
74 \newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}
75 \newcommand*\caption@teststar@[3]{%
76   \@ifstar{#1{#2}}{\caption@ifatletter{#1{#2}}{#1{#3}}}}
77 \AtBeginDocument{\let\caption@teststar@\caption@teststar}

78 \newcommand*\caption@ifatletter{%
79   \ifnum\the\catcode`\@=11
80     \expandafter\@firstoftwo
81   \else
82     \expandafter\@secondoftwo
83   \fi}
84 \AtBeginDocument{\let\caption@ifatletter\@secondoftwo}

\caption@withoptargs \caption@withoptargs{<cmd>}
85 \newcommand*\caption@withoptargs[1]{%
86   \@ifstar
87   {\def\caption@tempa{*}\caption@@withoptargs#1}%
88   {\def\caption@tempa{}\caption@@withoptargs#1}}
89 \def\caption@@withoptargs#1{%
90   \@ifnextchar[%]
91   {\caption@@@withoptargs#1}%
92   {\caption@@@withoptargs#1}}
93 \def\caption@@@withoptargs#1[#2]{%
94   \l@addto@macro\caption@tempa{[#2]}%
95   \caption@@withoptargs#1}
96 \def\caption@@@withoptargs#1{%
97   \expandafter#1\expandafter{\caption@tempa}}

\caption@CheckCommand \caption@CheckCommand{<macro>}{<definition of macro>}
\caption@IfCheckCommand checks if a command already exists, with the same definition. It can be used more-than-
once to check if one of multiple definitions will finally match. (It redefines itself later on
to \@gobbletwo if the two commands match fine, making further checks harmless.)
\caption@IfCheckCommand{<true>}{<false>}
will execute the <true> code if one match was finally given, the <false> code otherwise.
(It simply checks if \caption@CheckCommand is \@gobbletwo and restores the
starting definition of \caption@CheckCommand.)

```

```

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

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

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

```

`\caption@AtBeginDocument`

`\caption@AtBeginDocument*{<code>}`

Same as `\AtBeginDocument` but the execution of code will be surrounded by two `\PackageInfos`. The starred variant causes the code to be executed after all code specified using the non-starred variant.

```

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

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

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

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

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

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

```

1.3 Errors

<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   \let\KV@errx\KV@err
277   \edef\caption@keydepth{\caption@keydepth i}%

278   \caption@Debug{\protect\setkeys{#2}{#3}}%
279   \setkeys{#2}{#3}%

```

```

280 \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
281 \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
282 \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
283 \ifx\caption@keydepth\@empty \captionsetup@starfalse \fi
284 \@esphack}
285 \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.
286 \newcommand*\caption@ExecuteOptions[2]{%
287 \@expandtwoargs\caption@setkeys{#1}{#2}}%
288 \@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.)
289 \newcommand*\caption@ProcessOptions{%
290 \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
291 \@onlypreamble\caption@ProcessOptions

292 \newcommand*\caption@@ProcessOptions[2]{%
293 \let\@tempc\relax
294 \let\caption@tempa\@empty
295 #1{% \@firstofone -or- \@gobble
296 \@for\CurrentOption:=\@classoptionslist\do{%
297 \@ifundefined{KV@#2@\CurrentOption}{}{%
298 \@ifundefined{KV@#2@\CurrentOption @default}{%
299 \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
300 }{%
301 \PackageInfo{#2}{Global option '\CurrentOption' processed}%
302 \edef\caption@tempa{\caption@tempa,\CurrentOption,}%
303 \@expandtwoargs\@removeelement\CurrentOption
304 \@unusedoptionlist\@unusedoptionlist
305 }%
306 }%
307 }%
308 \let\CurrentOption\@empty
309 }%
310 \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
311 \AtEndOfPackage{\let\@unprocessedoptions\relax}}
312 \@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 v1.x.

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

```

313 \newdimen\captionmargin
314 \newdimen\captionmargin@
315 \newdimen\captionwidth

316 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
317 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
318 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
319 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}
320 \DeclareCaptionOptionNoValue{oneside}{\caption@set@bool\caption@iftwoside0}

321 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
322 \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.

323 \newcommand*\setcaptionmargin{%
324   \caption@teststar\caption@setmargin\@gobble\@firstofone}

325 \newcommand*\caption@setmargin[2]{%
326   #1{\captionwidth\z@}%
327   \caption@@setmargin#2,#2,\@nil}

328 \def\caption@@setmargin#1,#2,#3\@nil{%
329   \setlength\captionmargin@{#2}%
330   \setlength\captionmargin{#1}%
331   \addtolength\captionmargin@{-\captionmargin}}

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

332 \newcommand*\setcaptionwidth{%
333   \captionmargin\z@
334   \captionmargin@\z@
335   \setlength\captionwidth}

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

336 \newcommand*\caption@thecounter{0}

337 \newcommand*\caption@stepcounter{%
338   \@tempcnta\caption@thecounter
339   \advance\@tempcnta@ne
340   \xdef\caption@thecounter{\the\@tempcnta}}

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

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

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

342 \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`.

```

343 \newcommand*\caption@label[1]{%
344   \caption@@label
345   \protected@write\@auxout{\let\caption@thepage\relax}%
346     {\string\caption@newlabel{#1}{\caption@thepage}}}
347 \newcommand*\caption@@label{%
348   \global\let\caption@@label\relax
349   \protected@write\@auxout{}%
350     {\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`.

```

351 \newcommand*\caption@pageref[1]{%
352   \expandafter\ifx\csname caption@r@#1\endcsname\relax
353     \G@refundefinedtrue % => 'There are undefined references.'
354     \caption@Warning{Reference on page \thepage \space undefined}%
355   \else
356     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
357   \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.

```

358 \newcommand*\caption@ifoddpage{%
359   \caption@iftwoside{%
360     \caption@label\caption@thecounter
361     \caption@pageref\caption@thecounter
362     \ifodd\caption@thepage
363       \let\caption@ifoddpage\@firstoftwo
364     \else
365       \let\caption@ifoddpage\@secondoftwo
366     \fi
367   }{\let\caption@ifoddpage\@firstoftwo}%
368   \caption@ifoddpage}
```

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

```

369 \newcommand*\caption@setoptcmd[2]{%
370   \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` 371 \newdimen\caption@indent

`\caption@hangindent` 372 \newdimen\caption@parindent

```

373 \newdimen\caption@hangindent

374 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
375   \setlength\caption@indent{#1}}
376 \DeclareCaptionOption{indentation}[\leftmargini]{%
377   \setlength\caption@indent{#1}}
378 \DeclareCaptionOption{parindent}{%
```

```

379      \setlength\caption@parindent{#1}}
380 \DeclareCaptionOption{hangindent}{%
381      \setlength\caption@hangindent{#1}}
382 \DeclareCaptionOption{parskip}{%
383      \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

```

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

```

1.7 Styles

```

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

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!)

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.

```

411 \newcommand*\caption@setstyle{%
412   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

413 \newcommand*\caption@@setstyle[2]{%
414   \@ifundefined{caption@sty@#2}%
415     {#1{\caption@Error{Undefined style `#2'}}}%
416     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
417     \ifx\caption@setstyle@flag\@undefined
418       \let\caption@setstyle@flag\relax
419       \caption@resetstyle
420       \caption@xsetup\caption@sty
421       \let\caption@setstyle@flag\@undefined
422     \else
423       \caption@xsetup\caption@sty
424     \fi
425     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
426     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
427     \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!*

```

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

```

436 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
437 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{}
438   format=default,labelsep=default,textformat=default,%
439   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.

```

440 \newcommand*\DeclareCaptionFormat{%
441   \caption@teststar\caption@declareformat\@gobble\@firstofone}
442 \@onlypreamble\DeclareCaptionFormat
```



```

443 \newcommand*\caption@declareformat[2]{%
444   \@dblarg{\caption@@declareformat#1{#2}}
445 \@onlypreamble\caption@declareformat
446 \long\def\caption@@declareformat#1#2[#3]#4{%
447   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
448   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
449   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
450 \@onlypreamble\caption@@declareformat
451 \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`).

```

452 \newcommand*\caption@setformat[1]{%
453   \@ifundefined{caption@fmt@#1}%
454   {\caption@Error{Undefined format `#1'}}%
455   {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
456    \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
457    \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

`\DeclareCaptionDefaultFormat`

```

458 \newcommand*\DeclareCaptionDefaultFormat[1]{%
459   \expandafter\def\expandafter\caption@fmt@default\expandafter
460     {\csname caption@fmt@#1\endcsname}%
461   \expandafter\def\expandafter\caption@slfmt@default\expandafter
462     {\csname caption@slfmt@#1\endcsname}%
463   \expandafter\def\expandafter\caption@ifh@default\expandafter
464     {\csname caption@ifh@#1\endcsname}}
465 \@onlypreamble\DeclareCaptionDefaultFormat

```

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

```

466 \DeclareCaptionFormat{plain}{#1#2#3\par}
467 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
468   \caption@ifin@list\caption@lsep@crlist\caption@lsepname
469   {\caption@Error{%
470     The option ‘labelsep=\caption@lsepname’ does not work\MessageBreak
471     with ‘format=hang’}}%
472   {\@hangfrom{#1#2}%
473    \advance\caption@parindent\hangindent
474    \advance\caption@hangindent\hangindent
475    \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

476 \DeclareCaptionDefaultFormat{plain}

```

1.9 Label formats

```

\DeclareCaptionLabelFormat    \DeclareCaptionLabelFormat{<name>}{<code with #1 and #2>}
477 \newcommand*\DeclareCaptionLabelFormat[2]{%
478   \global\@namedef{caption@lfmt@#1}##1##2{#2}}
479 \@onlypreamble\DeclareCaptionLabelFormat

```

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

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

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

```
481 \newcommand*\caption@setlabelformat[1]{%
```

```
482 \ifundefined{caption@lfmt@#1}%
```

```
483 {\caption@Error{Undefined label format `#1'}}%
```

```
484 {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}
```

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

```
485 \DeclareCaptionLabelFormat{empty}{}%
```

```
486 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
```

```
487 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace} (#2)}
```

```
488 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace} #2}
```

‘default’ usually maps to ‘simple’.

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

1.10 Label separators

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

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

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

```
490 \newcommand\DeclareCaptionLabelSeparator{%
```

```
491 \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
```

```
492 \@onlypreamble\DeclareCaptionLabelSeparator
```

```
493 \newcommand\caption@declarelabelseparator[3]{%
```

```
494 \global\expandafter\let\csname caption@ifl@#2\endcsname#1%
```

```
495 \global\long\@namedef{caption@lsep@#2}{#3}%
```

```
496 \caption@@declarelabelseparator{#2}#3\\@nil}
```

```
497 \@onlypreamble\caption@declarelabelseparator
```

```
498 \long\def\caption@@declarelabelseparator#1#2\\#3\\@nil{%
```

```
499 \def\@tempa{#3}\ifx\@tempa\empty \else
```

```
500 \caption@g@addto@list\caption@lsep@#1%
```

```
501 \fi}
```

```
502 \@onlypreamble\caption@@declarelabelseparator
```

```
503 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
```

```
504 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

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

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

```
505 \newcommand*\caption@setlabelseparator[1]{%
```

```
506 \ifundefined{caption@lsep@#1}%
```

```
507 {\caption@Error{Undefined label separator `#1'}}%
```

```
508 {\edef\caption@lsepname{#1}%
```

```
509 \expandafter\let\expandafter\caption@ifl\csname caption@ifl@#1\endcsname
```

```
510 \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’.

```
511 \DeclareCaptionLabelSeparator{none}{}%
```

```

512 \DeclareCaptionLabelSeparator{colon}{: }
513 \DeclareCaptionLabelSeparator{period}{. }
514 \DeclareCaptionLabelSeparator{space}{ }
515 \DeclareCaptionLabelSeparator*{quad}{\quad}
516 \DeclareCaptionLabelSeparator*{newline}{\\}
517 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}
‘default’ usually maps to ‘colon’.
518 \def\caption@lsep@default{\caption@lsep@colon}
519 \def\caption@iflf@default{\caption@iflf@colon}

```

1.11 Text formats

```

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

523 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
524 \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`).

```

525 \newcommand*\caption@settextformat[1]{%
526   \ifundefined{caption@tfmt@#1}%
527     {\caption@Error{Undefined text format `#1'}}%
528     {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

529 \DeclareCaptionTextFormat{simple}{#1}
530 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```
531 \def\caption@tfmt@default{\caption@tfmt@simple}
```

1.12 Fonts

```

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

DeclareCaptionDefaultFont \DeclareCaptionDefaultFont{<name>}{<code>}
535 \newcommand*\DeclareCaptionDefaultFont[2]{%
536   \global\@namedef{caption#1@default}{#2}}
537 \@onlypreamble\DeclareCaptionDefaultFont

538 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
539 \DeclareCaptionOption{font+}{\caption@addtofont{font}{#1}}
540 \DeclareCaptionDefaultFont{font}{}

541 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
542 \DeclareCaptionOption{labelfont+}{\caption@addtofont{labelfont}{#1}}
543 \DeclareCaptionDefaultFont{labelfont}{}

```

```

544 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
545 \DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
546 \DeclareCaptionDefaultFont{textfont}{}

```

`\caption@setfont` `\caption@setfont{<name>}{<keyval-list of names>}`

Selecting a caption font means saving all the code snippets in `\caption<name>`.

```

547 \newcommand*\caption@setfont[1]{%
548   \expandafter\let\csname caption#1\endcsname\@empty
549   \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.

```

550 \newcommand*\caption@addtofont[2]{%
551   \begingroup
552     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
553     \define@key{caption@fnt}{default}[]{%
554       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
555     \caption@setkeys[caption]{caption@fnt}{#2}%
556     \global\let\caption@tempa\caption@fnt
557   \endgroup
558   \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`.

```

559 \newcommand*\caption@font{%
560   \caption@teststar\caption@@font\@firstofone
561   {\caption@setkeys[caption]{caption@fnt}}%
562 \newcommand*\caption@@font[2]{%
563   \begingroup
564   \def\caption@fnt{\endgroup}%
565   #1{#2}%
566   \caption@fnt}

```

These are the pre-defined font code snippets.

```

567 \DeclareCaptionFont{normalcolor}{\normalcolor}
568 \DeclareCaptionFont{color}{\color{#1}}

569 \DeclareCaptionFont{normalfont}{\normalfont}
570 \DeclareCaptionFont{up}{\upshape}
571 \DeclareCaptionFont{it}{\itshape}
572 \DeclareCaptionFont{sl}{\slshape}
573 \DeclareCaptionFont{sc}{\scshape}
574 \DeclareCaptionFont{md}{\mdseries}
575 \DeclareCaptionFont{bf}{\bfseries}
576 \DeclareCaptionFont{rm}{\rmfamily}
577 \DeclareCaptionFont{sf}{\sffamily}
578 \DeclareCaptionFont{tt}{\ttfamily}

579 \DeclareCaptionFont{scriptsize}{\scriptsize}
580 \DeclareCaptionFont{footnotesize}{\footnotesize}
581 \DeclareCaptionFont{small}{\small}
582 \DeclareCaptionFont{normalsize}{\normalsize}

```

```

583 \DeclareCaptionFont{large}{\large}
584 \DeclareCaptionFont{Large}{\Large}

585 \DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespace}% normally 1
586 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
587 \DeclareCaptionFont{doublespacing}{\doublespacing}
588 \DeclareCaptionFont{stretch}{\setstretch{#1}}
589 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }

590 %\DeclareCaptionFont{normal}{%
591 % \caption@font{normalcolor,normalfont,normalsize,singlespacing}
592 \DeclareCaptionFont{normal}{%
593 \caption@font*{%
594 \KV@caption@fnt@normalcolor\@unused
595 \KV@caption@fnt@normalfont\@unused
596 \KV@caption@fnt@normalsize\@unused
597 \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.

```

598 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
599 \DeclareCaptionDefaultFont{size}{}

```

1.13 Justifications

```

\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
600 \newcommand*\DeclareCaptionJustification[2]{%
601 \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
602 \DeclareCaptionFont{#1}{#2}}
603 \@onlypreamble\DeclareCaptionJustification

\captionDefaultJustification \DeclareCaptionDefaultJustification{<code>}
604 \newcommand*\DeclareCaptionDefaultJustification[1]{%
605 \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
606 \DeclareCaptionDefaultFont{@hj}{#1}}
607 \@onlypreamble\DeclareCaptionDefaultJustification

608 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
609 \DeclareCaptionDefaultJustification{}

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

```

These are the pre-defined justification code snippets.

```

611 \DeclareCaptionJustification{justified}{}
612 \DeclareCaptionJustification{centering}{\centering}
613 \DeclareCaptionJustification{centerfirst}{\centerfirst}
614 \DeclareCaptionJustification{centerlast}{\centerlast}
615 \DeclareCaptionJustification{raggedleft}{\raggedleft}
616 \DeclareCaptionJustification{raggedright}{\raggedright}

```

```

\centerfirst Please blame Frank Mittelbach for the code of \centerfirst :-)
617 \providecommand\centerfirst{%

```

```

618 \let\\\@centercr
619 \edef\caption@normaladjust{%
620   \leftskip\the\leftskip
621   \rightskip\the\rightskip
622   \parfillskip\the\parfillskip\relax}%
623 \leftskip\z@\@plus -1fil%
624 \rightskip\z@\@plus 1fil%
625 \parfillskip\z@skip
626 \noindent\hskip\z@\@plus 2fil%
627 \setpar{\@par\@restorepar\caption@normaladjust}}

```

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

```

628 \providecommand\centerlast{%
629   \let\\\@centercr
630   \leftskip\z@\@plus 1fil%
631   \rightskip\z@\@plus -1fil%
632   \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.

