

# The Implementation of the caption package<sup>\*</sup>

Axel Sommerfeldt  
`caption@sommerfee.de`

2007/12/23

## 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X document – it just provides functionality which can be used by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> packages which typesets captions, for example the caption and subfig packages.

The caption package itself redefines the L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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 <sup>1</sup>
<code>caption-deu.pdf</code>	The German documentation

---

<sup>\*</sup>This package has version number v3.1f, last revised 2007/12/23.

<sup>1</sup>Thanks a lot to Olga Lapko for this translation

# Contents

<b>1</b>	<b>Kernel</b>	<b>4</b>
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 . . . . .	18
1.12	Fonts . . . . .	19
1.13	Justifications . . . . .	21
1.13.1	The ragged2e package . . . . .	22
1.14	Vertical spaces before and after captions . . . . .	22
1.15	Positioning . . . . .	23
1.16	Hooks . . . . .	24
1.17	Lists . . . . .	24
1.18	Debug option . . . . .	25
1.19	Document classes & Babel support . . . . .	25
1.19.1	The standard L <sup>A</sup> T <sub>E</sub> X classes . . . . .	25
1.19.2	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes . . . . .	25
1.19.3	The beamer class . . . . .	27
1.19.4	The KOMA-Script classes . . . . .	28
1.19.5	The NTG Dutch classes . . . . .	28
1.19.6	The frenchb Babel option . . . . .	29
1.19.7	The frenchle/pro package . . . . .	29
1.20	Execution of options . . . . .	30
1.21	Making an ‘List of’ entry . . . . .	30
1.22	Typesetting the caption . . . . .	31
1.23	Types & sub-types . . . . .	34
1.24	subfig package adaption . . . . .	37
<b>2</b>	<b>Main package</b>	<b>39</b>
2.1	Identification . . . . .	39
2.2	Loading the kernel . . . . .	39
2.3	Check against incompatible packages . . . . .	39
2.4	Check document class . . . . .	39

2.5	Adaption to the $\mathcal{AMS}$ & SMF document classes	40
2.6	Emulation of the KOMA-Script commands	40
2.7	Declaration of options	42
2.7.1	Options for figure and table	42
2.7.2	Miscellaneous options	43
2.7.3	caption v1.x compatibility options	43
2.7.4	caption2 v2.x compatibility options	44
2.7.5	Obsolete caption v3.0 options	44
2.7.6	fltpage package support options	44
2.7.7	hyperref package support options	44
2.8	Processing of options	44
2.9	<code>\captionof</code> and <code>\ContinuedFloat</code>	44
2.10	Internal helpers	48
2.11	<code>\caption</code> , <code>\@caption</code> , and <code>\@makecaption</code>	50
2.12	Support for sub-captions	56
2.13	Document class & Babel package support	58
2.13.1	The $\mathcal{AMS}$ & SMF classes	58
2.13.2	The beamer class	58
2.13.3	The KOMA-Script classes	59
2.13.4	The frenchb Babel option	59
2.13.5	The frenchle/pro package	59
2.14	Package support	60
2.14.1	The float package	62
2.14.2	The floatflt package	65
2.14.3	The fltpage package	65
2.14.4	The hyperref package	68
2.14.5	The hypcap package	71
2.14.6	The listings package	72
2.14.7	The longtable package	72
2.14.8	The picinpar package	75
2.14.9	The picins package	76
2.14.10	The rotating package	78
2.14.11	The sidecap package	79
2.14.12	The subfigure package	80
2.14.13	The supertabular package	81
2.14.14	The threeparttable package	81
2.14.15	The wrapfig package	82

# 1 Kernel

## 1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2007/12/21 v3.1f caption3 kernel (AR)]
```

## 1.2 Generic helpers

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

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

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

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

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

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

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

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

would expand to yes.

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

```

```

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

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

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

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

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

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

\caption@ifempty@list \caption@ifempty@list{\langle cmd\rangle}{\langle true\rangle}{\langle false\rangle}
50 \newcommand*\caption@ifempty@list[1]{%
51 \ifx#1\@undefined
52 \expandafter\@firstoftwo
53 \else\ifx#1\relax
54 \expandafter\expandafter\expandafter\@firstoftwo
55 \else\ifx#1\@empty
56 \expandafter\expandafter\expandafter\expandafter
57 \expandafter\expandafter\expandafter\@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{\langle name\rangle}{\langle value\rangle}
\caption@ifbool (with value = false/true/no/yes/off/on/0/1)
\caption@undefbool \caption@ifbool{\langle name\rangle}{\langle if-clause\rangle}{\langle else-clause\rangle}
\caption@undefbool{\langle name\rangle}

```

```

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

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

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

\caption@teststar \caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
\caption@teststar@{<cmd>}{<star arg/token>}{<non-star arg/token>}
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 \newcommand*\caption@ifatletter{%
78   \ifnum\the\catcode`\@=11
79     \expandafter\@firstoftwo
80   \else
81     \expandafter\@secondoftwo
82   \fi}

\caption@withoptargs \caption@withoptargs{<cmd>}
83 \newcommand*\caption@withoptargs[1]{%
84   \@ifstar
85   {\def\caption@tempa{*}\caption@@withoptargs#1}%
86   {\def\caption@tempa{}\caption@@withoptargs#1}}

87 \def\caption@@withoptargs#1{%
88   \@ifnextchar[%]
89   {\caption@@@withoptargs#1}%
90   {\caption@@@withoptargs#1}}

91 \def\caption@@@withoptargs#1[#2]{%
92   \l@addto@macro\caption@tempa{[#{#2}]}%
93   \caption@@withoptargs#1}

94 \def\caption@@@withoptargs#1{%
95   \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.)

96 \newcommand\caption@DoCheckCommand[2]{%

```

```

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

107 \let\caption@CheckCommand\caption@DoCheckCommand
108 \@onlypreamble\caption@CheckCommand

109 \newcommand*\caption@ifCheckCommand{%
110 \ifx\caption@CheckCommand\@gobbletwo
111 \let\next\@firstoftwo
112 \else
113 \let\next\@secondoftwo
114 \fi
115 \let\caption@CheckCommand\caption@DoCheckCommand
116 \next}
117 \@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.