```

633 \DeclareCaptionJustification{Centering}{%
634   \caption@ragged\Centering\centering}
635 \DeclareCaptionJustification{RaggedLeft}{%
636   \caption@ragged\RaggedLeft\raggedleft}
637 \DeclareCaptionJustification{RaggedRight}{%
638   \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.

```

639 \newcommand*\caption@ragged{%
640   \caption@Debug{We need ragged2e}%
641   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
642   \global\let\caption@ragged\caption@@ragged
643   \caption@ragged}

644 \caption@AtBeginDocument{%
645   \@ifundefined{caption@r@ragged2e}{%
646     \newcommand*\caption@@ragged{%
647       \caption@Warning{%
648         'ragged2e' support has been changed.\MessageBreak
649         Rerun to get captions right}%
650     \global\let\caption@ragged\@secondoftwo % suppress further warnings
651     \caption@ragged}%
652   }{%
653     \caption@Debug{We load ragged2e}%

```

```

654 \IfFileExists{ragged2e.sty}{%
655 \RequirePackage{ragged2e}%
656 \let\caption@@ragged\@firstoftwo
657 }{%
658 \newcommand*\caption@@ragged[2]{%
659 \ifundefined{caption\string#1}{%
660 \caption@Warning{%
661 'ragged2e' package not loaded, therefore\MessageBreak
662 substituting \string#2 for \string#1\MessageBreak}%
663 \global\@namedef{caption\string#1}}}%
664 #2}%
665 }%
666 }}

```

1.14 Vertical spaces before and after captions

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

`\belowcaptionskip`

```

667 \ifundefined{abovecaptionskip}{%
668 \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
669 \ifundefined{belowcaptionskip}{%
670 \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

671 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
672 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
673 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

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

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

```

674 \newcommand*\caption@rule{\caption@ifrule\caption@hrule\relax}
675 \newcommand*\caption@hrule{\hrule\@height\z@}

676 \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.

```

677 \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.*

```

678 \newcommand*\caption@setposition[1]{%
679 \caption@ifinlist{#1}{d,default}{%
680 \let\caption@position\caption@defaultpos
681 }{\caption@ifinlist{#1}{t,top,above}{%

```

```

682 \let\caption@position\@firstoftwo
683 }\caption@ifinlist{#1}{b,bottom,below}{%
684 \let\caption@position\@secondoftwo
685 }\caption@ifinlist{#1}{a,auto}{%
686 \let\caption@position\@undefined
687 }{%
688 \caption@Error{Undefined position '#1'}%
689 }}}}

```

`\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.

```

690 %\caption@setdefaultpos{a}% default = auto
691 \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.)

```

692 \newcommand*\caption@iftop{%
693 \ifx\caption@position\@undefined
694 \let\caption@position\@secondoftwo
695 % = \caption@setposition b%
696 \fi
697 \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.

```

698 \newcommand*\caption@fixposition{%
699 \ifx\caption@position\@undefined
700 \caption@autoposition
701 \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.

```

702 \newcommand*\caption@autoposition{%
703 \ifvmode
704 \edef\caption@tempa{\the\prevdepth}%
705 \caption@Debug{\protect\prevdepth=\caption@tempa}%
706 \ifdim\prevdepth>-\p@
707 \let\caption@position\@secondoftwo
708 \else
709 \let\caption@position\@firstoftwo
710 \fi
711 % = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%

```



```

712 \else
713   \caption@Debug{no \protect\prevdepth}%
714   \let\caption@position\@secondoftwo
715 %   = \caption@setposition b%
716 \fi}

```

```

\caption@setautoposition \caption@setautoposition{<position>}
replaces the above algorithm by a different one (or a fixed position setting).
717 \newcommand*\caption@setautoposition[1]{%
718   \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.

```

719 \newcommand*\caption@beginhook{}
720 \newcommand*\caption@endhook{}
721 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
722 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

1.17 Lists

```

723 \DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
724 \DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}

```

```

\caption@setlist \caption@setlist{<boolean>}
725 \newcommand*\caption@setlist{\caption@set@bool\caption@iflist}

```

```

\DeclareCaptionListFormat \DeclareCaptionListFormat{<name>}{<code with #1 and #2>}
726 \newcommand*\DeclareCaptionListFormat[2]{%
727   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
728 \@onlypreamble\DeclareCaptionListFormat

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

```

```

\caption@setlistformat \caption@setlistformat{<name>}
Selecting a caption list format simply means saving the code (in \caption@lstfmt).

```

```

730 \newcommand*\caption@setlistformat[1]{%
731   \@ifundefined{caption@lstfmt@#1}%
732   {\caption@Error{Undefined list format `#1'}}%
733   {\expandafter\let\expandafter\caption@lstfmt
734     \csname caption@lstfmt@#1\endcsname}}

```

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

```

735 \DeclareCaptionListFormat{empty}{}
736 \DeclareCaptionListFormat{simple}{#1#2}
737 \DeclareCaptionListFormat{parens}{#1 (#2)}
738 \DeclareCaptionListFormat{subsimple}{#2}
739 \DeclareCaptionListFormat{subparens}{(#2)}

740 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}

```

1.18 Debug option

```
741 \DeclareCaptionOption{debug}[1]{%
742   \caption@set@bool\caption@ifdebug{#1}%
743   \caption@ifdebug
744     {\def\caption@Debug{\PackageInfo{caption}}}%
745     {\let\caption@Debug@gobble}}
746 \DeclareOption{debug}{\setkeys{caption}{debug}}
747 \setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

1.19.1 The standard L^AT_EX classes

```
748 \caption@CheckCommand\@makecaption{%
749   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
750   \long\def\@makecaption#1#2{%
751     \vskip\abovecaptionskip
752     \sbox\@tempboxa{#1: #2}%
753     \ifdim \wd\@tempboxa >\hsize
754       #1: #2\par
755     \else
756       \global \@minipagefalse
757       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
758     \fi
759     \vskip\belowcaptionskip}}
```

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

```
760 \@ifundefined{@captionheadfont}{}{%
761   \caption@CheckCommand\@makecaption{%
762     % amsart|amsproc|amsbook [2004/08/06 v2.20]
763     \long\def\@makecaption#1#2{%
764       \setbox\@tempboxa\vbox{\color@setgroup
765         \advance\hsize-2\captionindent\noindent
766         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
767           {\@cdr#2\@nil}\@captionfont\upshape\enspace#2}%
768       \unskip\kern-2\captionindent\par
769       \global\setbox\@ne\lastbox\color@endgroup}%
770     \ifhbox\@ne % the normal case
771       \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
772     \fi
773     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
774       \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
775     \else % tempboxa contained more than one line
776       \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
777         \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
778     \fi
779     \ifnum\@tempcnta<64 % if the float IS a figure...
780       \addvspace\abovecaptionskip
781       \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
782     \else % if the float IS NOT a figure...
783       \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
784       \nobreak
785       \vskip\belowcaptionskip
```

```

786     \fi
787     \relax
788   }}

789   \caption@CheckCommand\@makecaption{%
790     % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publiés par
791     \long\def\@makecaption#1#2{%
792       \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
793       \setbox\@tempboxa\vbox{\color@setgroup
794         \advance\hsize-2\captionindent\noindent
795         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
796           {\@cdr#2\@nil}\@addpunct{.}\@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         \@tempdima\wd\@ne\advance\@tempdima-\captionindent
805         \wd\@ne\@tempdima
806       \else % tempboxa contained more than one line
807         \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
808           \unvbox\@tempboxa\parskip\z@skip
809           \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
810       \fi
811       \ifnum\@tempcnta<64 % if the float IS a figure...
812         \addvspace\abovecaptionskip
813         \noindent\kern\captionindent\box\@ne
814       \else % if the float IS NOT a figure...
815         \noindent\kern\captionindent\box\@ne
816         \nobreak
817         \vskip\belowcaptionskip
818       \fi
819     \relax
820   }}

821   \let\captionmargin\captionindent % set to 3pc by AMS class
822   \begingroup\edef\@tempa{\endgroup
823     \noexpand\caption@g@addto@list\noexpand\caption@sty@default
824     {margin=\the\captionmargin
825       \@ifundefined{smf@makecaption}{},{,maxmargin=.1\linewidth}}}}
826   \@tempa
827   \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
828   \DeclareCaptionLabelSeparator{default}{.\enspace}
829   \DeclareCaptionDefaultFont{font}{\@captionfont}
830   \DeclareCaptionDefaultFont{labelfont}{\@captionheadfont}
831   \DeclareCaptionDefaultFont{textfont}{\@captionfont\upshape}
832   \captionsetup[figure]{position=b}
833   \captionsetup[table]{position=t}
834 }

```

1.19.3 The beamer class

```

835 \@ifclassloaded{beamer}{%

```

```

836 \caption@CheckCommand\beamer@makecaption{%
837   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
838   \long\def\beamer@makecaption#1#2{%
839     \def\insertcaptionname{\csname#1name\endcsname}%
840     \def\insertcaptionnumber{\csname the#1\endcsname}%
841     \def\insertcaption{#2}%
842     \nobreak\vskip\abovecaptionskip\nobreak
843     \sbox\@tempboxa{\usebeamertemplate**{caption}}%
844     \ifdim \wd\@tempboxa >\hsize
845       \usebeamertemplate**{caption}\par
846     \else
847       \global \@minipagefalse
848       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
849     \fi
850     \nobreak\vskip\belowcaptionskip\nobreak}}

851 \DeclareCaptionLabelFormat{default}{#1}
852 \DeclareCaptionDefaultJustification{\raggedright}
853 \DeclareCaptionDefaultFont{font}{%
854   \usebeamerfont*{caption}%
855   \usebeamercolor[fg]{caption}}
856 \DeclareCaptionDefaultFont{labelfont}{%
857   \usebeamercolor[fg]{caption name}%
858   \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.)

```

859 \defbeamertemplate{caption}{caption3}{%
860   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}

861 \DeclareOption{beamer}{%
862   % \usebeamertemplate**{caption} will set font
863   \DeclareCaptionDefaultFont{font}{}%
864   \setbeamertemplate{caption}[caption3]}

865 }{}

```

1.19.4 The KOMA-Script classes

```

866 \@ifundefined{scr@caption}{}{%
867   \caption@CheckCommand\@makecaption{%
868     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
869     \long\def\@makecaption#1#2{%
870       \if@captionabove
871         \vskip\belowcaptionskip
872       \else
873         \vskip\abovecaptionskip
874       \fi
875       \@@makecaption\@firstofone{#1}{#2}%
876       \if@captionabove
877         \vskip\abovecaptionskip
878       \else
879         \vskip\belowcaptionskip
880       \fi}}
881 \DeclareCaptionFormat{default}[#1#2#3\par]{%

```

```

882 \ifdofullc@p
883 \caption@ifin@list\caption@lsep@crlist\caption@lsep@name
884 {\caption@Error{%
885     The option 'labelsep=\caption@lsep@name' does not work\MessageBreak
886     with \noexpand\setcaphanging (which is set by default)}}%
887 {\caption@fmt@hang{#1}{#2}{#3}}%
888 \else
889 #1#2%
890 \ifdim\cap@indent<\z@
891 \par
892 \noindent\hspace*{-\cap@indent}%
893 \else\if@capbreak
894 \par
895 \fi\fi
896 #3\par
897 \fi}
898 \DeclareCaptionLabelSeparator{default}{\captionformat}
899 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
900 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
901 }

```

1.19.5 The NTG Dutch classes

```

902 \@ifundefined{CaptionFonts}{}{%
903 \caption@CheckCommand\@makecaption{%
904 % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
905 \long\def\@makecaption#1#2{%
906 \vskip\abovecaptionskip
907 \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
908 \ifdim \wd\@tempboxa >\hsize
909 {\CaptionLabelFont#1:} \CaptionTextFont#2\par
910 \else
911 \global \@minipagefalse
912 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
913 \fi
914 \vskip\belowcaptionskip}}
915 \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
916 \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
917 }

```

1.19.6 The thesis class

```

918 \@ifclassloaded{thesis}{%
919 \caption@CheckCommand\@makecaption{%
920 % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
921 \long\def\@makecaption#1#2{%
922 \vskip\abovecaptionskip
923 \setbox\@tempboxa\hbox{\cph@font #1:} {\cpb@font #2}}%
924 \ifdim \wd\@tempboxa >\hsize
925 \@hangfrom{\cph@font #1:} {\cpb@font #2\par}%
926 \else
927 \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
928 \fi
929 \vskip\belowcaptionskip}}

```

```

930 \DeclareCaptionDefaultFormat{hang}
931 \DeclareCaptionDefaultFont{labelfont}{\cph@font}
932 \DeclareCaptionDefaultFont{textfont}{\cpb@font}
933 {}

```

1.19.7 The frenchb Babel option

```

934 \@ifundefined{FB@makecaption}{}{%
935 \caption@CheckCommand\@makecaption{%
936 % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
937 % frenchb.ldf [2007/10/05 v2.0e French support from the babel system]
938 \long\def\@makecaption#1#2{%
939 \vskip\abovecaptionskip
940 \sbox\@tempboxa{#1\CaptionSeparator #2}%
941 \ifdim \wd\@tempboxa >\hsize
942 #1\CaptionSeparator #2\par
943 \else
944 \global \@minipagefalse
945 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
946 \fi
947 \vskip\belowcaptionskip}}
948 \ifx\@makecaption\STD@makecaption
949 \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
950 \def\caption@frenchb{% supress frenchb warning
951 \let\STD@makecaption\@makecaption
952 \let\FB@makecaption\@makecaption}
953 \else
954 \ifx\@makecaption\@undefined\else
955 \PackageInfo{caption}{%
956 The definition of \protect\@makecaption\space
957 has been changed,\MessageBreak
958 frenchb will NOT customize it}%
959 \fi
960 \fi
961 }

```

1.19.8 The frenchle/pro package

```

962 \@ifundefined{frenchTeXmods}{}{%
963 \caption@CheckCommand\@makecaption{%
964 % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
965 % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
966 \def\@makecaption#1#2{%
967 \ifFTY%
968 \def\@secondofmany##1##2\void{##2}%
969 \def\@tempa{\@secondofmany#2\void}%
970 \ifx\@tempa\empty%
971 \let\captionseparator\empty%
972 \fi%
973 \@mcORI{#1}{\relax\captionfont{#2}}%
974 \else
975 \@mcORI{#1}{#2}%
976 \fi}}

```

```

977 \caption@CheckCommand\@makecaption{%
978   % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
979   \def\@makecaption#1#2{%
980     \ifFTY%
981       \def\@secondofmany##1##2\void{##2}%
982       \protected@edef\@tempa{\@secondofmany#2\void}%
983       \ifx\@tempa\empty%
984         \let\captionseparator\empty%
985       \fi%
986       \@mcORI{#1}{\relax\captionfont{#2}}%
987     \else
988       \@mcORI{#1}{#2}%
989     \fi}}
990 \DeclareCaptionDefaultFont{textfont}{\itshape}%
991 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
992 }

```

1.20 Execution of options

```

993 \captionsetup{style=default,position=default,%
994               list,listformat=default,twoside=\if@twoside 1\else 0\fi}
995 \ProcessOptions*
996 \caption@IfCheckCommand{%
997   \caption@setbool{documentclass}{1}%
998 }{%
999   \caption@setbool{documentclass}{0}%
1000 \PackageInfo{caption}{%
1001   Unknown document class (or package),\MessageBreak
1002   standard defaults will be used}%
1003 \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1004 }

```

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.

```

1005 \newcommand*\caption@addcontentsline[2]{%
1006   \caption@iflist
1007     {\def\@tempa{#2}}%
1008     {\let\@tempa\@empty}%
1009   \ifx\@tempa\@empty \else
1010     {\let\@tempa\space
1011      \addcontentsline{\csname ext@#1\endcsname}{#1}%
1012                      {\protect\numberline
1013                       {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1014                       {\ignorespaces #2}}}%
1015   \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`.)

```

1016 \newif\ifcaption@star

\caption@fnum \caption@fnum{<float type>}
Typesets the caption label; as replacement for \fnum{<float type>}.
1017 \newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}

\caption@make \caption@make{<float name>}{<ref. number>}{<text>}
Typesets the caption.
1018 \newcommand\caption@make[2]{\caption@@make{\caption@lfmt{#1}{#2}}}

\caption@@make \caption@@make{<caption label>}{<caption text>}
1019 \newcommand\caption@@make[2]{%
1020 \begingroup
1021 \caption@stepcounter
1022 \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
1023 \ifx\caption@maxmargin\undefined \else
1024 \ifdim\captionmargin>\caption@maxmargin\relax
1025 \captionmargin\caption@maxmargin\relax
1026 \fi
1027 \fi
1028 \ifx\caption@minmargin\undefined \else
1029 \ifdim\captionmargin<\caption@minmargin\relax
1030 \captionmargin\caption@minmargin\relax
1031 \fi
1032 \fi

Special single-line treatment (option singlelinecheck=)
1033 \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%

Typeset the left margin (option margin=)
1034 \caption@calcmargin
1035 \@tempdima\captionmargin
1036 \ifdim\captionmargin@=\z@ \else
1037 \caption@ifoddpagel{\advance\@tempdima\captionmargin@}%
1038 \fi
1039 \caption@ifh{\advance\@tempdima\caption@indent}%
1040 \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*.