```

118 \let\caption@begindocumenthook\@empty
119 \let\caption@@begindocumenthook\@empty

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

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

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

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

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

```

### 1.3 Errors

```

\caption@Warning \caption@Error{<message>}
\caption@WarningNoLine 135 \newcommand*\caption@Warning[1]{%
\caption@Error 136 \caption@WarningNoLine{#1\on@line}}
\caption@KV@err 137 \newcommand*\caption@WarningNoLine[1]{%

```

```

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

```

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

```

\caption@eh 142 \newcommand*\caption@wh{%
143 See the caption package documentation for explanation.}
144 \newcommand*\caption@eh{%
145 If you do not understand this error, please take a closer look\MessageBreak
146 at the documentation of the 'caption' package, especially the\MessageBreak
147 section about errors.\MessageBreak\@ehc}

```

## 1.4 Using the keyval package

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

```

148 \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.

```

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

```

```

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

```

Analogous to \@onlypreamble from L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

```

151 \providecommand*\@preamble@keys{}
152 \providecommand*\@onlypreamble@key[2]{\@cons\@preamble@keys{{#1}{#2}}}
153 \@onlypreamble\@onlypreamble@key
154 \@onlypreamble\@preamble@keys
155 \providecommand*\@notprerr@key[1]{\KV@err{Can be used only in preamble}}
156 \caption@AtBeginDocument*{%
157 \def\@elt#1#2{\expandafter\let\csname KV@#1@#2\endcsname\@notprerr@key}%
158 \@preamble@keys
159 \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<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>), 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.

```

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

```

```

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

```

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



```

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

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

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

```

`\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.

```

180 \newcommand*\captionsetup{%
181   \caption@teststar@\@captionsetup\@gobble\@firstofone}

182 \newcommand*\@captionsetup[1]{%
183   \@ifnextchar[{\caption@setup@options#1}\caption@setup}

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

185 \def\caption@setup@options#1[#2]#3{%
186   \@bsphack
187   % #1 is \@gobble in star form, and \@firstofone otherwise
188   #1{\caption@addtooptlist{#2}}}%
189   \expandafter\caption@l@addto@list\csize caption@opt@#2\endcsname{#3}%
190   \@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.

```

191 \newcommand*\clearcaptionsetup{%
192   \caption@teststar@\@clearcaptionsetup\@gobble\@firstofone}

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

196 \def\@clearcaptionsetup[#1]#2{%
197   \@bsphack
198   \expandafter\caption@ifempty@list\csize caption@opt@#2\endcsname
199   {\caption@tempa{\caption@Warning{Option list ‘#2’ undefined}}}%
200   {\ifx,#1,%
201     \caption@clearsetup{#2}%
202   \else

```

```

203         \caption@@removefromsetup{#1}{#2}%
204     \fi}%
205 \esphack}

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

209 \newcommand*\caption@removefromsetup{%
210     \let\caption@tempa\@gobble
211     \caption@@removefromsetup}

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

```

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

241 \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.

```

242 \newcommand*\caption@setoptions[1]{%
243     \caption@Debug{options=#1}%
244     \expandafter\let\expandafter\caption@opt\csname caption@opt@#1\endcsname
```

```

245 \ifx\caption@opt\relax \else
246 \caption@xsetup\caption@opt
247 \caption@clearsetup{#1}%
248 \fi}

249 \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”).

```

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

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

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

263 \AtEndDocument{%
264 \caption@for@list\caption@optlist{%
265 \PackageWarningNoLine{caption}{%
266 Unused \string\captionsetup{#1}
267 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.

```

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

269 \long\def\caption@@setkeys[#1]#2#3{%
270 \@bsphack

271 \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
272 \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
273 \expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
274 \let\KV@errx\KV@err
275 \edef\caption@keydepth{\caption@keydepth i}%

276 \caption@Debug{\protect\setkeys{#2}{#3}}%
277 \setkeys{#2}{#3}%

278 \edef\caption@keydepth{\expandafter\@cdr\caption@keydepth\@nil}%
279 \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
280 \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname

281 \@esphack}

282 \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.
283 \newcommand*\caption@ExecuteOptions[2]{%
284   \@expandtwoargs\caption@setkeys{#1}{#2}}%
285 \@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.)
286 \newcommand*\caption@ProcessOptions{%
287   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
288 \@onlypreamble\caption@ProcessOptions

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