1041 \@tempdima\captionwidth
1042 \caption@ifh{\advance\@tempdima-\caption@indent}%
1043 \caption@parbox\@tempdima{%

Typeset the indentation (option indentation=)
Bugfix 04-05-05: \hspace-\caption@indent replaced by \ifdim\caption@indent=\z@...
1044 \caption@ifh{%
1045 \ifdim\caption@indent=\z@
1046 \leavevmode
1047 \else
1048 \hspace-\caption@indent
1049 \fi}%

```


Typeset the caption itself and close the \caption@parbox

```
1050 \caption@@@make{#1}{#2}}%
```

Typeset the right margin (option margin=)

```
1051 \@tempdima\captionmargin
1052 \ifdim\captionmargin@=\z@ \else
1053 \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
1054 \fi
1055 \hspace\@tempdima
1056 \caption@endhook
1057 \endgroup
1058 \global\caption@starfalse}
```

\caption@calcmargin \caption@calcmargin

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

```
1059 \newcommand*\caption@calcmargin{%
1060 \ifdim\captionwidth=\z@
1061 \captionwidth\linewidth
1062 \advance\captionwidth by -2\captionmargin
1063 \advance\captionwidth by -\captionmargin@
1064 \else
1065 \captionmargin\linewidth
1066 \advance\captionmargin by -\captionwidth
1067 \divide\captionmargin by 2
1068 \captionmargin@\z@
1069 \fi
1070 \caption@Debug{%
1071 \string\hsize=\the\hsize,
1072 \string\linewidth=\the\linewidth,\MessageBreak
1073 \string\leftmargin=\the\leftmargin,
1074 \string\rightmargin=\the\rightmargin,\MessageBreak
1075 \string\margin=\the\captionmargin,
1076 \string\margin@=\the\captionmargin@,
1077 \string\width=\the\captionwidth}%
1078 }
```

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}

This one does the single-line-check.

```
1079 \newcommand\caption@slc[4]{%
1080 \caption@Debug{Begin SLC}%
1081 \begingroup
1082 \caption@singleline
1083 \let\caption@hj\@empty
1084 \caption@calcmargin % calculate #3 if necessary
1085 \caption@prepareslc
1086 \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1087 \ifdim\wd\@tempboxa>#3%
1088 \endgroup
1089 \else
1090 \endgroup
1091 \caption@singleline
1092 #4%
```

```

1093 \fi
1094 \caption@Debug{End SLC}}

1095 \newcommand*\caption@singleline{%
1096 \caption@xsetup\caption@opt@singleline
1097 \let\caption@fmt\caption@slfmt}

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1098 \newcommand*\caption@prepareslc{%
1099 \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1100 \def\label{\caption@withoptargs\@gobbletwo}%
1101 \let\stepcounter\caption@l@stepcounter
1102 \let\refstepcounter\stepcounter\let\H\refstepcounter\stepcounter}
1103 \newcommand*\caption@l@stepcounter[1]{\advance\c@#1\endcsname\@ne\relax}

\caption@parbox \caption@parbox{\width}{\contents}
This macro defines the box which surrounds the caption paragraph.
1104 \newcommand*\caption@parbox{\parbox[b]}

\caption@@@make \caption@@@make{\caption label}{\caption text}
This one finally typesets the caption paragraph, without margin and indentation.
1105 \newcommand\caption@@@make[2]{%
If the label is empty, we use no caption label separator.
1106 \sbox\@tempboxa{#1}%
1107 \ifdim\wd\@tempboxa=\z@
1108 \let\caption@lsep\relax
1109 % \@capbreakfalse
1110 \fi
If the text is empty, we use no caption label separator, too.
1111 \caption@ifempty{#2}{%
1112 \let\caption@lsep\relax
1113 % \@capbreakfalse
1114 \let\caption@ifstrut\@secondoftwo
1115 }%
Take care that \caption@parindent and \caption@hangindent will be used
to typeset the paragraph.
1116 \@setpar{\@par\caption@@par}\caption@@par
Finally typeset the caption.
1117 \caption@hj\captionfont\captionsize\caption@fmt
1118 {\ifcaption@star\else\captionlabelfont#1}\fi}%
1119 {\ifcaption@star\else\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1120 {\captiontextfont
1121 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}%
1122 \nobreak\hskip\z@skip % enable hyphenation
1123 \caption@tfmt{#2}%
1124 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}%
1125 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}%
1126 \par}}

```

`\caption@ifempty` `\caption@ifempty{<text>}{<true>}{(no <false>)}`

This one tests if the *<text>* is actually empty.

Note: This will be done without expanding the text, therefore this is far away from being bullet-proof.

Note: This macro is re-defining itself so only the first test (in a group) will actually be done.

```

1127 \newcommand\caption@ifempty[1]{%
1128   \caption@if@empty{#1}%
1129   \caption@ifempty@unused}

1130 \newcommand\caption@if@empty[1]{%
1131   \def\caption@tempa{#1}%
1132   \ifx\caption@tempa\@empty
1133     \let\caption@ifempty\@secondoftwo
1134   \else
1135     \expandafter\def\expandafter\caption@tempa\expandafter{%
1136       \caption@car#1\caption@if@empty\caption@nil}%
1137     \def\caption@tempb{\caption@if@empty}%
1138     \ifx\caption@tempa\caption@tempb
1139       \let\caption@ifempty\@secondoftwo
1140     \else
1141       \def\caption@tempb{\ignorespaces}%
1142       \ifx\caption@tempa\caption@tempb
1143         \expandafter\caption@if@empty\expandafter{\@gobble#1}%
1144       \else
1145         \def\caption@tempb{\label}%
1146         \ifx\caption@tempa\caption@tempb
1147           \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1148         \else
1149           \def\caption@tempb{\index}%
1150           \ifx\caption@tempa\caption@tempb
1151             \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1152           \else
1153             \def\caption@tempb{\glossary}%
1154             \ifx\caption@tempa\caption@tempb
1155               \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1156             \else
1157               \let\caption@ifempty\@gobbletwo
1158             \fi
1159           \fi
1160         \fi
1161       \fi
1162     \fi
1163   \fi}

1164 \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.

```

1165 \newcommand*\caption@@par{%
1166   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

1.23 Types & sub-types

`\DeclareCaptionType` `\DeclareCaptionType[<options>]{<environment>}[<name>][<list name>]`

```

1167 \newcommand*{\DeclareCaptionType{%
1168   \@testopt\@DeclareCaptionType{}}
1169 \@onlypreamble\DeclareCaptionType

1170 \def\@DeclareCaptionType[#1]#2{%
1171   \def\caption@type{#2}%
1172   \caption@Debug{New type `#2'}%
1173   \newcounter{#2}\@namedef{theH#2}{}%
1174   \KV@caption@DCT@within\caption@within@default
1175   \KV@caption@DCT@placement{tbp}%
1176   \@ifundefined{c@float@type}%
1177     {\newcounter{float@type}%
1178      \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1179     {}%
1180   \caption@Debug{float type `#2'=\the\value{float@type}}%
1181   \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1182   \addtocounter{float@type}{\value{float@type}}%
1183   \KV@caption@DCT@fileext{lo#2}%
1184   \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1185   \newenvironment{#2}{\@float{#2}}{\end@float}%
1186   \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1187   \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1188   \@ifundefined{l@figure}%
1189     {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1190     {\expandafter\let\csname l@#2\endcsname\l@figure}%

1191   \expandafter\newcommand\csname #2name\endcsname{}%
1192   \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
1193   \uppercase\expandafter{\@tempa}%
1194   \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
1195   \@tempb
1196   \expandafter\let\csname #2name\endcsname\@tempa
1197   \expandafter\newcommand\csname list#2name\endcsname{}%
1198   \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%

1199   \@cons\caption@typelist{{#2}}%
1200   \caption@setkeys[caption]{caption@DCT}{#1}%

1201   \@ifundefined{float@exts}{\newtoks\float@exts}{}%
1202   \let\float@do=\relax
1203   \edef\@tempa{\noexpand\float@exts{\the\float@exts\float@do{\@nameuse{ext@#2}}}}%
1204   \@tempa
1205   \@ifundefined{float@addtolists}{%
1206     \newcommand\float@addtolists[1]{%
1207       \def\float@do###1{\addtocontents{###1}{##1}}\the\float@exts}%
1208     \@ifundefined{@chapter}{}{\caption@PatchChapter}}{}%

1209   \@ifnextchar[{\@DeclareCaptionType\relax}
1210 \@onlypreamble\@DeclareCaptionType

1211 \def\@@DeclareCaptionType[#1]{%
1212   \KV@caption@DCT@name{#1}%
1213   \@ifnextchar[{\@@DeclareCaptionType\relax}
1214 \@onlypreamble\@@DeclareCaptionType

1215 \def\@@@DeclareCaptionType[#1]{%
1216   \KV@caption@DCT@listname{#1}}
1217 \@onlypreamble\@@@DeclareCaptionType

```

```

1218 \let\DeclareFloatingEnvironment\DeclareCaptionType % old command name
1219 \@onlypreamble\DeclareFloatingEnvironment

\caption@within@default The default ‘within’ value.
1220 \newcommand*\caption@within@default{\@ifundefined{c@chapter}{none}{chapter}}
1221 \@onlypreamble\caption@within@default

\caption@listof \caption@listof{<float type>}
1222 \newcommand*\caption@listof[1]{%
1223   \begingroup
1224     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1225     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1226     \let\caption@ORI@starttoc\@starttoc
1227     \renewcommand*\@starttoc[1]{%
1228       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1229     \listoffigures
1230   \endgroup}

\caption@typelist An \@elt-list containing the caption types defined with \DeclareCaptionType.
1231 \newcommand*\caption@typelist{}

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

1232 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
1233 \@onlypreamble@key{caption@DCT}{fileext}
1234 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1235 \@onlypreamble@key{caption@DCT}{listname}
1236 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1237 \@onlypreamble@key{caption@DCT}{name}
1238 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
1239 \@onlypreamble@key{caption@DCT}{placement}
1240 \define@key{caption@DCT}{within}{%
1241   \@ifundefined{c@chapter}{\@removefromreset\caption@type{chapter}}%
1242   \@removefromreset\caption@type{section}%
1243   \begingroup
1244     \caption@setkeys[caption]{caption@within}{#1}%
1245   \endgroup}
1246 \@onlypreamble@key{caption@DCT}{within}
1247 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1248 \@onlypreamble@key{caption@DCT}{without}

1249 \define@key{caption@within}{none}[]{%
1250   \caption@within{}}
1251 \@onlypreamble@key{caption@within}{none}
1252 \@ifundefined{c@chapter}{%
1253   \define@key{caption@within}{section}[]{%
1254     \@addtoreset\caption@type{section}%
1255     \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
1256 }{%
1257   \define@key{caption@within}{chapter}[]{%
1258     \@addtoreset\caption@type{chapter}%
1259     \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
1260 \@onlypreamble@key{caption@within}{chapter}
1261 \define@key{caption@within}{section}[]{%

```

```

1262 \addtoreset\caption@type{chapter}%
1263 \addtoreset\caption@type{section}%
1264 \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi
1265 \ifnum\c@section>\z@ \thesection.\fi}{%
1266 \theHchapter.\theHsection.}}
1267 }\@onlypreamble@key{caption@within}{section}

\caption@within \caption@within{<the code>}{<theHcode>}
1268 \newcommand*\caption@within{%
1269 \expandafter\caption@within@\expandafter{\caption@type}}
1270 \@onlypreamble\caption@within
1271 \newcommand*\caption@within@[3]{%
1272 \global\@namedef{the#1}{#2\arabic{#1}}}%
1273 \@ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
1274 {\global\@namedef{theH#1}{#3\arabic{#1}}}%
1275 \@onlypreamble\caption@within@

\@removefromreset This code was taken from the remreset package which is part of the ‘carlisle’ package
bundle. (Copyright 1997 David Carlisle)
1276 \providecommand*\@removefromreset[2]{%
1277 \expandafter\let\csname c@#1\endcsname\@removefromreset
1278 \def\@elt##1{%
1279 \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1280 \else
1281 \noexpand\@elt{##1}%
1282 \fi}%
1283 \expandafter\xdef\csname cl@#2\endcsname{%
1284 \csname cl@#2\endcsname}}

\caption@PatchChapter We try to patch \chapter so \float@addtolists will be supported. (Note: The
KOMAScript classes already support \float@addtolists.)
1285 \newcommand*\caption@PatchChapter{%
1286 \providecommand*\@chapterlistsgap{10\p@}%

1287 % report.cls [2005/09/16 v1.4f Standard LaTeX document class]
1288 \caption@patchchapter{report}{%
1289 \ifnum \c@secnumdepth >\m@ne
1290 \refstepcounter{chapter}%
1291 \typeout{\@chapapp\space\thechapter.}%
1292 \addcontentsline{toc}{chapter}%
1293 {\protect\numberline{\thechapter}##1}%
1294 \else
1295 \addcontentsline{toc}{chapter}{##1}%
1296 \fi
1297 \chaptermark{##1}%
1298 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1299 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1300 \if@twocolumn
1301 \topnewpage[\@makechapterhead{##2}]%
1302 \else
1303 \makechapterhead{##2}%
1304 \afterheading
1305 \fi
1306 }{%

```

```

1307 \ifnum \c@secnumdepth >\m@ne
1308 \refstepcounter{chapter}%
1309 \typeout{\@chapapp\space\thechapter.}%
1310 \addcontentsline{toc}{chapter}%
1311 {\protect\numberline{\thechapter}##1}%
1312 \else
1313 \addcontentsline{toc}{chapter}{##1}%
1314 \fi
1315 \chaptermark{##1}%
1316 \ifdim \@chapterlistsgap>\z@
1317 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1318 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1319 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1320 \fi
1321 \if@twocolumn
1322 \topnewpage[\@makechapterhead{##2}]%
1323 \else
1324 \makechapterhead{##2}%
1325 \afterheading
1326 \fi}%

1327 % book.cls [2005/09/16 v1.4f Standard LaTeX document class]
1328 \caption@patch@chapter{book}{%
1329 \ifnum \c@secnumdepth >\m@ne
1330 \if@mainmatter
1331 \refstepcounter{chapter}%
1332 \typeout{\@chapapp\space\thechapter.}%
1333 \addcontentsline{toc}{chapter}%
1334 {\protect\numberline{\thechapter}##1}%
1335 \else
1336 \addcontentsline{toc}{chapter}{##1}%
1337 \fi
1338 \else
1339 \addcontentsline{toc}{chapter}{##1}%
1340 \fi
1341 \chaptermark{##1}%
1342 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1343 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1344 \if@twocolumn
1345 \topnewpage[\@makechapterhead{##2}]%
1346 \else
1347 \makechapterhead{##2}%
1348 \afterheading
1349 \fi
1350 }{%
1351 \ifnum \c@secnumdepth >\m@ne
1352 \if@mainmatter
1353 \refstepcounter{chapter}%
1354 \typeout{\@chapapp\space\thechapter.}%
1355 \addcontentsline{toc}{chapter}%
1356 {\protect\numberline{\thechapter}##1}%
1357 \else
1358 \addcontentsline{toc}{chapter}{##1}%
1359 \fi
1360 \else

```

```

1361     \addcontentsline{toc}{chapter}{##1}%
1362     \fi
1363     \chaptermark{##1}%
1364     \ifdim \@chapterlistsgap>\z@
1365         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1366         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1367         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1368     \fi
1369     \if@twocolumn
1370         \@topnewpage[\@makechapterhead{##2}]%
1371     \else
1372         \@makechapterhead{##2}%
1373         \@afterheading
1374     \fi}%

1375 % amsbook.cls [2004/08/06 v2.20]
1376 % smfbook.cls [1999/11/15 v1.2f Classe LaTeX pour les monographies editees par
1377 \caption@patch@chapter{ams/smfbook}{%
1378     \refstepcounter{chapter}%
1379     \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1380     \else \let\@secnumber\thechapter \fi
1381     \typeout{\chaptername\space\@secnumber}%
1382     \def\@toclevel{0}%
1383     \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1384     \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1385     \chaptermark{##1}%
1386     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1387     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1388     \@makechapterhead{##2}\@afterheading
1389 }{%
1390     \refstepcounter{chapter}%
1391     \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1392     \else \let\@secnumber\thechapter \fi
1393     \typeout{\chaptername\space\@secnumber}%
1394     \def\@toclevel{0}%
1395     \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1396     \else \@tocwriteb\tocchapter{chapter}{##2}\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     \@makechapterhead{##2}\@afterheading}%

1404 % scrreprt/scrbook.cls
1405 \@ifundefined{KOMAClassName}{}{%
1406     \caption@Debug{document class '\KOMAClassName' detected}%
1407     \let\caption@patch@chapter\@gobblethree}%

1408 % rapport1/3.cls [2004/06/07 v2.1a NTG LaTeX document class]
1409 \caption@patch@chapter{rapport}{%
1410     \ifnum \c@secnumdepth >\m@ne
1411         \refstepcounter{chapter}%
1412         \typeout{\@chapapp\space\thechapter.}%
1413         \addcontentsline{toc}{chapter}%

```