```

## 1.5 Margin resp. width

```

\captionmargin \captionmargin and \captionwidth contain the extra margin resp. the total
\captionwidth width used for captions. Please never set these values in a direct way, they are just acces-
sible in user documents to provide compatibility to vI.x.
Note that we can only set one value at a time, ‘margin’ or ‘width’. If \captionwidth
is not zero we will take this value afterwards, otherwise \captionmargin and
\captionmargin@.
310 \newdimen\captionmargin
311 \newdimen\captionmargin@
312 \newdimen\captionwidth

313 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
314 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
315 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
316 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}
317 \DeclareCaptionOption{NoValue}{oneside}{\caption@set@bool\caption@iftwoside0}

```

```

318 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
319 \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.
320 \newcommand*\setcaptionmargin{%
321   \caption@teststar\caption@setmargin\@gobble\@firstofone}

322 \newcommand*\caption@setmargin[2]{%
323   #1{\captionwidth\z@}%
324   \caption@@setmargin#2,#2,\@nil}

325 \def\caption@@setmargin#1,#2,#3\@nil{%
326   \setlength\captionmargin@{#2}%
327   \setlength\captionmargin{#1}%
328   \addtolength\captionmargin@{-\captionmargin}}

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

\caption@counter This counter numbers the captions. At the moment it will be used inside \caption@ifoddpage
only.
333 \newcommand*\caption@thecounter{0}
334 \newcommand*\caption@stepcounter{%
335   \@tempcnta\caption@thecounter
336   \advance\@tempcnta\@ne
337   \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.)
338 \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.
339 \newcommand*\caption@thepage{\the\c@page}

\caption@label This command is a modified version of \label from LATEX2ε. It will be used inside
\caption@ifoddpage and \FP@helpNote.
340 \newcommand*\caption@label[1]{%
341   \caption@@label
342   \protected@write\@auxout{\let\caption@thepage\relax}%
343     {\string\caption@newlabel{#1}{\caption@thepage}}
344 \newcommand*\caption@@label{%
345   \global\let\caption@@label\relax
346   \protected@write\@auxout{}%
347     {\string\providecommand*\string\caption@newlabel[2]{}}}

```

`\caption@pageref` This command is a modified version of `\pageref` from L<sup>A</sup>T<sub>E</sub>X2e. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

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

```

`\caption@ifoddpage` At the moment this macro uses an own label...ref mechanism, but an alternative implementation method would be using the `refcount` package[23] and `\ifodd\getpagerefnumber{...}`.  
*Note:* This macro re-defines itself so the `.aux` file will only be used once per group.

```

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

```

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

```

366 \newcommand*\caption@setoptcmd[2]{%
367   \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 368 \newdimen\caption@indent
\caption@hangindent 369 \newdimen\caption@parindent
370 \newdimen\caption@hangindent

371 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
372   \setlength\caption@indent{#1}}
373 \DeclareCaptionOption{indentation}[\leftmargini]{%
374   \setlength\caption@indent{#1}}
375 \DeclareCaptionOption{parindent}{%
376   \setlength\caption@parindent{#1}}
377 \DeclareCaptionOption{hangindent}{%
378   \setlength\caption@hangindent{#1}}
379 \DeclareCaptionOption{parskip}{%
380   \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).<sup>2</sup>

```

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

```

## 1.7 Styles

```

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

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

```

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

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

```

406 \newcommand*\caption@setstyle{%
407   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

408 \newcommand*\caption@@setstyle[2]{%
409   \ifundefined{caption@sty@#2}%
410     {#1\caption@Error{Undefined style '#2'}}%
411     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
412     \ifx\caption@setstyle@flag\@undefined

```

<sup>2</sup>This 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!)

```

413     \let\caption@setstyle@flag\relax
414     \caption@resetstyle
415     \caption@xsetup\caption@sty
416     \let\caption@setstyle@flag\@undefined
417   \else
418     \caption@xsetup\caption@sty
419   \fi
420   \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
421   \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
422   \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!*

```

423 \newcommand*\caption@resetstyle{%
424   \caption@setup{%
425     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
426     justification=justified,font=,size=,labelfont=,textfont=,%
427     margin=0pt,minmargin=0,maxmargin=0,%
428     indent=0pt,parindent=0pt,hangindent=0pt,%
429     singlelinecheck=1,strut=1}%
430   \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<sup>A</sup>T<sub>E</sub>X document classes (and was called ‘default’ in the caption package v3.0), the second one matches the document class actually used.

```

431 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
432 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{}
433   format=default,labelsep=default,textformat=default,%
434   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 `indenton=` option.

```

435 \newcommand*\DeclareCaptionFormat{%
436   \caption@teststar\caption@declareformat\@gobble\@firstofone}
437 \@onlypreamble\DeclareCaptionFormat

438 \newcommand*\caption@declareformat[2]{%
439   \@dblarg{\caption@@declareformat#1{#2}}
440 \@onlypreamble\caption@declareformat

441 \long\def\caption@@declareformat#1#2[#3]#4{%
442   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
443   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
444   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
445 \@onlypreamble\caption@@declareformat

446 \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`).

```

447 \newcommand*\caption@setformat[1]{%
448   \ifundefined{caption@fmt@#1}%
449     {\caption@Error{Undefined format `#1'}}%
450     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
451      \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
452      \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

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

```

453 \DeclareCaptionFormat{plain}{#1#2#3\par}
454 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
455   \caption@ifin@list\caption@lsepclist\caption@lsepname
456   {\caption@Error{%
457     The option `labelsep=\caption@lsepname' does not work\MessageBreak
458     with `format=hang'}}%
459   {\@hangfrom{#1#2}%
460    \advance\caption@parindent\hangindent
461    \advance\caption@hangindent\hangindent
462    \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