```

1414         {\protect\numberline{\thechapter}\toc@font0 ##1}%
1415     \else
1416         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1417     \fi
1418     \chaptermark{##1}%
1419     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1420     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1421     \if@twocolumn
1422         \topnewpage[\@makechapterhead{##2}]%
1423     \else
1424         \@makechapterhead{##2}%
1425         \@afterheading
1426     \fi
1427 }{%
1428     \ifnum \c@secnumdepth >\m@ne
1429         \refstepcounter{chapter}%
1430         \typeout{\@chapapp\space\thechapter.}%
1431         \addcontentsline{toc}{chapter}%
1432             {\protect\numberline{\thechapter}\toc@font0 ##1}%
1433     \else
1434         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1435     \fi
1436     \chaptermark{##1}%
1437     \ifdim \@chapterlistsgap>\z@
1438         \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1439         \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1440         \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1441     \fi
1442     \if@twocolumn
1443         \topnewpage[\@makechapterhead{##2}]%
1444     \else
1445         \@makechapterhead{##2}%
1446         \@afterheading
1447     \fi}%
1448 % boek(3).cls [2004/06/07 v2.1a NTG LaTeX document class]
1449 \caption@patch@chapter{boek}{%
1450     \ifnum \c@secnumdepth >\m@ne
1451         \if@mainmatter
1452             \refstepcounter{chapter}%
1453             \typeout{\@chapapp\space\thechapter.}%
1454             \addcontentsline{toc}{chapter}%
1455                 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1456         \else
1457             \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1458         \fi
1459     \else
1460         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1461     \fi
1462     \chaptermark{##1}%
1463     \addtocontents{lof}{\protect\addvspace{10\p@}}%
1464     \addtocontents{lot}{\protect\addvspace{10\p@}}%
1465     \if@twocolumn
1466         \topnewpage[\@makechapterhead{##2}]%
1467     \else

```

```

1468     \@makechapterhead{##2}%
1469     \@afterheading
1470   \fi
1471 }{%
1472   \ifnum \c@secnumdepth >\m@ne
1473     \if@mainmatter
1474       \refstepcounter{chapter}%
1475       \typeout{\@chapapp\space\thechapter.}%
1476       \addcontentsline{toc}{chapter}%
1477         {\protect\numberline{\thechapter}\toc@font0 ##1}%
1478     \else
1479       \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1480     \fi
1481   \else
1482     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1483   \fi
1484   \chaptermark{##1}%
1485   \ifdim \@chapterlistsgap>\z@
1486     \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1487     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1488     \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1489   \fi
1490   \if@twocolumn
1491     \@topnewpage[\@makechapterhead{##2}]%
1492   \else
1493     \@makechapterhead{##2}%
1494     \@afterheading
1495   \fi}%
1496 % thesis.cls [1996/25/01 1.0g LaTeX document class (wm).]
1497 \caption@patch@chapter{thesis}{%
1498   \ifnum \c@secnumdepth >\m@ne
1499     \if@mainmatter
1500       \refstepcounter{chapter}%
1501       \typeout{\chaptername\space\thechapter.}
1502       \if@thema
1503         \ifx\@shortauthor\@empty
1504           \addcontentsline{toc}{chapter}{%
1505             \protect\numberline{\thechapter.}##1}%
1506         \else
1507           \addcontentsline{toc}{chapter}{%
1508             \protect\numberline{\thechapter.}%
1509             \@shortauthor\hfill\mbox{}}\vskip\normallineskip ##1}%
1510         \fi
1511       \else
1512         \addcontentsline{toc}{chapter}{%
1513           \protect\numberline{\thechapter.}##1}%
1514       \fi
1515     \else
1516       \addcontentsline{toc}{chapter}{##1}
1517     \fi
1518   \else
1519     \addcontentsline{toc}{chapter}{##1}
1520   \fi
1521   \chaptermark{##1}

```

```

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

1566 \ifx\caption@patch@chapter\@gobblethree \else
1567 \caption@Debug{%
1568 Unsupported document class detected,\MessageBreak
1569 or \noexpand\@chapter was redefined by another package}%
1570 \fi
1571 \let\caption@PatchChapter\@undefined
1572 \@onlypreamble\caption@PatchChapter

1573 \newcommand\caption@patch@chapter[3]{%
1574 \begingroup

```

```

1575 % \let\if@twocolumn\iffalse
1576 \let\if@mainmatter\iffalse
1577 \let\if@thema\iffalse
1578 \def\@tempa[##1]##2{#2}%
1579 \ifx\@tempa\@chapter
1580 \caption@Debug{document class ‘#1’ detected}%
1581 \gdef\@chapter[##1]##2{#3}%
1582 \global\let\caption@patch@chapter\@gobblethree
1583 \fi
1584 \endgroup}
1585 \@onlypreamble\caption@patch@chapter
1586 \long\def \@gobblethree #1#2#3{}

```

`\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>*.

```

1587 \newcommand*\DeclareCaptionSubType{%
1588 \caption@teststar\@DeclareCaptionSubType\@firstoftwo\@secondoftwo}
1589 \@onlypreamble\DeclareCaptionSubType

1590 \newcommand*\@DeclareCaptionSubType[1]{%
1591 \@testopt{\@@DeclareCaptionSubType{#1}}{alph}}
1592 \@onlypreamble\@DeclareCaptionSubType

1593 \def\@@DeclareCaptionSubType#1[#2]#3{%
1594 \@ifundefined{c@#3}%
1595 {\caption@Error{No float type ‘#3’ defined}}%
1596 {\@ifundefined{c@sub#3}%
1597 {\caption@Debug{New subtype ‘sub#3’}%
1598 \newcounter{sub#3}%
1599 \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1600 \@ifundefined{l@chapter}%
1601 {\edef\@tempa{\expandafter\expandafter\expandafter\noexpand
1602 \expandafter\@car\l@subsubsection\@nil}%
1603 \def\@tempb{\@dottedtocline}%
1604 \ifx\@tempa\@tempb % \l@subsubsection starts with \@dottedtocline
1605 \expandafter\edef\csname l@sub#3\endcsname{%
1606 \noexpand\@dottedtocline{2}%
1607 \expandafter\expandafter\expandafter\noexpand
1608 \expandafter\@gobbletwo\l@subsubsection}%
1609 \else
1610 \@namedef{l@sub#3}{\@dottedtocline{2}{3.8em}{3.2em}}%
1611 \fi}%
1612 {\expandafter\let\csname l@sub#3\endcsname\l@subsection}%
1613 \@cons\caption@subtypelist{{#3}}}%
1614 {\caption@Debug{Modify caption ‘sub#3’}}%

1615 \@namedef{sub#3name}{}%
1616 \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
1617 #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
1618 {\@namedef{p@sub#3}}}%
1619 \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1620 {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1621 \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%

```

```

1622     \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}%
1623     }}
1624 \@onlypreamble\@@DeclareCaptionSubType
\caption@subtypelist An \@elt-list containing the subtypes defined with \DeclareCaptionSubType.
1625 \newcommand*\caption@subtypelist{}
\caption@For \caption@For{<elt-list>}{<code with #1>}
\caption@For*{<elt-list>}{<code with #1>}
1626 \newcommand*\caption@For{\caption@withoptargs\caption@For}
1627 \@onlypreamble\caption@For
1628 \newcommand\caption@@For[3]{%
1629   \caption@AtBeginDocument#1{%
1630     \def\@elt##1{#3}%
1631     \@nameuse{caption@#2}%
1632     \let\@elt\relax}%
1633 \@onlypreamble\caption@@For

```

1.24 subfig package adaption

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

```

1634 \caption@AtBeginDocument{%
1635   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1636   \ifx\subref\@tempa
1637     \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1638     \let\caption@setfloattype\@gobble
1639     \let\@dottedxxxline\sf@NEW@dottedxxxline
1640     \let\sf@subfloat\sf@NEW@subfloat

```

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

```

1641   \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1642   \fi
1643   \let\sf@NEW@dottedxxxline\@undefined
1644   \let\sf@NEW@subfloat\@undefined}
1645 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1646   \begingroup
1647     \caption@setfloattype{#1}%
1648     \caption@setoptions{subfloat}%
1649     \caption@setoptions{sub#1}%
1650     \ifnum #3>\@nameuse{c@#2depth}\else
1651       \@dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1652     \fi
1653   \endgroup}
1654 \def\sf@NEW@subfloat{%
1655   \begingroup
1656     \caption@setfloattype\@capttype
1657     \sf@ifpositiontop{%
1658       \maincaptiontoptrue
1659     }{%
1660       \maincaptiontopfalse

```

```

1661 }%
1662 \caption@setoptions{subfloat}%
1663 \caption@setoptions{sub\@captive}%
1664 \let\sfo@oldlabel=\label
1665 \let\label=\subfloat@label
1666 \ifmaincaptiontop\else
1667   \advance\@nameuse{c@\@captive}\@ne
1668 \fi
1669 \refstepcounter{sub\@captive}%
1670 \setcounter{sub\@captive @save}{\value{sub\@captive}}%
1671 \@ifnextchar [% %] match left bracket
1672   {\sf@@subfloat}%
1673   {\sf@@subfloat[\@empty]}}

```

2 Main package

2.1 Identification

```
1674 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1675 \ProvidesPackage{caption}[2008/08/24 v3.1j Customizing captions (AR)]
1676 %\@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.

1677 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1678 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1679 \RequirePackage{caption3}[2008/08/24] % needs v3.1j or newer
```

2.3 Check against incompatible packages

```
1680 \@ifpackageloaded{caption2}{%
1681   \caption@Error{%
1682     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1683     the (current) caption package}%
1684   \endinput
1685 }{}

1686 \caption@AtBeginDocument{%
1687   \@ifpackageloaded{ftcap}{\caption@DisablePositionOption{ftcap}}{}%
1688   \@ifpackageloaded{nonfloat}{\caption@DisablePositionOption{nonfloat}}{}%
1689   \@ifpackageloaded{topcapt}{\caption@DisablePositionOption{topcapt}}{}%
\caption@DisablePositionOption \caption@DisablePositionOption{package}
disables the 'position' option.

1690 \newcommand*\caption@DisablePositionOption[1]{%
1691   \caption@Info{%
1692     '#1' package detected; setting 'position=b' for compatibility reasons}%
1693   \caption@setposition b%

1694   \DeclareCaptionOption{position}{%
1695     \caption@Error{Usage of the 'position' option is incompatible\MessageBreak
1696       to the '#1' package}}%
1697 \@onlypreamble\caption@DisablePositionOption
```

2.4 Check document class

```
1698 \caption@ifbool{documentclass}{}{%
1699   \caption@WarningNoLine{%
1700     Unsupported document class (or package) detected,\MessageBreak
1701     usage of the caption package is not recommended}%
1702   \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1703 }
```

2.5 Adaption to the $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes

```
1704 \@ifundefined{@captionheadfont}{}{%
1705   \caption@Info{AMS or SMF document class}%
```

```

1706 \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1707 }

```

2.6 Emulation of the KOMA-Script commands

```

1708 \@ifundefined{scr@caption}{}{%
1709 \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 1710 \g@addto@macro\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1711 \g@addto@macro\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1712 \if@tablecaptionabove
1713 \@tablecaptionabovetrue
1714 \else
1715 \@tablecaptionabovefalse
1716 \fi

\onelinecaptionstrue
\onelinecaptionfalse 1717 \g@addto@macro\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1718 \g@addto@macro\onelinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1719 \ifonelinecaptions
1720 \onelinecaptionstrue
1721 \else
1722 \onelinecaptionfalse
1723 \fi

\@captionabovetrue
\@captionabovefalse 1724 \g@addto@macro\@captionabovetrue{\let\caption@position\@firstoftwo}
1725 \g@addto@macro\@captionabovefalse{\let\caption@position\@secondoftwo}

\setcapindent
1726 \let\caption@KOMA@setcapindent\@setcapindent
1727 \renewcommand*\@setcapindent[1]{%
1728 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

1729 \let\caption@KOMA@@setcapindent\@setcapindent
1730 \renewcommand*\@@setcapindent[1]{%
1731 \caption@KOMA@@setcapindent{#1}\caption@setcapindent}

1732 \newcommand*\caption@setcapindent{%
1733 \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}

1734 \@ifundefined{cap@indent}{}{\caption@setcapindent}

\setcapwidth Note: The optional argument of \setcapwidth if not supported (yet), so we issue a warning if
used. (Since this does not seem to have an negative effect when used by the captionbeside
environment, we suppress the warning here.)

1735 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1736 \csname\string\setcapwidth\endcsname
1737 \@namedef{\string\setcapwidth}[1]#2{%
1738 \caption@KOMA@setcapwidth[1]{#2}\caption@setcapwidth{#1}}

```



```

1739 \newcommand*\caption@setcapwidth[1]{%
1740   \ifx\#1\\\else
1741     \@ifundefined{cap@margin}{}{%
1742       \def\@tempa{captionbeside}%
1743       \ifx\@tempa\@currenvir\else\caption@Warning{%
1744         Ignoring optional argument [#1] of \string\setcapwidth\MessageBreak}%
1745       \fi}%
1746   \fi
1747   \captionsetup{width=\cap@width}}

1748 \def\caption@tempa{\hsize}%
1749 \ifx\caption@tempa\cap@width \else
1750   \caption@setcapwidth{?}
1751 \fi

\setcapmargin
1752 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1753   \csname\string\@setcapmargin\endcsname
1754 \@namedef{\string\@setcapmargin}[#1]#2{%
1755   \caption@KOMA@setcapmargin[#1]#2\caption@setcapmargin}

1756 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1757   \csname\string\@@setcapmargin\endcsname
1758 \@namedef{\string\@@setcapmargin}[#1]#2{%
1759   \caption@KOMA@@setcapmargin[#1]#2\caption@setcapmargin}

1760 \newcommand*\caption@setcapmargin{%
1761   \begingroup
1762   \let\onelinecaptionsfalse\relax
1763   \def\@twoside{0}%
1764   \def\if@twoside{\def\@twoside{1}\iffalse}%
1765   \cap@margin
1766   \def\@tempa{\endgroup}%
1767   \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1768     \def\hspace##1##{\@firstofone}%
1769     \edef\@tempa{\endgroup
1770       \noexpand\captionsetup{%
1771         twoside=\@twoside,slc=0,%
1772         margin={\cap@left,\cap@right}}}%
1773   \fi\fi
1774   \@tempa}

1775 \ifx\cap@margin\relax \else
1776   \caption@setcapmargin
1777 \fi
1778 }

```

2.7 Declaration of options

2.7.1 Options for figure and table

```

1779 \DeclareCaptionOption{figureposition}{%
1780   \captionsetup*[figure]{position=#1}}
1781 \@onlypreamble@key{caption}{figureposition}

1782 \DeclareCaptionOption{tableposition}{%

```

```

1783 \captionsetup*[table]{position=#1}
1784 \@onlypreamble@key{caption}{tableposition}

1785 \DeclareCaptionOption{figurename}{\caption@SetName{figure}{#1}}
1786 \DeclareCaptionOption{tablename}{\caption@SetName{table}{#1}}
1787 \DeclareCaptionOption{name}{\caption@setname\@capttype{#1}}

1788 \DeclareCaptionOption{listfigurename}{\caption@SetName{listfigure}{#1}}
1789 \DeclareCaptionOption{listtablename}{\caption@SetName{listtable}{#1}}

\caption@SetName \caption@SetName{<cmd>}{<value>}
1790 \newcommand*\caption@SetName[2]{%
1791 \caption@setname{#1}{#2}%
1792 \begin{group}
1793 \ifundefined{language}{\language}{%
1794 \ifundefined{captions\language}{\language}{%
1795 \expandafter\g@addto@macro\csname captions\language\endcsname
1796 {\caption@setname{#1}{#2}}}%
1797 \end{group}}

1798 \newcommand*\caption@setname[2]{\@namedef{#1name}{#2}}
1799 \caption@AtBeginDocument{\let\caption@SetName\caption@setname}

1800 \DeclareCaptionOption{figurewithin}{\caption@Within{figure}{#1}}
1801 \@onlypreamble@key{caption}{figurewithin}
1802 \DeclareCaptionOption{figurewithout}{\KV@caption@figurewithin{none}}
1803 \@onlypreamble@key{caption}{figurewithout}

1804 \DeclareCaptionOption{tablewithin}{\caption@Within{table}{#1}}
1805 \@onlypreamble@key{caption}{tablewithin}
1806 \DeclareCaptionOption{tablewithout}{\KV@caption@tablewithin{none}}
1807 \@onlypreamble@key{caption}{tablewithout}

1808 \DeclareCaptionOption{within}{%
1809 \ifundefined{c@figure}{\caption@Within{figure}{#1}}%
1810 \ifundefined{c@table}{\caption@Within{table}{#1}}%
1811 \def\caption@within@default{#1}}
1812 \@onlypreamble@key{caption}{within}
1813 \DeclareCaptionOption{without}{\KV@caption@within{none}}
1814 \@onlypreamble@key{caption}{without}

\caption@within
1815 \newcommand*\caption@Within[1]{\def\caption@type{#1}\KV@caption@DCT@within}
1816 \@onlypreamble\caption@Within

```

2.7.2 Miscellaneous options

```

1817 \DeclareCaptionOption*{config}[caption]{%
1818 \InputIfFileExists{#1.cfg}%
1819 {\typeout{*** Local configuration file #1.cfg used ***}}%
1820 {\caption@Warning{Configuration file #1.cfg not found}}}

1821 \DeclareCaptionOption{@minipage}{%
1822 \caption@ifinlist{#1}{auto,default}%
1823 {\let\caption@if@minipage\@gobbletwo}%
1824 {\caption@set@bool\caption@if@minipage{#1}}}
1825 \captionsetup{@minipage=default}

```

2.7.3 caption v1.x compatibility options

```
1826 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1827 \@onlypreamble@key{caption}{compatibility}

1828 \DeclareCaptionOptionNoValue*{normal}{%
1829   \caption@setformat{plain}%
1830   \caption@setjustification{justified}}
1831 \DeclareCaptionOptionNoValue*{isu}{%
1832   \caption@setformat{hang}%
1833   \caption@setjustification{justified}}
1834 \DeclareCaptionOptionNoValue*{hang}{%
1835   \caption@setformat{hang}%
1836   \caption@setjustification{justified}}
1837 \DeclareCaptionOptionNoValue*{center}{%
1838   \caption@setformat{plain}%
1839   \caption@setjustification{centering}}
1840 \DeclareCaptionOptionNoValue*{anne}{%
1841   \caption@setformat{plain}%
1842   \caption@setjustification{centerlast}}
1843 \DeclareCaptionOptionNoValue*{centerlast}{%
1844   \caption@setformat{plain}%
1845   \caption@setjustification{centerlast}}

1846 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1847 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1848 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1849 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1850 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1851 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1852 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1853 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1854 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1855 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1856 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1857 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1858 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1859 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1860 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1861 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}
1862 \caption@setbool{ruled}{0}
1863 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}
```

2.7.4 caption2 v2.x compatibility options

```
1864 \DeclareCaptionOptionNoValue*{flushleft}{%
1865   \caption@setformat{plain}%
1866   \caption@setjustification{raggedright}}
1867 \DeclareCaptionOptionNoValue*{flushright}{%
1868   \caption@setformat{plain}%
1869   \caption@setjustification{raggedleft}}

1870 \DeclareCaptionOptionNoValue*{online}{\caption@setbool{slc}{1}}
1871 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
1872   \caption@WarningNoLine{Obsolete option 'ignoreLTcapwidth' ignored}}
```

2.7.5 Obsolete caption v3.0 options

```
1873 \DeclareCaptionOption*{caption}{%
1874   \caption@setbool{temp}{#1}%
1875   \caption@ifbool{temp}{}{%
1876     \caption@Error{%
1877       The package option 'caption=#1' is obsolete.\MessageBreak
1878       Please pass this option to the subfig package instead\MessageBreak
1879       and do *not* load the caption package anymore}}}
```

2.7.6 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.

```
1880 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1881 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1882 \@onlypreamble@key{caption}{FPlist}
1883 \@onlypreamble@key{caption}{FPref}

1884 \newcommand*\caption@setFPoption[2]{%
1885   \edef\@tempa{\@car#2\@nil}%
1886   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1887 \@onlypreamble\caption@setFPoption

1888 \captionsetup{FPlist=caption,FPref=figure}
```

2.7.7 hyperref package support options

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

```
1889 \DeclareCaptionOption{hypcap}[1]{\caption@setbool{hypcap}{#1}}
1890 \DeclareCaptionOption{hypcapSPACE}{\def\caption@hypcapSPACE{#1}}
1891 \captionsetup{hypcap=1,hypcapSPACE=.5\baselineskip}
```

2.8 Processing of options

```
1892 \caption@ProcessOptions*{caption}
```

2.9 \captionof and \captionlistentry

```
1893 \caption@AtBeginDocument{%
1894   \DeclareCaptionOption{type}{\caption@settype{#1}}%
1895   \DeclareCaptionOption{type*}{\caption@settype*{#1}}%
1896   \DeclareCaptionOption{subtype}[sub\@capttype]{\caption@setsubtype{#1}}%
1897   \DeclareCaptionOption{subtype*}[sub\@capttype]{\caption@setsubtype*{#1}}%
1898 }
```

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).

```

1899 \newcommand*\caption@settype{%
1900   \caption@@settype{}}
1901 \newcommand*\caption@setsubtype{%
1902   \caption@iftype
1903     {\caption@settype{sub}}%
1904     {\caption@Error{Option 'subtype=' outside float}}}%
1905 \newcommand*\caption@@settype[1]{%
1906   \caption@teststar{\caption@@settype{#1}}\@firstoftwo\@secondoftwo}
1907 \newcommand*\caption@@settype[3]{%
1908   % #1 = "" or "sub"
1909   % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1910   % #3 = <type>, e.g. "figure" or "table"
1911   \@ifundefined{c@#3}%
1912     {\caption@Error{No float type '#3' defined}}%
1913     {\caption@Debug{#1type=#3}}%
1914     \caption@checkgrouplevel{#1}{%
1915       \captionsetup{#1type#2*\@empty=...}#2{ or
1916         \@backslashchar#1captionof}{}}%
1917     \edef\@tempa{#3}%
1918     \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1919       \ifcaptionsetup@star\else\@nameuse{caption@#1type@warning}\fi
1920     \fi
1921     \expandafter\let\csname @#1capttype\endcsname\@tempa
1922     \@nameuse{caption@#1typehook}%
1923     \caption@setoptions{#3}%
1924     \ifx\caption@opt\relax
1925       \@nameundef{caption@#1type@warning}%
1926     \else
1927       \@namedef{caption@#1type@warning}{\caption@Warning{%
1928         The #1caption type was already set to
1929         '\csname @#1capttype\endcsname'\MessageBreak}}%
1930     \fi
1931     \let\caption@ifrefstepcounter\@secondoftwo
1932     #2{}{%
1933       \let\@currentlabel\caption@undefinedlabel
1934     % \let\@currentHlabel\@undefined
1935       \ifx\caption@ORI@label\@undefined
1936         \let\caption@ORI@label\label
1937       \let\label\caption@xlabel
1938     \fi
1939     \caption@start}}}
```

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

```
1940 \newcommand*\caption@typehook{}
```

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

```

1941 \newcommand*\caption@iftype{%
1942   \@ifundefined{@capttype}{\let\@capttype\@undefined\@secondoftwo}\@firstoftwo}

\caption@checkgrouplevel Checks if \captionsetup{type=...} or \caption is done inside a group or not
– in the latter case a warning message will be issued. (needs  $\epsilon$ -TeX)
1943 \begingroup\expandafter\expandafter\expandafter\endgroup
1944 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1945   \caption@Debug{TeX engine: TeX}
1946   \let\caption@checkgrouplevel\@gobbletwo
1947 \else
1948   \caption@Debug{TeX engine: e-TeX}
1949   \newcommand*\caption@checkgrouplevel[2]{%
1950     \@ifundefined{#1caption@grouplevel}{%
1951       \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}{}%
1952       \ifnum\currentgrouplevel>\caption@grouplevel\relax
1953         \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1954           \the\currentgrouplevel}%
1955       \else
1956         \caption@Warning{\string#2MessageBreak outside box or environment}%
1957       \fi
1958     }{} }
1959 \fi

\caption@undefinedlabel This label will be used for \currentlabel inside (floating) environments as default.
(see above)
1960 \newcommand*\caption@undefinedlabel{%
1961   \protect\caption@xref{\caption@labelname}{\on@line}}
1962 \newcommand*\caption@xref[2]{%
1963   \caption@WarningNoLine{\noexpand\label before \string\caption#2}%
1964   \@setref\relax\@undefined{#1}}
1965 \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.
1966 \newcommand*\caption@xlabel[1]{%
1967   \caption@@xlabel
1968   \def\caption@labelname{#1}%
1969   \caption@ORI@label{#1}}
1970 \newcommand*\caption@@xlabel{%
1971   \global\let\caption@@xlabel\@empty
1972   \@bsphack
1973     \protected@write\@auxout{}%
1974       {\string\providecommand*\string\caption@xref[2]{%
1975         \string\@setref\string\relax\string\@undefined{\string##1}}}%
1976   \@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)
1977 \caption@AtBeginDocument{%
1978   \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}

```

```

1979 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

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

1980 \newcommand*\captionlistentry{%
1981   \caption@teststar\@captionlistentry\@firstoftwo\@secondoftwo}

1982 \newcommand*\@captionlistentry[1]{%
1983   \@testopt{\caption@listentry{#1}}{\@capttype}

1984 \def\caption@listentry#1[#2]#3{%
1985   \@bsphack
1986     #1{\def\@currentlabelname{#3}}%
1987     {\caption@refstepcounter{#2}}%
1988     \caption@makecurrent{#2}{#3}}%
1989   \caption@addcontentsline{#2}{#3}%
1990   \@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.

```

1991 \def\ContinuedFloat{%
1992   \@ifnextchar{\@Continued@Float\@ContinuedFloat}

1993 \def\@Continued@Float[#1]{\addtocounter{#1}\m@ne}

1994 \def\@ContinuedFloat{%
1995   \caption@iftype
1996     {\addtocounter\@capttype\m@ne
1997     \caption@ContinuedFloat\@capttype}%
1998     {\caption@Error{\noexpand\ContinuedFloat outside float}}}

1999 \def\caption@ContinuedFloat#1{%
2000   \@ifstar{\caption@Continued@Float@{#1}}{\caption@Continued@Float{#1}}}

```

```

2001 \def\caption@Continued@Float@{%
2002   \addtocounter\@capttype\@ne
2003   \@stpelt{ContinuedFloat}\stepcounter{ContinuedFloat}%
2004   \def\caption@resetContinuedFloat##1{\xdef\caption@CFtype{##1}}%
2005   \caption@@ContinuedFloat}

2006 \def\caption@Continued@Float#1{%
2007   \edef\@tempa{#1}%
2008   \ifx\@tempa\caption@CFtype
2009     \stepcounter{ContinuedFloat}%
2010     \let\caption@resetContinuedFloat\@gobble
2011     \caption@@ContinuedFloat{#1}%
2012     \sf@ContinuedFloat{#1}%
2013   \else
2014     \caption@Error{Continued `#1' after `\caption@CFtype'}%
2015   \fi}

2016 \def\caption@@ContinuedFloat#1{%
2017   \expandafter\l@addto@macro\csname the#1\endcsname\theContinuedFloat
2018   \@ifundefined{theH#1}{}{%
2019     \expandafter\l@addto@macro\csname theH#1\endcsname{%
2020       \@alph@c@ContinuedFloat}}%
2021   \caption@setoptions{ContinuedFloat}%
2022   \caption@setoptions{continued#1}}

2023 \providecommand*\sf@ContinuedFloat[1]{}

2024 \newcommand*\caption@CFtype{??}

```

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

```

2025 \newcounter{ContinuedFloat}
2026 \let\theContinuedFloat\@empty

```

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

```

2027 \newcommand*\caption@resetContinuedFloat[1]{%
2028   \@stpelt{ContinuedFloat}\xdef\caption@CFtype{#1}}

```

2.11 Internal helpers

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

```

2029 \newcommand*\caption@refstepcounter[1]{%
2030   \caption@resetContinuedFloat{#1}%
2031   \caption@@refstepcounter{#1}%
2032   \let\caption@ifrefstepcounter\@firstoftwo}

2033 \newcommand*\caption@@refstepcounter{\refstepcounter}
2034 \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[]{}).`

```

2035 \@ifundefined{kernel@ifnextchar}%

```



```

2036 {\newcommand\caption@dblarg[1]{\@ifnextchar[{\#1}{\caption@xdblarg{\#1}}}%
2037 {\newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{\#1}}}%
2038 \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`)

```

2039 \newcommand*\caption@begin[1]{%
2040   \begingroup
2041     \caption@setfnum{\#1}%
2042     \caption@fixposition
2043     \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>`.

```

2044 \newcommand\caption@beginex[3]{%
2045   \caption@begin{\#1}%
2046   \caption@addcontentsline{\#1}{\#2}%
2047   \caption@ifempty{\#3}{}%

```

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

```

2048 \newcommand*\caption@end{%
2049   \endgroup
2050   \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.

```

2051 \newcommand*\caption@setfnum[1]{%
2052   \@ifundefined{fnum@\#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
2053   \@namedef{fnum@\#1}{\caption@fnum{\#1}}%
2054   \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\-\@dischph
  \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`.

```

2055 \newcommand*\caption@boxrestore{%
2056   \let\if@nobreak\iffalse
2057   \let\if@noskipsec\iffalse
2058   \let\par\@par
2059 % \let\-\@dischyph
2060 % \let'\@acci\let'\@accii\let\=\@acciii
2061   \parindent\z@ \parskip\z@skip
2062   \everypar{}%
2063 % \linewidth\hsize
2064 % \@totalleftmargin\z@
2065   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
2066   \parfillskip\@flushglue \lineskip\normallineskip
2067   \baselineskip\normalbaselineskip
2068   \sloppy
2069   \let\\\@normalcr
2070 }

```

`\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).

```

2071 \newcommand*\caption@normalsize{%
2072   \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.

```
2073 \let\caption@setfloatcapt\@firstofone
```

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

```

\caption@makeanchor 2074 \newcommand*\caption@makecurrent[2]{%
\caption@start      2075 \let\caption@makeanchor\@firstofone
\caption@@start     2076 \let\caption@start\relax
\caption@freezeHref 2077 \let\caption@@start\relax
\caption@defrostHref 2078 \let\caption@freezeHref\relax
2079 \let\caption@defrostHref\relax

```

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

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

of the `captcont` package, which brings its own definition of `\caption*`, was quite common these days.

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

```

2080 \caption@AtBeginDocument{%
2081   \caption@setbool{incompatible}{0}%
2082   \caption@CheckCommand\caption{%
2083     % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2084     \def\caption{%
2085       \ifx\@capttype\@undefined
2086         \@latex@error{\noexpand\caption outside float}\@ehd
2087         \expandafter\@gobble
2088       \else
2089         \refstepcounter\@capttype
2090         \expandafter\@firstofone
2091       \fi
2092       {\@dblarg{\@caption\@capttype}}%
2093     }%
2094   \caption@CheckCommand\caption{%
2095     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
2096     \def\caption{
2097       \ifx\@capttype\@undefined
2098         \@latex@error{\noexpand\caption outside figure or table}\@ehd
2099         \expandafter\@gobble
2100       \else
2101         \refstepcounter\@capttype
2102         \expandafter\@firstofone
2103       \fi
2104       {\@dblarg{\@caption\@capttype}}%
2105     }%
2106   \caption@CheckCommand\caption{%
2107     % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2108     \renewcommand\caption{%
2109       \ifx\@capttype\@undefined
2110         \@latex@error{\noexpand\caption outside float}\@ehd
2111         \expandafter\@gobble
2112       \else
2113         \refstepcounter\@capttype
2114         \let\@tempf\@caption
2115         \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
2116           \expandafter\expandafter\let
2117             \expandafter\@tempf\csname @float@c@\@capttype\endcsname
2118         \fi
2119       \fi
2120       \@dblarg{\@tempf\@capttype}}%
2121   \caption@CheckCommand\caption{%
2122     % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2123     % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2124     % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2125     \def\caption{%

```

```

2126 \ifx\@capttype\@undefined
2127 \latexerror{\noexpand\caption outside float}\@ehd
2128 \expandafter\@gobble
2129 \else
2130 \H@refstepcounter\@capttype
2131 \ifundefined{fst@\@capttype}{%
2132 \let\Hy@tempa\@caption
2133 }{%
2134 \let\Hy@tempa\Hy@float@caption
2135 }%
2136 \expandafter\@firstofone
2137 \fi
2138 {\@dblarg{\Hy@tempa\@capttype}}%
2139 }%

2140 \caption@CheckCommand\caption{%
2141 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2142 \def\caption{%
2143 \ifx\@capttype\@undefined
2144 \latexerror{\noexpand\caption outside float}\@ehd
2145 \expandafter\@gobble
2146 \else
2147 \H@refstepcounter\@capttype
2148 \let\Hy@tempa\@caption
2149 \ifundefined{float@caption}{%
2150 }{%
2151 \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
2152 \let\Hy@tempa\Hy@float@caption
2153 \fi
2154 }%
2155 \expandafter\@firstofone
2156 \fi
2157 {\@dblarg{\Hy@tempa\@capttype}}%
2158 }%

2159 \caption@IfCheckCommand{}{%
2160 \caption@Info{%
2161 Incompatible package detected (regarding \string\caption).\MessageBreak
2162 \string\caption\space=\space meaning\caption}%
2163 \caption@setbool{incompatible}{1}}%

2164 \caption@CheckCommand\@caption{%
2165 % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
2166 \long\def\@caption#1[#2]#3{%
2167 \par
2168 \addcontentsline{\csname ext@#1\endcsname}{#1}%
2169 {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
2170 \begingroup
2171 \@parboxrestore
2172 \if@minipage
2173 \setminipage
2174 \fi
2175 \normalsize
2176 \makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2177 \endgroup}%
2178 \caption@CheckCommand\@caption{%