463 \def\caption@fmt@default{\caption@fmt@plain}
464 \def\caption@ifh@default{\caption@ifh@plain}

```

## 1.9 Label formats

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

```

465 \newcommand*\DeclareCaptionLabelFormat[2]{%
466   \global\@namedef{caption@lfmt@#1}##1##2##3{#2}}
467 \@onlypreamble\DeclareCaptionLabelFormat
468 \DeclareCaptionOption{labelformat}{\caption@setlabelformat{#1}}

```

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

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

```

469 \newcommand*\caption@setlabelformat[1]{%
470   \ifundefined{caption@lfmt@#1}%
471     {\caption@Error{Undefined label format `#1'}}%
472     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}

```

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

```

473 \DeclareCaptionLabelFormat{empty}{}
474 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
475 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace} (#2)}
476 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace} #2)}

```

‘default’ usually maps to ‘simple’.

```

477 \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.
478 \newcommand\DeclareCaptionLabelSeparator{%
479   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
480 \@onlypreamble\DeclareCaptionLabelSeparator
481 \newcommand\caption@declarelabelseparator[3]{%
482   \global\expandafter\let\csname caption@ifl@#2\endcsname#1%
483   \global\long\@namedef{caption@lsep@#2}{#3}%
484   \caption@@declarelabelseparator{#2}#3\\@nil}
485 \@onlypreamble\caption@declarelabelseparator
486 \long\def\caption@@declarelabelseparator#1#2\\#3\@nil{%
487   \def\@tempa{#3}\ifx\@tempa\@empty \else
488     \caption@gaddto@list\caption@lsep@list{#1}%
489   \fi}
490 \@onlypreamble\caption@@declarelabelseparator
491 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
492 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

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

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

```
493 \newcommand*\caption@setlabelseparator[1]{%
494   \ifundefined{caption@lsep@#1}%
495     {\caption@Error{Undefined label separator '#1'}}%
496     {\edef\caption@lsepname{#1}%
497       \expandafter\let\expandafter\caption@ifl\csname caption@ifl@#1\endcsname
498       \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’.

```
499 \DeclareCaptionLabelSeparator{none}{}
500 \DeclareCaptionLabelSeparator{colon}{: }
501 \DeclareCaptionLabelSeparator{period}{. }
502 \DeclareCaptionLabelSeparator{space}{ }
503 \DeclareCaptionLabelSeparator*{quad}{\quad}
504 \DeclareCaptionLabelSeparator*{newline}{\\}
505 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}
```

‘default’ usually maps to ‘colon’.

```
506 \def\caption@lsep@default{\caption@lsep@colon}
507 \def\caption@ifl@default{\caption@ifl@colon}
```

## 1.11 Text formats

```
\DeclareCaptionTextFormat {<name>} {<code with #1>}
508 \newcommand*\DeclareCaptionTextFormat[2]{%
509   \global\long\@namedef{caption@tfmt@#1}##1{#2}}
510 \@onlypreamble\DeclareCaptionTextFormat
```

```

511 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
512 \DeclareCaptionOption{strut}{\caption@set@bool\caption@ifstrut{#1}}

```

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

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

```

513 \newcommand*\caption@settextformat[1]{%
514   \ifundefined{caption@tfmt@#1}%
515     {\caption@Error{Undefined text format `#1'}}%
516     {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

517 \DeclareCaptionTextFormat{simple}{#1}
518 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

## 1.12 Fonts

```

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

```

```

DeclareDefaultCaptionFont \DeclareDefaultCaptionFont{<name>}{<code>}
523 \newcommand*\DeclareDefaultCaptionFont[2]{%
524   \global\@namedef{caption#1@default}{#2}}
525 \@onlypreamble\DeclareDefaultCaptionFont

526 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
527 \DeclareCaptionOption{font+}{\caption@addtofont{font}{#1}}
528 \DeclareDefaultCaptionFont{font}{}

529 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
530 \DeclareCaptionOption{labelfont+}{\caption@addtofont{labelfont}{#1}}
531 \DeclareDefaultCaptionFont{labelfont}{}

532 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
533 \DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
534 \DeclareDefaultCaptionFont{textfont}{}

```

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

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

```

535 \newcommand*\caption@setfont[1]{%
536   \expandafter\let\csname caption#1\endcsname\empty
537   \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.

```

538 \newcommand*\caption@addtofont[2]{%
539   \begingroup
540     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
541     \define@key{caption@fnt}{default}[]{}%

```

```

542     \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
543     \caption@setkeys[caption]{caption@fnt}{#2}%
544     \global\let\caption@tempa\caption@fnt
545 \endgroup
546 \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`.

```

547 \newcommand*\caption@font{%
548   \caption@teststar\caption@@font\@firstofone
549   {\caption@setkeys[caption]{caption@fnt}}
550 \newcommand*\caption@@font[2]{%
551   \begingroup
552   \def\caption@fnt{\endgroup}%
553   #1{#2}%
554   \caption@fnt}