```

```

2179 % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
2180 \long\def\@caption#1[#2]#3{% second argument ignored
2181     \par\nobreak
2182     \begingroup
2183     \@parboxrestore
2184     \if@minipage
2185         \@setminipage
2186     \fi
2187     \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
2188     \endgroup}}%

2189 % \caption@CheckCommand\float@caption{%
2190 % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
2191 % \long\def\float@caption#1[#2]#3{%
2192 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%
2193 %     {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
2194 %     \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
2195 %     \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
2196 %     \@ifnextchar[{\float@ccon}{\egroup}}%
2197 %     \long\def\float@ccon[#1]{#1\par\egroup}}%

2198 \caption@CheckCommand\@caption{%
2199 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
2200 \long\def\@caption#1[#2]#3{%
2201     \hyper@makecurrent{\@cuptype}%
2202     \def\@currentlabelname{#2}%
2203     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2204         \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2205     }%
2206     \begingroup
2207     \@parboxrestore
2208     \if@minipage
2209         \@setminipage
2210     \fi
2211     \normalsize
2212     \@makecaption{\csname fnum@#1\endcsname}{%
2213         \ignorespaces
2214         \ifHy@nesting
2215             \hyper@@anchor{\@currentHref}{#3}%
2216         \else
2217             \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2218         \fi
2219     }%
2220     \par
2221     \endgroup
2222 }}%

2223 \caption@CheckCommand\@caption{%
2224 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
2225 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
2226 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
2227 \long\def\@caption#1[#2]#3{%
2228     \expandafter\ifx\csname if@capstart\expandafter\endcsname
2229         \csname iftrue\endcsname
2230     \global\let\@currentHref\hc@currentHref
2231     \else

```

```

2232         \hyper@makecurrent{\@captype}%
2233     \fi
2234     \def\@currentlabelname{#2}%
2235     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
2236         \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
2237     }%
2238     \begingroup
2239         \@parboxrestore
2240         \if@minipage
2241             \@setminipage
2242         \fi
2243         \normalsize
2244         \expandafter\ifx\csname if@capstart\expandafter\endcsname
2245             \csname iftrue\endcsname
2246             \global\@capstartfalse
2247             \makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
2248         \else
2249             \makecaption{\csname fnum@#1\endcsname}{%
2250                 \ignorespaces
2251                 \ifHy@nesting
2252                     \hyper@@anchor{\@currentHref}{#3}%
2253                 \else
2254                     \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
2255                 \fi
2256             }%
2257         \fi
2258     \par
2259 \endgroup
2260 } }%

2261 \caption@CheckCommand\@caption{%
2262 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
2263 \long\def\@caption#1[#2]{%
2264     \def\@currentlabelname{#2}%
2265     \NR@@caption{#1} [{#2}]%
2266 } }%

2267 \caption@CheckCommand\@caption{%
2268 % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
2269 \long\def\@caption#1[#2]#3{%
2270     \ifundefined{if#1topcap}%
2271     {\subfig@oldcaption{#1} [{#2}] {#3}}%
2272     {\@nameuse{if#1topcap}%
2273         \@listsubcaptions{#1}%
2274         \subfig@oldcaption{#1} [{#2}] {#3}%
2275     \else
2276         \subfig@oldcaption{#1} [{#2}] {#3}%
2277         \@listsubcaptions{#1}%
2278     \fi}}}%

2279 \caption@CheckCommand\@caption{%
2280 % subfig.sty [2005/06/28 ver: 1.3 subfig package]
2281 \def\@caption{\caption}%
2282 % \long\def\caption@#1[#2]#3{%
2283 %     \ifundefined{caption@setfloattype}%
2284 %     \caption@settype

```

```

2285 % \caption@setfloattype
2286 % \@capttype
2287 % \sf@ifpositiontop{%
2288 % \@listsubcaptions{#1}%
2289 % \sf@old@caption{#1}[{#2}]{#3}%
2290 % }{%
2291 % \sf@old@caption{#1}[{#2}]{#3}%
2292 % \@listsubcaptions{#1}%
2293 % }}%
2294 %}%
2295 \caption@ifcheckcommand{}{%
2296 \caption@info{%
2297 Incompatible package detected (regarding \string\@caption).\MessageBreak
2298 \string\@caption\space=\space\meaning\@caption}%
2299 \caption@setbool{incompatible}{1}}%

```

The option compatibility= will override the compatibility mode.

```

2300 \@ifundefined{caption@ifcompatibility}%
2301 {\let\caption@ifcompatibility\caption@ifincompatible
2302 \let\@tempa\caption@WarningNoLine}%
2303 {\let\@tempa\@gobble}% suppress warning
2304 \caption@ifcompatibility{%
2305 \@tempa{%
2306 \noexpand\caption will not be redefined since it's already\MessageBreak
2307 redefined by a document class or package which is\MessageBreak
2308 unknown to the caption package}%

```

\ContinuedFloat is not supported in compatibility mode.

```

2309 \renewcommand*\caption@ContinuedFloat[1]{%
2310 \caption@error{Not available in compatibility mode}}%

```

\caption@start is not supported in compatibility mode.

```

2311 \caption@atbegindocument*{%
2312 \let\caption@start\relax
2313 \@ifundefined{caption@ORI@capstart}{}{%
2314 \caption@debug{%
2315 Restore hypcap definition of \string\capstart\@gobble}%
2316 \let\capstart\caption@ORI@capstart}%
2317 \@ifundefined{caption@ORI@float@makebox}{}{%
2318 \caption@debug{%
2319 Restore hyperref redefinition of \string\float@makebox\@gobble}%
2320 \let\float@makebox\caption@ORI@float@makebox}%
2321 }%

```

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

```

2322 \newcommand*\caption@star[2]{#1#2}%
2323 }{%
2324 \caption@ifincompatible{%
2325 \caption@WarningNoLine{%
2326 Forced redefinition of \noexpand\caption since the\MessageBreak
2327 unsupported(!) package option 'compatibility=false'\MessageBreak
2328 was given}%
2329 }{}%

```

`\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`.

```

2330 \def\caption{%
2331 \caption@iftyp
2332 {\caption@checkgrouplevel\@empty\caption
2333 \caption@star
2334 {\caption@refstepcounter\@capttype}%
2335 {\caption@dblarg{\@caption\@capttype}}}%
2336 {\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.

```

2337 \newcommand*\caption@star[2]{%
2338 \ifstar{\global\caption@startrue#2[]}{#1#2}}%

```

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

```

2339 \long\def\@caption#1[#2]#3{%
2340 \ifcaption@star \else
2341 \caption@prepareanchor{#1}{#2}%
2342 \fi
2343 \par
2344 \caption@beginex{#1}{#2}{#3}%
2345 \caption@setfloatcapt{%
2346 \caption@boxrestore
2347 \if@minipage
2348 \@setminipage
2349 \fi
2350 \caption@normalsize
2351 \ifcaption@star
2352 \let\caption@makeanchor\@firstofone
2353 \fi
2354 \@makecaption{\csname fnum@#1\endcsname}%
2355 {\ignorespaces\caption@makeanchor{#3}}\par
2356 \caption@if@minipage\@minipagetrue\@minipagefalse}%
2357 \caption@end}%

```

`\caption@prepareanchor`

```

2358 \newcommand*\caption@prepareanchor[2]{%
2359 \caption@makecurrent{#1}{#2}%
2360 \caption@ifhycap\caption@@start{}%
2361 }%
2362 \caption@AtBeginDocument*{%
2363 \let\caption@ORI@capstart\@undefined
2364 \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.
2365 \let\caption@ORI@xfloat\@xfloat
2366 \def\@xfloat#1[#2]{%
2367     \caption@ORI@xfloat{#1}[#2]%
2368     \caption@settype{#1}}%
2369 }

\@makecaption \@makecaption{<label>}{<text>}
We do basically the same as the original code (from the standard LATEX document classes),
but take care of the position= setting and use \caption@@make from the caption
kernel to finally typeset the caption.
2370 \long\def\@makecaption#1#2{%
2371     \caption@iftop
2372     {\vskip\belowcaptionskip}%
2373     {\caption@rule\vskip\abovecaptionskip}%
2374     \caption@@make{#1}{#2}%
2375     \caption@iftop
2376     {\vskip\abovecaptionskip\caption@rule}%
2377     {\vskip\belowcaptionskip}}

```

2.13 Support for sub-captions

```

\caption@DeclareSubType \caption@DeclareSub initializes the usage of \caption in sub-floats.
2378 \def\caption@DeclareSubType sub#1\@nil{%
2379     \caption@Debug{Initializing subtype for `#1'\@gobble}%
2380     \@namedef{caption@c@#1}{0}%
2381     \@namedef{caption@beginsub#1}{\caption@beginsubfloat{#1}}
2382     \@onlypreamble\caption@DeclareSubType

Initialize the sub-captions defined with \DeclareCaptionSubType...
2383 \caption@For*{subtypelist}{\caption@DeclareSubType sub#1\@nil}

Initialize the sub-captions defined with \newsubfloat[18]...
2384 \caption@AtBeginDocument*{%
2385     \ifundefined{sf@counterlist}{\}%
2386     \@for\s@temp:=\sf@counterlist\do{%
2387         \expandafter\caption@DeclareSubType\s@temp\@nil}}}

\caption@subtypehook Hook, will be used inside \caption@setsubtype.
2388 \newcommand*\caption@subtypehook{%
2389     \ifx\caption\caption@subcaption \else
2390         \caption@ifrefstepcounter{\}%
2391         % no \caption or \subcaption in this (floating) environment yet
2392         \caption@Debug{Increment \@c@type\ counter =\the\value\@c@type}%
2393         \caption@l@stepcounter\@c@type
2394         \let\addcontentsline\caption@addsubcontentsline}%
2395     \ifnum\csname caption@c@\@c@type\endcsname=\value\@c@type \else
2396         \caption@Debug{Reset sub\@c@type\ counter}%
2397         \expandafter\xdef\csname caption@c@\@c@type\endcsname{%
2398             \the\value\@c@type}%
2399         \@stpelt\@subc@type
2400     \fi

```

```

2401 \c@ContinuedFloat=0\relax
2402 \let\caption@resetContinuedFloat\@gobble
2403 \let\caption@addcontentsline\caption@kernel@addcontentsline
2404 \let\caption@setfloatcapt\@firstofone
2405 \caption@clearmargin
2406 \caption@iflist{}\let\caption@setlist\@gobble}%
2407 \caption@setoptions{sub}%
2408 \caption@setoptions{subfloat}% for subfig-package compatibility
2409 \let\caption\caption@subcaption
2410 \fi}%

```

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

```

2411 \newcommand*\caption@subcaption{%
2412 \caption@iftype
2413 {\caption@checkgrouplevel{sub}\subcaption
2414 \caption@star
2415 {\caption@refstepcounter\@subcaption}%
2416 {\caption@dblarg{\caption\@subcaption}}}%
2417 {\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.

```

2418 \let\caption@kernel@addcontentsline\caption@addcontentsline
2419 \renewcommand*\caption@addcontentsline[2]{%
2420 \sf@ifpositiontop{\@listsubcaptions{#1}}}%
2421 \caption@kernel@addcontentsline{#1}{#2}%
2422 \sf@ifpositiontop{}\@listsubcaptions{#1}}}%
2423 \caption@addsubcontentslines{#1}}
2424 \newcommand*\caption@addsubcontentslines[1]{%
2425 \begingroup
2426 \caption@subcontentslines
2427 \endgroup
2428 \caption@clearsubcontentslines}%
2429 \caption@AtBeginDocument*{%
2430 \@ifundefined{sf@ifpositiontop}{\let\sf@ifpositiontop\@gobbletwo}}}%
2431 \caption@clearsubcontentslines
2432 \g@addto@macro\caption@typehook{\caption@checksubcontentslines}%
2433 \AtEndDocument{\caption@checksubcontentslines}}%

```

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

```

2434 \newcommand*\caption@addsubcontentsline[3]{%
2435 \begingroup
2436 \let\label\@gobble \let\index\@gobble \let\glossary\@gobble
2437 \protected@edef\@tempa{\endgroup
2438 \noexpand\g@addto@macro\noexpand\caption@subcontentslines{%
2439 \noexpand\@namedef{the#2}{\csname the#2\endcsname}%
2440 \ifx\@currentHref\@undefined \else
2441 \noexpand\def\noexpand\@currentHref{\@currentHref}%
2442 \fi
2443 \protect\addcontentsline{#1}{#2}{#3}}}%
2444 \@tempa}

```

```

ion@checksubcontentslines Checks if the list of pending sub-captions is empty, if not, a warning will be issued.
2445 \newcommand*\caption@checksubcontentslines{%
2446   \ifx\caption@subcontentslines\@empty \else
2447     \caption@Error{%
2448       Something's wrong--perhaps a missing \protect\caption\MessageBreak
2449       in the last figure or table}%
2450     \caption@clearsubcontentslines
2451   \fi}

```

```

ion@clearsubcontentslines Clear pending sub-caption list entries.
2452 \newcommand*\caption@clearsubcontentslines{%
2453   \global\let\caption@subcontentslines\@empty}

```

2.14 Document class & Babel package support

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

```

2454 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}

```

2.14.2 The beamer class

```

2455 \@ifclassloaded{beamer}{%
2456   \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.

```

2457   \captionsetup{list=false}
2458   \DeclareCaptionOption{list}[1]{}
2459   \DeclareCaptionOption{listof}[1]{}

```

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

```

2460   \expandafter\let\expandafter\caption@ORI@figure
2461     \csname\string\figure\endcsname
2462   \@namedef{\string\figure}[#1]{%
2463     \caption@ORI@figure[#1]%
2464     \caption@settype{figure}}
2465   \expandafter\let\expandafter\caption@ORI@table
2466     \csname\string\table\endcsname
2467   \@namedef{\string\table}[#1]{%
2468     \caption@ORI@table[#1]%
2469     \caption@settype{table}}
2470 }{}

```

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.

```

2471 \@ifundefined{scr@caption}{}{%
2472   \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)

```
2473 \@nameuse{caption@frenchb}\@nameundef{caption@frenchb}
```

2.14.5 The frenchle/pro package

```
2474 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}}{ }{ }%
2475 \caption@Info{frenchle/pro package is loaded}%

2476 \let\captionfont@ORI\captionfont
2477 \let\captionlabelfont@ORI\captionlabelfont
2478 \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.

```
2479 \@ifundefined{GOfrench}%
2480   {\let\@tempa\@firstofone}%
2481   {\def\@tempa{\g@addto@macro\GOfrench}}%
2482 \@tempa{%

2483   \let\captionfont\captionfont@ORI
2484   \let\captionfont@ORI\@undefined
2485   \let\captionlabelfont\captionlabelfont@ORI
2486   \let\captionlabelfont@ORI\@undefined
2487   \let\@makecaption\@makecaption@ORI
2488   \let\@makecaption@ORI\@undefined
```

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

```
2489 \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.

```
2490 \let\caption@tcORI\@tablescaption
2491 \def\@tablescaption{\caption@star\relax\caption@tcORI}%
```

\f@ffrench \f@ffrench and \f@tfrench reflect \fnum@figure and \fnum@table when 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.

```
2492 \let\@eatDP\@undefined
2493 \let\@tempa\@empty
2494 \ifx\f@ffrench\fnum@figure
2495   \l@addto@macro\@tempa{\let\fnum@figure\f@ffrench}%
2496 \fi
2497 \ifx\f@tfrench\fnum@table
2498   \l@addto@macro\@tempa{\let\fnum@table\f@tfrench}%
2499 \fi
2500 \def\f@ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
2501 \def\f@tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
2502 \@tempa
```

```

2503 }%
2504 } }

```

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.

2505 \newcommand\caption@ifpackageloaded[1]{%
2506   \@testopt{\caption@ifpackageloaded{#1}}{}}
2507 \@onlypreamble\caption@ifpackageloaded

2508 \long\def\caption@@ifpackageloaded#1[#2]#3#4{%
2509   \@ifpackageloaded{#1}\@firstofone{%
2510     \caption@Debug{#1 package is not loaded (yet)\@gobble}%
2511     \caption@AtBeginDocument{%
2512       \caption@@ifpackageloaded{#1}{#2}{#3}{#4}}
2513 \@onlypreamble\caption@@ifpackageloaded

2514 \newcommand\caption@ifpackageloaded[1]{%
2515   \@testopt{\caption@ifpackageloaded{#1}}{}}
2516 \@onlypreamble\caption@ifpackageloaded

2517 \long\def\caption@@ifpackageloaded#1[#2]{%
2518   \@ifpackageloaded{#1}{%
2519     \caption@Info{#1 package is loaded}%
2520     \@ifpackagelater{#1}{#2}\@firstoftwo{%
2521       \caption@Error{%
2522         For a successful cooperation we need at least version\MessageBreak
2523         `#2' of package #1,\MessageBreak
2524         but only version\MessageBreak
2525         `\'csname ver@#1.\@pkgextension\endcsname'\MessageBreak
2526         is available}%
2527       \@secondoftwo}%
2528     }{\@secondoftwo}}
2529 \@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.