```

These are the pre-defined font code snippets.

```

555 \DeclareCaptionFont{normalcolor}{\normalcolor}
556 \DeclareCaptionFont{color}{\color{#1}}

557 \DeclareCaptionFont{normalfont}{\normalfont}
558 \DeclareCaptionFont{up}{\upshape}
559 \DeclareCaptionFont{it}{\itshape}
560 \DeclareCaptionFont{sl}{\slshape}
561 \DeclareCaptionFont{sc}{\scshape}
562 \DeclareCaptionFont{md}{\mdseries}
563 \DeclareCaptionFont{bf}{\bfseries}
564 \DeclareCaptionFont{rm}{\rmfamily}
565 \DeclareCaptionFont{sf}{\sffamily}
566 \DeclareCaptionFont{tt}{\ttfamily}

567 \DeclareCaptionFont{scriptsize}{\scriptsize}
568 \DeclareCaptionFont{footnotesize}{\footnotesize}
569 \DeclareCaptionFont{small}{\small}
570 \DeclareCaptionFont{normalsize}{\normalsize}
571 \DeclareCaptionFont{large}{\large}
572 \DeclareCaptionFont{Large}{\Large}

573 \DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespace}% normally 1
574 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
575 \DeclareCaptionFont{doublespacing}{\doublespacing}
576 \DeclareCaptionFont{stretch}{\setstretch{#1}}
577 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }

578 %\DeclareCaptionFont{normal}{%
579 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
580 \DeclareCaptionFont{normal}{%
581   \caption@font*{%
582     \KV@caption@fnt@normalcolor\@unused
583     \KV@caption@fnt@normalfont\@unused
584     \KV@caption@fnt@normalsize\@unused
585     \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.

```
586 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
587 \DeclareDefaultCaptionFont{size}{}
```

## 1.13 Justifications

```
\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
588 \newcommand*\DeclareCaptionJustification[2]{%
589   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
590   \DeclareCaptionFont{#1}{#2}}
591 \@onlypreamble\DeclareCaptionJustification

\DeclareDefaultCaptionJustification{<code>}
592 \newcommand*\DeclareDefaultCaptionJustification[1]{%
593   \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
594   \DeclareDefaultCaptionFont{@hj}{#1}}
595 \@onlypreamble\DeclareDefaultCaptionJustification

596 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
597 \DeclareDefaultCaptionJustification{}

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

These are the pre-defined justification code snippets.

```
599 \DeclareCaptionJustification{justified}{}
600 \DeclareCaptionJustification{centering}{\centering}
601 \DeclareCaptionJustification{centerfirst}{\centerfirst}
602 \DeclareCaptionJustification{centerlast}{\centerlast}
603 \DeclareCaptionJustification{raggedleft}{\raggedleft}
604 \DeclareCaptionJustification{raggedright}{\raggedright}
```

```
\centerfirst Please blame Frank Mittelbach for the code of \centerfirst :-)
605 \providecommand\centerfirst{%
606   \let\\@centercr
607   \edef\caption@normaladjust{%
608     \leftskip\the\leftskip
609     \rightskip\the\rightskip
610     \parfillskip\the\parfillskip\relax}%
611   \leftskip\z@\@plus -1fil%
612   \rightskip\z@\@plus 1fil%
613   \parfillskip\z@skip
614   \noindent\hskip\z@\@plus 2fil%
615   \@setpar{\@@par\@restorepar\caption@normaladjust}}
```

```
\centerlast This is based on code from Anne Brüggemann-Klein[22]
616 \providecommand\centerlast{%
617   \let\\@centercr
618   \leftskip\z@\@plus 1fil%
619   \rightskip\z@\@plus -1fil%
620   \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.

```
621 \DeclareCaptionJustification{Centering}{%
622   \caption@ragged\Centering\centering}
623 \DeclareCaptionJustification{RaggedLeft}{%
624   \caption@ragged\RaggedLeft\raggedleft}
625 \DeclareCaptionJustification{RaggedRight}{%
626   \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.

```
627 \newcommand*\caption@ragged{%
628   \caption@Debug{We need ragged2e}%
629   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
630   \global\let\caption@ragged\caption@@ragged
631   \caption@ragged}

632 \caption@AtBeginDocument{%
633   \@ifundefined{caption@r@ragged2e}{%
634     \newcommand*\caption@@ragged{%
635       \caption@Warning{%
636         'ragged2e' support has been changed.\MessageBreak
637         Rerun to get captions right}%
638       \global\let\caption@ragged\@secondoftwo % suppress further warnings
639       \caption@ragged}%
640   }{%
641     \caption@Debug{We load ragged2e}%
642     \IfFileExists{ragged2e.sty}{%
643       \RequirePackage{ragged2e}%
644       \let\caption@@ragged\@firstoftwo
645     }{%
646       \newcommand*\caption@@ragged[2]{%
647         \@ifundefined{caption\string#1}{%
648           \caption@Warning{%
649             'ragged2e' package not loaded, therefore\MessageBreak
650             substituting \string#2 for \string#1\MessageBreak}%
651           \global\@namedef{caption\string#1}{}%
652           #2}%
653     }%
654   }}
```

### 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

```

655 \@ifundefined{abovecaptionskip}{%
656   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
657 \@ifundefined{belowcaptionskip}{%
658   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

659 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
660 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
661 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

## 1.15 Positioning

These macros handle the right position of the caption. Note that the position is actually *not* controlled by the caption3 kernel options, but by the user (or a specific package like the float package) instead. The user can put the `\caption` command wherever he likes! So this stuff is only to give us a *hint* where to put the right skips, the user usually has to take care for himself that this hint actually matches the right position.

```

662 \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.*

```

663 \newcommand*\caption@setposition[1]{%
664   \caption@ifinlist{#1}{d,default}{%
665     \let\caption@position\caption@defaultpos
666   }{\caption@ifinlist{#1}{t,top,above}{%
667     \let\caption@position\@firstoftwo
668   }{\caption@ifinlist{#1}{b,bottom,below}{%
669     \let\caption@position\@secondoftwo
670   }{\caption@ifinlist{#1}{a,auto}{%
671     \let\caption@position\@undefined
672   }{%
673     \caption@Error{Undefined position '#1'}%
674   }}}}}

```

```

\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’ correspondents to the `\@makecaption` implementation in the standard L<sup>A</sup>T<sub>E</sub>X document classes, but ‘auto’ should give better results in most cases.

```

675 %\caption@setdefaultpos{a}% default = auto
676 \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.)

```

677 \newcommand*\caption@iftop{%
678   \ifx\caption@position\@undefined
679     \let\caption@position\@secondoftwo
680 %   = \caption@setposition b%
681   \fi
682   \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.
683 \newcommand*\caption@fixposition{%
684   \ifx\caption@position\@undefined
685     \caption@autoposition
686   \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.
```

```
687 \newcommand*\caption@autoposition{%
688   \ifvmode
689     \edef\caption@tempa{\the\prevdepth}%
690     \caption@Debug{\protect\prevdepth=\caption@tempa}%
691     \ifdim\prevdepth>-\p@
692       \let\caption@position\@secondoftwo
693     \else
694       \let\caption@position\@firstoftwo
695     \fi
696 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
697   \else
698     \caption@Debug{no \protect\prevdepth}%
699     \let\caption@position\@secondoftwo
700 %   = \caption@setposition b%
701   \fi}
```

```
\caption@setautoposition \caption@setautoposition{<position>}
replaces the above algorithm by a different one (or a fixed position setting).
702 \newcommand*\caption@setautoposition[1]{%
703   \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.
704 \newcommand*\caption@beginhook{}
705 \newcommand*\caption@endhook{}
706 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
707 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}
```

## 1.17 Lists

```
708 \DeclareCaptionOption{list}{\caption@set@bool\caption@iflist{#1}}
709 \DeclareCaptionOption{listof}{\caption@set@bool\caption@iflist{#1}}
```



```

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

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

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

```

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

```

714 \newcommand*\caption@setlistformat[1]{%
715   \ifundefined{caption@lstfmt@#1}%
716     {\caption@Error{Undefined list format `#1'}}%
717     {\expandafter\let\expandafter\caption@lstfmt
718       \csname caption@lstfmt@#1\endcsname}}

```

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

```

719 \DeclareCaptionListFormat{empty}{}
720 \DeclareCaptionListFormat{simple}{#1#2}
721 \DeclareCaptionListFormat{parens}{#1(#2)}
722 \DeclareCaptionListFormat{subsimple}{#2}
723 \DeclareCaptionListFormat{subparens}{(#2)}
724 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}

```

## 1.18 Debug option

```

725 \DeclareCaptionOption{debug}[1]{%
726   \caption@set@bool\caption@ifdebug{#1}%
727   \caption@ifdebug
728     {\def\caption@Debug{\PackageInfo{caption}}}%
729     {\let\caption@Debug@gobble}}

730 \DeclareOption{debug}{\setkeys{caption}{debug}}
731 \setkeys{caption}{debug=0}

```

## 1.19 Document classes & Babel support

```

732 \caption@CheckCommand\@makecaption{\let\@makecaption\@undefined}

```

### 1.19.1 The standard L<sup>A</sup>T<sub>E</sub>X classes

```

733 \caption@CheckCommand\@makecaption{%
734   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
735   \long\def\@makecaption#1#2{%
736     \vskip\abovecaptionskip
737     \sbox\@tempboxa{#1: #2}%
738     \ifdim \wd\@tempboxa >\hsize
739       #1: #2\par
740     \else
741       \global \@minipagefalse
742       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
743     \fi
744     \vskip\belowcaptionskip}}

```

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

```

745 \ifundefined{@captionheadfont}{}{}%

```

```

746 \caption@CheckCommand\@makecaption{%
747   % amsart|amsproc|amsbook [2004/08/06 v2.20]
748   \long\def\@makecaption#1#2{%
749     \setbox\@tempboxa\vbox{\color@setgroup
750       \advance\hsize-2\captionindent\noindent
751       \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
752         {\@cdr#2\@nil}\{.\@captionfont\upshape\enspace#2}%
753       \unskip\kern-2\captionindent\par
754       \global\setbox\@ne\lastbox\color@endgroup}%
755     \ifhbox\@ne % the normal case
756       \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
757     \fi
758     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
759       \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
760     \else % tempboxa contained more than one line
761       \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
762         \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
763     \fi
764     \ifnum\@tempcnta<64 % if the float IS a figure...
765       \addvspace\abovecaptionskip
766       \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
767     \else % if the float IS NOT a figure...
768       \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
769       \nobreak
770       \vskip\belowcaptionskip
771     \fi
772   \relax
773 }
774 \caption@CheckCommand\@makecaption{%
775   % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
776   \long\def\@makecaption#1#2{%
777     \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
778     \setbox\@tempboxa\vbox{\color@setgroup
779       \advance\hsize-2\captionindent\noindent
780       \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
781         {\@cdr#2\@nil}\{\@addpunct{.}\@captionfont\upshape\enspace#2}%
782       \unskip\kern-2\captionindent\par
783       \global\setbox\@ne\lastbox\color@endgroup}%
784     \ifhbox\@ne % the normal case
785       \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
786     \fi
787     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
788       \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
789       \@tempdima\wd\@ne\advance\@tempdima-\captionindent
790       \wd\@ne\@tempdima
791     \else % tempboxa contained more than one line
792       \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
793         \unvbox\@tempboxa\parskip\z@skip
794         \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
795     \fi
796     \ifnum\@tempcnta<64 % if the float IS a figure...
797       \addvspace\abovecaptionskip
798       \noindent\kern\captionindent\box\@ne
799     \else % if the float IS NOT a figure...