2530 \newcommand*\caption@clearmargin{%
2531   \setcaptionmargin\z@
2532   \let\caption@minmargin\@undefined}

2533 \caption@setbool{needfreeze}{0}
2534 \caption@AtBeginDocument*{%
2535   \caption@ifneedfreeze{%

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

2536 \newcommand*\caption@freeze{%
2537   \caption@teststar\caption@@freeze\@gobble\@firstofone}%

2538 \newcommand*\caption@@freeze[1]{%
2539   \global\let\caption@SCcontinued\relax

```

```

2540 \global\let\caption@SCsetup\@undefined
2541 \global\let\caption@SClentry\@undefined
2542 \global\let\caption@SCtext\@undefined
2543 \global\let\caption@SClabel\@undefined

2544 \let\caption@ORI@ContinuedFloat\ContinuedFloat
2545 \def\ContinuedFloat{%
2546   \caption@withoptargs\caption@SC@ContinuedFloat}%
2547 \def\caption@SC@ContinuedFloat##1{%
2548   \let\caption@ORI@setcounter\setcounter
2549   \let\caption@ORI@addtocounter\addtocounter
2550   \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
2551   \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
2552   \caption@ORI@ContinuedFloat##1%
2553   \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
2554   \let\setcounter\caption@ORI@setcounter
2555   \let\addtocounter\caption@ORI@addtocounter}%
2556 \let\caption@ORI@setup\captionsetup
2557 \def\captionsetup{%
2558   \caption@withoptargs\caption@SC@setup}%
2559 \def\caption@SC@setup##1##2{%
2560   \caption@g@addto@list\caption@SCsetup{##2}%
2561   \caption@ORI@setup##1{##2}}%
2562 \let\caption@ORI\caption
2563 \def\caption{%
2564   \def\caption{\caption@Error{%
2565     Only one \noexpand\caption can be placed in this environment}}%
2566   \let\captionsetup\caption@setup
2567   \let\caption@refstepcounter\caption@l@stepcounter
2568   \caption@ORI}%
2569 \long\def\@caption##1[##2]##3{%
2570   \@bsphack
2571   \gdef\caption@SClentry{##2}%
2572   \gdef\caption@SCtext{##3}%
2573   \@esphack}%
2574 #1{% is \@gobble in star form, and \@firstofone otherwise
2575   \def\label##1{\@bsphack\gdef\caption@SClabel{##1}\@esphack}}%
2576 }%

```

\caption@defrost \caption@defrost

```

2577 \newcommand*\caption@defrost{%
2578   \ifx\caption@ORI@ContinuedFloat\@undefined
2579     \caption@defrost@setup
2580     \ifx\caption@SCtext\@undefined \else
2581       \expandafter\expandafter\expandafter\caption
2582       \expandafter\expandafter\expandafter[%
2583       \expandafter\expandafter\expandafter{%
2584       \expandafter\caption@SClentry\expandafter}\expandafter]%
2585       \expandafter{\caption@SCtext}%
2586     \fi
2587     \ifx\caption@SClabel\@undefined \else
2588       \expandafter\label\expandafter{\caption@SClabel}%
2589     \fi
2590   \else

```

```

2591 \caption@Error{Internal Error:\MessageBreak
2592 \noexpand\caption@defrost in same group as \string\caption@freeze}%
2593 \fi}%
2594 \newcommand*\caption@defrost@setup{%
2595 \caption@SCcontinued
2596 \ifx\caption@SCsetup\@undefined \else
2597 \expandafter\captionsetup\expandafter{\caption@SCsetup}%
2598 \fi}%
2599 }{}%
2600 \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.)

```

2601 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
2602 \@ifpackageloaded{floatrow}{%
2603 \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]}{}%
2604 }{}%

```

`\@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>]`.

```

2605 \let\caption@ORI@float@setevery\@float@setevery
2606 \def\@float@setevery#1{%
2607 \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.

```

2608 \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.

```

2609 \renewcommand\caption@setfloatcapt{%
2610 \let\@makecaption\caption@make
2611 \global\setbox\@floatcapt\vbox}%

```

To allow different caption styles for different float styles we also determine the current float style (e.g. ‘ruled’) and select a caption style (and additional settings) with the same name, if defined.

```

2612     \float@getstyle\float@style{#1}%
2613     \caption@setstyle*\float@style
2614     \caption@setoptions\float@style
2615     }{}%
2616     \caption@freezeHref % will be defrosted in \float@makebox
2617     \caption@ORI@float@setevery{#1}}%

```

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

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

But the float package doesn’t follow this common naming convention: For floats defined with `\newfloat` it uses `\fname@⟨type⟩` instead, which breaks with our code (and with `\autoref` and some other things as well). So we have to map the float package name to the common one here.

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

```

2618 \g@addto@macro\caption@typehook{%
2619     \expandafter\ifx\csname #1name\endcsname\relax
2620     \expandafter\let\csname #1name\endcsname\expandafter\endcsname
2621     \csname fname@#1\endcsname
2622     \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.

```

2623 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2624 \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).

```

2625 \providecommand*\float@ifstyle[1]{%
2626     \expandafter\ifx\csname fst@#1\endcsname\relax
2627     \expandafter\@secondoftwo
2628     \else
2629     \expandafter\@firstoftwo
2630     \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.

```
2631 \providecommand*\float@getstyle[2]{%
2632   \edef#1{%
2633     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2634     \expandafter\expandafter\expandafter\noexpand
2635     \csname fst@#2\endcsname}%
2636   \edef#1{#1}%
2637   \caption@Debug{floatstyle{#2} = '#1'}}%
```

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

```
2638 \providecommand*\float@setstyle[2]{%
2639   \expandafter\edef\csname fst@#1\endcsname{%
2640     \expandafter\noexpand\csname fs@#2\endcsname}}%
```

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

```
2641 \providecommand*\float@dostyle[1]{%
2642   \@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.)

```
2643 \providecommand*\float@ifcaption[1]{%
2644   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2645   \expandafter\@firstoftwo
2646   \else
2647   \expandafter\@secondoftwo
2648   \fi}%
```

```
2649 } }{%
2650 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2651 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2652 % \clearcaptionsetup{boxed}% used by the floatrow package?
2653 }
```

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

```
2654 \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’.

```
2655 \caption@ifbool{ruled}{%
2656   \captionsetup[ruled]{margin=0pt,minmargin=0,slc=0}%
```

```

2657 }{%
2658   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2659 }
2660 \caption@undefbool{ruled}

```

2.15.2 The floatflt package

```

2661 \caption@IfPackageLoaded{floatflt}[1996/02/27 v1.3]{%
\floatingfigure We patch \floatingfigure so \caption@floatflt will be used.
2662   \let\caption@ORI@floatingfigure\floatingfigure
2663   \def\floatingfigure{%
2664     \caption@floatflt{figure}%
2665     \caption@ORI@floatingfigure}%

\floatingtable Same with \floatingtable...
2666   \let\caption@ORI@floatingtable\floatingtable
2667   \def\floatingtable{%
2668     \caption@floatflt{table}%
2669   %   \caption@setautoposition b%
2670     \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.

2671   \newcommand*\caption@floatflt[1]{%
2672     \caption@settype{#1}%
2673     \caption@clearmargin
2674     \caption@setoptions{floating#1}%
2675     \let\caption@boxrestore\@parboxrestore}%

2676 }{}

```

2.15.3 The fltpage package

```

2677 \caption@IfPackageLoaded{fltpage}[1998/10/29 v.0.3]{%
2678   \caption@setbool{needfreeze}{1}%

\FP@helpNote Original code:

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

2679   \renewcommand\FP@helpNote[2]{%
2680     \begingroup % save \caption@thepage
2681       \caption@pageref{#2}%
2682       \typeout{FP#1 is inserted on page \caption@thepage!}%
2683     \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][]{%
    \gdef\FP@optionalCaptionText{##1}\gdef\FP@CaptionText{##2}}%
  \begin{lrbox}{\FP@floatCorpusBOX}%
}%

2684 \renewcommand*\FP@floatBegin[1]{%
2685   \def\@captype{#1}%
2686   \let\FP@LabelText\@empty
2687   \begin{lrbox}{\FP@floatCorpusBOX}%
2688   \caption@ifFPrefcap
2689     {\caption@freeze\relax}%
2690     {\def\label##1{\@bsphack\gdef\FP@LabelText{##1}\@esphack}%
2691     \caption@freeze*}%

```

\FP@floatEnd **Original code:**

```

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

```

```

\end{\@capttype}}%
}%

2692 \renewcommand*\FP@floatEnd{%
2693   \end{lrbox}%
2694   \stepcounter{FP@\@capttype C}%
2695   \caption@label\FP@positionLabel
2696   \FP@helpNote\@capttype\FP@positionLabel
2697   \edef\FP@RestoreCounter{%
2698     \noexpand\setcounter{\@capttype}{\the\value\@capttype}%
2699     \noexpand\setcounter{ContinuedFloat}{\the\value{ContinuedFloat}}}%
2700   \FP@float
2701   {\FP@positionLabel}% location label test
2702   {\begin\@capttype[p!}%
2703     \usebox\FP@floatCorpusBOX
2704     \caption@defrost@setup
2705     \caption@ifFPlistcap
2706       {\caption@refstepcounter\@capttype
2707         \expandafter\caption@makecurrent\expandafter\@capttype
2708           \expandafter{\caption@SCLentry}}}%
2709       {\expandafter\captionlistentry\expandafter{\caption@SCLentry}}}%
2710   \caption@makeanchor\relax
2711   \ifx\FP@LabelText\@empty \else
2712     \expandafter\label\expandafter{\FP@LabelText}%
2713   \fi
2714   \end\@capttype}%
2715   {\FP@RestoreCounter
2716     \@ifundefined{theH\@capttype}{}{%
2717       \expandafter\l@addto@macro\csname theH\@capttype\endcsname{.FP}}}%
2718   {\begin\@capttype[b!}%
2719     \let\FP@savedSetfnumCommand\caption@setfnum
2720     \def\caption@setfnum##1{%
2721       \FP@savedSetfnumCommand{##1}%
2722       \ifx\FP@guide\@empty \else
2723         \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ {\FP@guide}}%
2724       \fi}%
2725     \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above captio
2726     \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below captio
2727     \caption@setoptions{FP\@capttype}%
2728     \FP@separatorCaption
2729     \caption@ifFPlistcap{}{\let\caption@addcontentsline\@gobbletwo}%
2730     \caption@defrost
2731   \end\@capttype}%
2732 }%

2733 \caption@For{typelist}{%
2734   \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2735 }%
2736 \let\caption@ifFPlistcap\@undefined
2737 \let\caption@ifFPrefcap\@undefined
2738 }

```

2.15.4 The hyperref package

```

2739 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2740   \ifundefined{hyper@makecurrent}{% hyperref has stopped early
2741     \caption@WarningNoLine{%
2742       Hyperref support is turned off\MessageBreak
2743       because hyperref has stopped early}%
2744   }{%
2745     \g@addto@macro\caption@prepareslc{\measuring@true}%

```

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

```

2746   \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

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

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

```

2747   \renewcommand*\caption@makecurrent[2]{%
2748     \caption@makecurrentHref{#1}%
2749     \caption@Debug{hyperref current=\@currentHref}%
2750     \def\@currentlabelname{#2}}%
2751   \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

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

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

```

2752   \renewcommand\caption@makeanchor[1]{%
2753     \caption@Debug{hyperref anchor: \@currentHref}%
2754     % If we cannot have nesting, the anchor is empty.
2755     \ifHy@nesting
2756       \hyper@@anchor{\@currentHref}{#1}%
2757     \else
2758       \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2759     \fi}%
2760   \g@addto@macro\caption@prepareslc{\let\caption@makeanchor\@firstofone}%

```

The hypcap option

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

```

2761   \newif\if@capstart

```

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

```

2762   \def\caption@start{\caption@ifhypcap{%

```

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

```

2763     \caption@makestart\@captype
2764     \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 hypcap package.)

```

2765     \global\@capstarttrue

```

```

2766 \let\hc@currentHref\@currentHref
2767 \def\caption@makecurrentHref##1{%
2768   \global\@capstartfalse
2769   \global\let\@currentHref\hc@currentHref}%

```

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

```

2770 \let\caption@makeanchor\@firstofone
2771 }{}%

```

\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.

```

2772 \newcommand*\caption@makestart[1]{%
2773   \begingroup
2774   \Hy@hypertexnamesfalse
2775   \gdef\@currentHlabel{}%
2776   \hyper@makecurrent{#1.caption}%
2777   \endgroup
2778   \caption@Debug{hypcap start=\@currentHref}%

```

\caption@startanchor \caption@startanchor{<Href>} sets a hyperref anchor inside \caption@start. This code was taken from the `hypcap` 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 `hypcap` package is not always sufficient (for example with “Direct pdfmark support” and `breaklinks=true`), we use `\caption@anchor` instead of `\hyper@@anchor` here.

```

2779 \newcommand*\caption@startanchor[1]{%
2780   \ifvmode\begingroup
2781   \caption@Debug{hypcap anchor: #1 (vertical mode)}%
2782   \@tempdima\prevdepth
2783   \nointerlineskip
2784   \vspace*{-\caption@hypcapspace}%
2785   \caption@anchor{#1}%
2786   \vspace*{\caption@hypcapspace}%
2787   \prevdepth\@tempdima
2788   \endgroup\else

```

```

2789         \caption@Debug{hypcap anchor: #1 (horizontal mode)}%
2790         \caption@anchor{#1}%
2791     \fi}%

\caption@anchor \caption@anchor{<Href>} sets a hyperref anchor.
2792     \newcommand*\caption@anchor[1]{%
2793         \ifmeasuring@ \else
2794             \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2795         \fi}%

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

2796     \ifx\HyperRaiseLinkLength\@tempdima
2797         \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2798     \else
2799         \let\caption@raisedlink\Hy@raisedlink
2800     \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.
2801     \def\caption@@start{%
2802         \ifundefined{hc@currentHref}{%
2803             \caption@Warning{%
2804                 The option ‘hypcap=true’ will be ignored for this\MessageBreak
2805                 particular \string\caption}}}%

\caption@freezeHref Suppress \caption@start from generating a hyperref label and setting a hyper-
ref anchor. Instead if \@caption generates a hyperref label, it will be stored in
\caption@currentHref. Furthermore we need to redefine \caption@setfloatcapt
so no hyperref anchor will be placed in \@caption.
2806     \def\caption@freezeHref{%
2807         \let\caption@ORI@start\caption@start
2808         \def\caption@start{\let\caption@start\caption@ORI@start}%

2809 %     \let\caption@ORI@@start\caption@@start
2810 %     \l@addto@macro\caption@subtypehook{%
2811 %         \let\caption@@start\caption@ORI@@start}%

2812     \global\let\caption@currentHref\@undefined
2813     \def\caption@@start{\global\let\caption@currentHref\@currentHref}%

2814     \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2815     \renewcommand*\caption@setfloatcapt{%
2816         \ifx\caption@currentHref\@undefined \else
2817             \let\caption@makeanchor\@firstofone
2818         \fi
2819         \caption@ORI@setfloatcapt}%

\caption@defrostHref If there is a freed \@currentHref, we set the hyperref anchor here.
2820     \def\caption@defrostHref{%
2821         \ifx\caption@currentHref\@undefined \else
2822             \caption@startanchor\caption@currentHref
2823             \global\let\caption@currentHref\@undefined
2824         \fi}%

```

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

```

2825 \ifundefined{HyOrg@float@makebox}{}{}%
2826 \caption@Debug{%
2827   Redefining \noexpand\float@makebox (again)\@gobble}%
2828 \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
2829 \renewcommand\float@makebox[1]{%
2830   \HyOrg@float@makebox{#1\relax \caption@defrostHref}}%
2831 }%
2832 }{}{}

```

2.15.5 The hypcap package

```

2833 \caption@IfPackageLoaded{hypcap}{% v1.0
2834   \ifx\caption@start\relax \else % hyperref hasn't stopped early

```

If the `hypcap` package was loaded, we give up our own hyperlink placement algorithm and give the control over the placement to the `hypcap` package instead.

`\capstart` We do this simply by mapping `\capstart` to `\caption@start`, although our code does not behave exactly like the original one: The original `\capstart` has an effect on the next `\caption` only but our version affects *all* `\captions` in the same environment, at least unless a new `\capstart` will be placed.

```

2835 \let\caption@ORI@capstart\capstart % save for compatibility mode
2836 \let\capstart\caption@start
2837 \let\caption@start\relax
2838 \let\caption@@start\relax

```

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

```

2839 \caption@set@bool\caption@ifhypcap1%
2840 \renewcommand*\caption@hypcapspace{\hypcapspace}%
2841 \fi}{}

```

2.15.6 The listings package

```

2842 \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.

```

2843 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2844 \def\lst@MakeCaption#1{% #1 is ‘t’ or ‘b’
2845   \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`.)

```

2846 \caption@setposition{#1}%
2847 \caption@iftop{%

```



```

2848         \@tempdima\belowcaptionskip
2849         \belowcaptionskip\abovecaptionskip
2850         \abovecaptionskip\@tempdima\}{}%

```

Afterwards we set the local ‘lstlisting’ options.

```

2851         \caption@setoptions{lstlisting}%

```

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

```

2852         \caption@setautoposition{#1}%

```

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

```

2853         \caption@begin{lstlisting}%
2854         \caption@ORI@lst@MakeCaption{#1}%
2855         \caption@end
2856     \endgroup}%

```

`\lst@makecaption` Wrapper macros for typesetting the caption= resp. title= value.

```

\lst@maketitle 2857 \def\lst@makecaption{\caption@starfalse\@makecaption}%
                2858 \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.

```

2859 \providecommand*\ext@lstlisting{lol}%
2860 {}{}

```

2.15.7 The longtable package

`\LTcapttype` `\LTcapttype` is preset to table.