```

```

800      \noindent\kern\captionindent\box\@ne
801      \nobreak
802      \vskip\belowcaptionskip
803      \fi
804      \relax
805      }}

806 \let\captionmargin\captionindent % set to 3pc by AMS class
807 \begingroup\edef\@tempa{\endgroup
808   \noexpand\caption@g@addto@list\noexpand\caption@sty@default
809     {margin=\the\captionmargin
810      \@ifundefined{smf@makecaption}{},{,maxmargin=.1\linewidth}}
811 \@tempa
812 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
813 \DeclareCaptionLabelSeparator{default}{.\enspace}
814 \DeclareDefaultCaptionFont{font}{\@captionfont}
815 \DeclareDefaultCaptionFont{labelfont}{\@captionheadfont}
816 \DeclareDefaultCaptionFont{textfont}{\@captionfont\upshape}
817 \captionsetup[figure]{position=b}
818 \captionsetup[table]{position=t}
819 }

```

### 1.19.3 The beamer class

```

820 \@ifclassloaded{beamer}{%
821   \caption@CheckCommand\beamer@makecaption{%
822     % beamerbaselocalstructure.sty, v 1.53 2007/01/28 20:48:21 tantau
823     \long\def\beamer@makecaption#1#2{%
824       \def\insertcaptionname{\csname#1name\endcsname}%
825       \def\insertcaptionnumber{\csname the#1\endcsname}%
826       \def\insertcaption{#2}%
827       \nobreak\vskip\abovcaptionskip\nobreak
828       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
829       \ifdim \wd\@tempboxa >\hsize
830         \usebeamertemplate**{caption}\par
831       \else
832         \global \@minipagefalse
833         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
834       \fi
835       \nobreak\vskip\belowcaptionskip\nobreak}}
836 \DeclareCaptionLabelFormat{default}{#1}
837 \DeclareDefaultCaptionJustification{\raggedright}
838 \DeclareDefaultCaptionFont{font}{%
839   \usebeamerfont*{caption}%
840   \usebeamerfont*{caption}}
841 \DeclareDefaultCaptionFont{labelfont}{%
842   \usebeamerfont*{caption name}%
843   \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.)

```

844 \defbeamertemplate{caption}{caption3}{%
845   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}

```

```

846 \DeclareOption{beamer}{%
847   % \usebeamertemplate**{caption} will set font
848   \DeclareDefaultCaptionFont{font}{}%
849   \setbeamertemplate{caption}[caption3]}
850 {}

```

#### 1.19.4 The KOMA-Script classes

```

851 \@ifundefined{scr@caption}{}{%
852   \caption@CheckCommand\@makecaption{%
853     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
854     \long\def\@makecaption#1#2{%
855       \if@captionabove
856         \vskip\belowcaptionskip
857       \else
858         \vskip\abovecaptionskip
859       \fi
860       \@makecaption\@firstofone{#1}{#2}%
861       \if@captionabove
862         \vskip\abovecaptionskip
863       \else
864         \vskip\belowcaptionskip
865       \fi}}
866 \DeclareCaptionFormat{default}[#1#2#3\par]{%
867   \ifdofullc@p
868     \caption@ifin@list\caption@lsep@crlist\caption@lsep@name
869     {\caption@Error{%
870       The option 'labelsep=\caption@lsep@name' does not work\MessageBreak
871       with \noexpand\setcaphanging (which is set by default)}}%
872     {\caption@fmt@hang{#1}{#2}{#3}}%
873   \else
874     #1#2%
875     \ifdim\cap@indent<\z@
876       \par
877       \noindent\hspace*{-\cap@indent}%
878     \else\if@capbreak
879       \par
880     \fi\fi
881     #3\par
882   \fi}
883 \DeclareCaptionLabelSeparator{default}{\captionformat}
884 \DeclareDefaultCaptionFont{font}{\scr@fnt@caption}
885 \DeclareDefaultCaptionFont{labelfont}{\scr@fnt@captionlabel}
886 }

```

#### 1.19.5 The NTG Dutch classes

```

887 \@ifundefined{CaptionFonts}{}{%
888   \caption@CheckCommand\@makecaption{%
889     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
890     \long\def\@makecaption#1#2{%
891       \vskip\abovecaptionskip
892       \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
893       \ifdim \wd\@tempboxa >\hsize

```

```

894         {\CaptionLabelFont#1:} \CaptionTextFont#2\par
895     \else
896         \global \@minipagefalse
897         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
898     \fi
899     \vskip\belowcaptionskip}}
900 \DeclareDefaultCaptionFont{labelfont}{\CaptionLabelFont}
901 \DeclareDefaultCaptionFont{textfont}{\CaptionTextFont}
902 }

```

### 1.19.6 The frenchb Babel option