```

2861 \providecommand*\LTcapttype{table}

2862 \caption@IfPackageLoaded{longtable}[1995/05/24 v3.14]{%
2863     \RequirePackage{ltcaption}[2007/09/01]%
2864     \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`.

```

2865 \caption@AtBeginDocument{%
2866     \let\caption@ORI@LT@array\LT@array
2867     \renewcommand*\LT@array{%

```

`\captionsetup` for longtable:

```

2868         \global\let\caption@opt@@longtable\@undefined
2869         \def\captionsetup{%
2870             \noalign\bgroup
2871             \@ifstar\@captionsetup\@captionsetup}% gobble *
2872         \def\@captionsetup##1{\LT@captionsetup{##1}\egroup}%
2873         \def\LT@captionsetup##1{%
2874             \captionsetup@startrue\caption@setup@options[@longtable]{##1}%
2875             \global\let\caption@opt@@longtable\caption@opt@@longtable}%

```

```

\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2876 \def\@captionabovetrue{\LT@captionsetup{position=t}}%
2877 \def\@captionabovefalse{\LT@captionsetup{position=b}}%
\captionlistentry for longtable:
2878 \def\captionlistentry{%
2879 \noalign\bgroup
2880 \@ifstar{\egroup\LT@captionlistentry}% gobble *
2881 {\egroup\LT@captionlistentry}}%
2882 \def\LT@captionlistentry##1{%
2883 \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
2884 % \caption@ifhyccap{%
2885 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2886 % \def\hyper@makecurrent##1{%
2887 % \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2888 % \caption@makestart{##1}%
2889 % \let\Hy@LT@currentHlabel\@currentHlabel
2890 % \let\Hy@LT@currentHref\@currentHref
2891 % \def\hyper@makecurrent###1{%
2892 % \let\@currentHlabel\Hy@LT@currentHlabel
2893 % \let\@currentHref\Hy@LT@currentHref}}%
2894 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
2895 % \def\ContinuedFloat{\noalign{%
2896 % \gdef\caption@setContinuedFloat{%
2897 % \let\caption@resetContinuedFloat\@gobble}%
2898 % \def\caption@setoptions####1{%
2899 % \g@addto@macro\caption@setContinuedFloat{%
2900 % \caption@setoptions{####1}}}%
2901 % \let\@capttype\LTcapttype
2902 % \caption@ORI@ContinuedFloat}}%
2903 % }{%
2904 % \def\ContinuedFloat{\noalign{%
2905 % \caption@Error{%
2906 % \noexpand\ContinuedFloat inside longtables\MessageBreak
2907 % is only available with 'hyccap=true'}}}%
2908 % }%
2909 % \global\let\caption@setContinuedFloat\@empty
2910 \def\ContinuedFloat{\noalign{%
2911 \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2912 \caption@ORI@LT@array}}%

```

\LT@c@ption The original implementation:

```

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

```

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

```
2913 \long\def\LT@caption#1[#2]#3{%
2914     \LT@makecaption#1{\csname fnum@\LTcapttype\endcsname}{#3}%
2915     \LT@captionlistentry{#2}}%
```

```
\LT@makecaption \LT@makecaption{\langle cmd\rangle}{\langle label\rangle}{\langle text\rangle}
```

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:

```
2916 \renewcommand\LT@makecaption[3]{%
2917     \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`.)

```
2918     \caption@settype*\LTcapttype
2919     \ifdim\LTcapwidth=4in \else
2920         \setcaptionwidth\LTcapwidth
2921     \fi
2922     \caption@setoptions{longtable}%
2923 %     \caption@setContinuedFloat
2924     \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!

```
2925     \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
```

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

```
2926     \caption@startrue#1\caption@starfalse
2927     \caption@resetContinuedFloat\LTcapttype
2928     \caption@begin\LTcapttype
2929     \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...)

```
2930 \vskip-\ht\strutbox
```

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

```
2931 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2932 \caption@@make{#2}{#3}\endgraf
2933 \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2934 \caption@end}}%

2935 }{ }
```

2.15.8 The picinpar package

```
2936 \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.

```
2937 \long\def\figwindow[#1,#2,#3,#4] {%
2938 \caption@window{figure}%
2939 \caption@setoptions{figwindow}%
2940 \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%

2941 \long\def\tabwindow[#1,#2,#3,#4] {%
2942 \caption@window{table}%
2943 \caption@setoptions{tabwindow}%
2944 \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.

```
2945 \newcommand*\caption@window[1]{%
2946 \let\caption@boxrestore\@parboxrestore
2947 \let\caption@makecaption\caption@@make
2948 \caption@setautoposition b%
2949 \caption@settype{#1}%
2950 \caption@clearmargin}%

\caption@wincaption
```

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

```
2951 \newcommand\caption@wincaption[1]{%
```

This will be done twice for every `figwindow` & `tabwindow` caption – on the first run `\picwd` is 0pt, on the second run `\picwd` is `\hsize`.

```
2952 \ifdim\picwd=\z@
2953 \let\caption@makecurrent\@gobbletwo
2954 \let\caption@@start\relax
2955 \caption@prepareslc
2956 \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 L^{OF}] {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.

```

2957 \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2958 \if\@tempa*%
2959 \let\@tempa\@firstofone
2960 \else\if\@tempa[%
2961 \let\@tempa\@firstofone
2962 \else
2963 \let\@tempa\@empty
2964 \fi\fi
2965 \expandafter\caption\@tempa{#1}}%

2966 {}{}

```

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.

```

2967 \newcommand*\piccaptiontype[1]{\def\@piccaptiontype{#1}}

2968 \caption@ifpackageloaded{picins}{%

```

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

```

2969 \@ifundefined{piccaptiontype}{%
2970 \caption@iftype{%
2971 \let\@piccaptiontype\@capttype
2972 }{%
2973 \def\@piccaptiontype{figure}%
2974 }%
2975 }{}%
2976 \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.

```
2977 \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.

```

2978 \let\caption@ORI@make@piccaption\make@piccaption
2979 \def\make@piccaption{%
2980   \let\caption@ORI\caption
2981   \long\def\caption[##1]##2{%
2982     \caption@freezeHref % will be defrosted in \ivparpic
2983     \caption@settype\@piccaptiontype
2984 %   \ifnum\c@piccaptionpos>2\relax
2985     \caption@clearmargin
2986 %   \else
2987 %     \captionwidth\z@ % do not use "width=" setting
2988 %   \fi
2989   \caption@setoptions{parpic}%
2990   \let\caption@boxrestore\@parboxrestore
2991   \caption@setautoposition b%
2992   \expandafter\expandafter\expandafter\caption@ORI
2993     \expandafter\expandafter\expandafter[%
2994     \expandafter\expandafter\expandafter{%
2995     \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
2996   \caption@ORI@make@piccaption
2997   \let\caption\caption@ORI}%

```

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

```

2998 \let\caption@ORI@ivparpic\ivparpic
2999 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
3000   \let\caption@ORI@noindent\noindent
3001   \def\noindent{%
3002     \caption@defrostHref
3003     \let\noindent\caption@ORI@noindent
3004     \noindent}%
3005   \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
3006   \let\noindent\caption@ORI@noindent}%

```

```

3007 }{%
3008 \let\piccaptiontype\@undefined
3009 }

```

2.15.10 The rotating package

```

3010 \caption@ifPackageLoaded{rotating}[1995/08/22 v2.10]{%

```

```

\rotcaption Make \rotcaption* work.

```

```

3011 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
3012 % \let\@rotcaption\@undefined

```

```

\rotcaptionof Make \rotcaptionof(*) work.

```

```

3013 \def\rotcaptionof{%
3014 \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

```

```

\@makerotcaption Original (bugfixed) code:

```

```

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

```

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

```

3015 \long\def\@makerotcaption#1#2{%
3016 \ifdim\captionwidth=\z@
3017 \setcaptionwidth{.8\textheight}%
3018 \caption@slc{#1}{#2}{.8\vszise}{%
3019 \let\caption@makerot\caption@@make
3020 \caption@clearmargin
3021 % \long\def\caption@parbox##1##2{\hbox{\hszise=.8\textheight\relax##2}}%
3022 % (not needed because \rotatebox uses an \hbox anyway)
3023 \let\caption@parbox\@secondoftwo}%
3024 \caption@set@bool\caption@ifslc0% been there, done that
3025 \fi
3026 \rotatebox{90}{\caption@makerot{#1}{#2}}%
3027 \nobreak\hspace{12pt}}%
3028 \newcommand\caption@makerot[2]{%
3029 \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
3030 \caption@For{typelist}{%
3031 \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
3032 \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
3033 }{}

```

2.15.11 The sidecap package

```

3034 \caption@ifpackageloaded{sidecap}[1999/05/11 v1.4d]{%
3035   \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.)
3036   \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.
3037   \let\caption@ORI@SC@zfloat\SC@zfloat
3038   \def\SC@zfloat#1#2#3[#4]{%

First we use the original definition, but save & restore \caption so \caption@freeze
will work correctly.
3039       \let\caption@ORI\caption
3040       \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
3041       \let\caption\caption@ORI

Since the sidecap package uses our \caption code outside the environment the reg-
ular \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.
3042       \caption@settype*{#2}%
3043       \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.
3044   \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
3045   \def\endSC@FLOAT{%

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

3046       \let\caption@ORI@settype\caption@settype
3047       \def\caption@settype##1{% will be done in \xfloat
3048         \caption@ORI@settype*{##1}% do not change \@currentlabel
3049         \caption@setSC@justify
3050 %%% \caption@setoptions{SCfloat}%
3051         \caption@setoptions{SC\@captive}%
3052         \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

3053   \def\caption@setSC@justify{%
3054     \caption@clearmargin
3055     \ifundefined{SC@justify}{}{}%
```



```

3056         \ifx\SC@justify\@empty \else
3057             \let\caption@hj\SC@justify
3058             \let\SC@justify\@empty
3059         \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!

```

3060     \let\caption\SC@orig@caption
3061     \def\SC@orig@caption[#1]##2{\caption@defrost}%

```

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

```

3062     \caption@setSC@justify % for compatibility mode
3063     \caption@ORI@endSC@FLOAT}%

3064     \newcommand*\caption@For@SC[2]{%
3065         \def#1{b}% = \sidecaptionvpos{#2}{b} (v1.6)
3066         \newenvironment{SC#2}%
3067             {\SC@float[#1]{#2}}{\endSC@float}%
3068         \newenvironment{SC#2*}%
3069             {\SC@dblfloat[#1]{#2}}{\endSC@dblfloat}}%
3070     \@onlypreamble\caption@For@SC
3071     \caption@For{typelist}{%
3072         \expandafter\caption@For@SC\cscname SC@#1@vpos\endcsname{#1}}%
3073 }{}

```

2.15.12 The subfigure package

```

3074 \caption@IfPackageLoaded{subfigure}[2002/01/23 v2.1]{%

```

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

```

3075     \def\sf@ifpositiontop{%
3076         \ifx\@captype\@undefined
3077             \expandafter\@gobbletwo
3078         \else\ifx\@captype\relax
3079             \expandafter\expandafter\expandafter\@gobbletwo
3080         \else
3081             \expandafter\expandafter\expandafter\sf@if@position@top
3082         \fi\fi}

3083     \def\sf@if@position@top{%
3084         \@ifundefined{if\@captype topcap}%
3085             {\@gobbletwo}%
3086             {\@nameuse{if\@captype topcap}%
3087                 \expandafter\@firstoftwo
3088             \else
3089                 \expandafter\@secondoftwo
3090             \fi}}

3091 }{}

```

2.15.13 The supertabular and xtab packages

```

3092 \caption@ifPackageLoaded{supertabular}[2002/07/19 v4.1e]{%

\tablecaption Make \topcaption* and \bottomcaption* work.
3093 \renewcommand*\tablecaption{%
3094 \caption@star
3095 {\refstepcounter{table}}%
3096 {\caption@dblarg{\@xtablecaption}}}%

\@xtablecaption Make \nameref and \autoref work.
3097 \let\caption@ORI@xtablecaption\@xtablecaption
3098 \long\def\@xtablecaption[#1]#2{%
3099 \def\@currentlabelname{#2}%
3100 \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}

3101 \long\def\ST@caption#1[#2]#3{\par%
3102 \caption@settype*{#1}%
3103 \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.

3104 \def\caption@fixposition{%
3105 \caption@setposition{\if@topcaption t\else b\fi}}%

3106 \caption@beginex{#1}{#2}{#3}%
3107 \caption@boxrestore
3108 \caption@normalsize
3109 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3110 \caption@end}%

3111 {}

3112 \caption@ifPackageLoaded{xtab}[2000/04/09 v2.3]{%

\tablecaption Make \topcaption* and \bottomcaption* work.
3113 \renewcommand*\tablecaption{%
3114 \caption@star
3115 {\refstepcounter{table}}%
3116 {\caption@dblarg{\@xtablecaption}}}%

```

```

\@xtablecaption Make \nameref and \autoref work.
3117 \let\caption@ORI@xtablecaption\@xtablecaption
3118 \long\def\@xtablecaption[#1]#2{%
3119     \def\@currentlabelname{#2}%
3120     \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}}%

3121 \long\def\ST@caption#1[#2]#3{\par%
3122     \caption@settype*{#1}%
3123     \caption@setoptions{xtabular}%
3124     \def\caption@fixposition{%
3125         \caption@setposition{\if@topcaption t\else b\fi}}%
3126     \@initisotab
3127     \caption@beginex{#1}{#2}{#3}%
3128     \caption@boxrestore
3129     \caption@normalsize
3130     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
3131     \caption@end
3132     \global\advance\ST@pageleft -\PWSTcapht
3133     \ST@trace\tw@{Added caption. Space left for xtabular: \the\ST@pageleft}}%
3134 }{}

```

2.15.14 The threeparttable package

```

3135 \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.
3136 \let\caption@ORI@threeparttable\threeparttable
3137 \renewcommand*\threeparttable{%
3138     \caption@settype{table}%
3139     \caption@setposition a% ?
3140     \caption@clearmargin
3141     \caption@setoptions{threeparttable}%
3142     \caption@ORI@threeparttable}%

```

`\measuredfigure` Same here...

```
3143 \let\caption@ORI@measuredfigure\measuredfigure
3144 \renewcommand*\measuredfigure{%
3145   \caption@settype{figure}%
3146   \caption@setposition a% ?
3147   \caption@clearmargin
3148   \caption@setoptions{measuredfigure}%
3149   \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}

3150 \def\TPT@caption#1[#2]#3{%
3151   \gdef\TPT@docapt{%
3152     \global\let\TPT@docapt\@undefined
3153     \caption@setautoposition\caption@TPT@position
3154     \TPT@LA@caption{#1}[{#2}]{#3}}%
3155   \ifx\TPT@hsize\@empty
3156     \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
3157     \gdef\caption@TPT@position{t}%
3158     \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
3159   \else
3160     \def\caption@TPT@position{b}%
3161     \TPT@docapt
3162   \fi
3163   \ignorespaces}%
3164   %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
3165   \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%

3166 {} }
```

2.15.15 The wrapfig package

```
3167 \caption@IfPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)
```

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

```
3168 \providecommand*\float@ifstyle[1]{%
3169   \expandafter\ifx\csname fst@#1\endcsname\relax
3170   \expandafter\@secondoftwo
3171   \else
3172   \expandafter\@firstoftwo
3173   \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.

```

3174 \newcommand*\caption@restylewrapfloat[1]{%
3175   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
3176     \csname wrap#1\endcsname
3177   \caption@Error{%
3178     For a successful cooperation of the 'wrapfig' package\MessageBreak
3179     with the 'float' package you should load the 'wrapfig'\MessageBreak
3180     package *after* the 'float' package}%
3181   \else
3182     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
3183       \csname wrap#1\endcsname
3184     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
3185     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
3186       \csname wrap#1\endcsname
3187   \fi}%

```

`\caption@wrapfloat`

```

3188 \newcommand*\caption@wrapfloat[1]{%
3189   \caption@settype*{#1}%
3190   \float@ifstyle{#1}{%
3191     \ifx\WF@floatstyhook\@undefined
3192       \caption@Error{%
3193         For a successful cooperation of the 'wrapfig' package\MessageBreak
3194         with the 'float' package you should use at least\MessageBreak
3195         'wrapfig' version 3.6}%
3196     \else
3197       \float@dostyle{#1}%
3198     \fi}%
3199   \caption@clearmargin
3200   \caption@setoptions{wrapfloat}%
3201   \caption@setoptions{wrap#1}%
3202   \@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`.)

```

3203 \caption@restylewrapfloat{figure}%
3204 \caption@restylewrapfloat{table}%
3205 \caption@For{typelist}{%
3206   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
3207   \caption@restylewrapfloat{#1}}%
3208 \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.

```

3209 \@ifundefined{float@restyle}{}{%
3210   \toks@=\expandafter{\float@restyle{#1}% (env may or may not be defined)

```

```

3211         \caption@restylewrapfloat{#1}}%
3212         \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
3213         \@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.

3214     \let\caption@ORI@wrapfloat\wrapfloat
3215     \def\wrapfloat#1{%
3216         \float@ifstyle{#1}{%
3217             \caption@Error{%
3218                 For a successful cooperation of the 'wrapfig' package\MessageBreak
3219                 with the 'float' package you should load the 'wrapfig'\MessageBreak
3220                 package *right after* the 'float' package}}}%
3221         \caption@ORI@wrapfloat{#1}}%

3222     \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:

3223 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
3224     \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
3225     \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
3226     \caption@start
3227     \ifdim\hsize>\z@ \@parboxrestore \else
3228     \setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
3229     \ignorespaces \fi}%

3230 } {}

```

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