```

903 \@ifundefined{FB@makecaption}{}{%
904     \caption@CheckCommand\@makecaption{%
905         % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
906         \long\def\@makecaption#1#2{%
907             \vskip\abovecaptionskip
908             \sbox\@tempboxa{#1\CaptionSeparator #2}%
909             \ifdim \wd\@tempboxa >\hsize
910                 #1\CaptionSeparator #2\par
911             \else
912                 \global \@minipagefalse
913                 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
914             \fi
915             \vskip\belowcaptionskip}}
916 \ifx\@makecaption\STD@makecaption
917     \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
918     \def\caption@frenchb{% supress frenchb warning
919         \let\STD@makecaption\@makecaption
920         \let\FB@makecaption\@makecaption}
921 \else
922     \ifx\@makecaption\@undefined\else
923         \PackageInfo{caption}{%
924             The definition of \protect\@makecaption\space
925             has been changed,\MessageBreak
926             frenchb will NOT customize it}%
927     \fi
928 \fi
929 }

```

### 1.19.7 The frenchle/pro package

```

930 \@ifundefined{frenchTeXmods}{}{%
931     \caption@CheckCommand\@makecaption{%
932         % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
933         % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
934         \def\@makecaption#1#2{%
935             \ifFTY%
936                 \def\@secondofmany##1##2\void{##2}%
937                 \def\@tempa{\@secondofmany#2\void}%
938                 \ifx\@tempa\empty%
939                     \let\captionseparator\empty%
940                 \fi%

```

```

941     \@mcORI{#1}{\relax\captionfont{#2}}%
942     \else
943     \@mcORI{#1}{#2}%
944     \fi}}
945 \caption@CheckCommand\@makecaption{%
946   % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
947   \def\@makecaption#1#2{%
948     \ifFTY%
949     \def\@secondofmany##1##2\void{##2}%
950     \protected@edef\@tempa{\@secondofmany#2\void}%
951     \ifx\@tempa\empty%
952     \let\captionseparator\empty%
953     \fi%
954     \@mcORI{#1}{\relax\captionfont{#2}}%
955     \else
956     \@mcORI{#1}{#2}%
957     \fi}}
958 \DeclareDefaultCaptionFont{textfont}{\itshape}%
959 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
960 }

```

## 1.20 Execution of options

```

961 \captionsetup{style=default,position=default,list=1,listformat=default,%
962               twoside=\if@twoside 1\else 0\fi}
963 \ProcessOptions*
964 \caption@IfCheckCommand{%
965   \caption@setbool{documentclass}{1}%
966 }{%
967   \caption@setbool{documentclass}{0}%
968   \PackageInfo{caption}{%
969     Unknown document class (or package),\MessageBreak
970     standard defaults will be used}%
971   \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
972 }

```

## 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`.

```

973 \newcommand*\caption@addcontentsline[2]{%
974   \caption@iflist
975   {\def\@tempa{#2}}%
976   {\let\@tempa\@empty}%
977   \ifx\@tempa\@empty \else
978   {\let\\space
979     \addcontentsline{\csname ext@#1\endcsname}{#1}%
980                     {\protect\numberline
981                      {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
982                      {\ignorespaces #2}}}%
983   \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.)
                  984 \newif\ifcaption@star

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

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

\caption@@make   \caption@@make{<caption label>}{<caption text>}
                  989 \newcommand\caption@@make[2]{%
                  990   \begingroup
                  991   \caption@stepcounter
                  992   \caption@beginhook

                  Check margin, if \caption@minmargin or \caption@maxmargin is set
                  993   \ifx\caption@maxmargin\undefined \else
                  994     \ifdim\captionmargin>\caption@maxmargin\relax
                  995       \captionmargin\caption@maxmargin\relax
                  996     \fi
                  997   \fi
                  998   \ifx\caption@minmargin\undefined \else
                  999     \ifdim\captionmargin<\caption@minmargin\relax
                  1000       \captionmargin\caption@minmargin\relax
                  1001     \fi
                  1002   \fi

                  Special single-line treatment (option singlelinecheck=)
                  1003   \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%

                  Typeset the left margin (option margin=)
                  1004   \caption@calcmargin
                  1005   \@tempdima\captionmargin
                  1006   \ifdim\captionmargin@=\z@ \else
                  1007     \caption@ifoddpagel{\advance\@tempdima\captionmargin@}%
                  1008   \fi
                  1009   \caption@ifh{\advance\@tempdima\caption@indent}%
                  1010   \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*.
                  1011   \@tempdima\captionwidth
                  1012   \caption@ifh{\advance\@tempdima-\caption@indent}%
                  1013   \caption@parbox\@tempdima{

```

Typeset the indentation (option `indention=`)

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

```
1014 \caption@ifh{%
1015 \ifdim\caption@indent=\z@
1016 \leavevmode
1017 \else
1018 \hskip-\caption@indent
1019 \fi}%
```

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

```
1020 \caption@@@make{#1}{#2}}%
```

Typeset the right margin (option `margin=`)

```
1021 \@tempdima\captionmargin
1022 \ifdim\captionmargin@=\z@ \else
1023 \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
1024 \fi
1025 \hspace\@tempdima
1026 \caption@endhook
1027 \endgroup
1028 \global\caption@starfalse}
```

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

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

```
1029 \newcommand*\caption@calcmargin{%
1030 \ifdim\captionwidth=\z@
1031 \captionwidth\linewidth
1032 \advance\captionwidth by -2\captionmargin
1033 \advance\captionwidth by -\captionmargin@
1034 \else
1035 \captionmargin\linewidth
1036 \advance\captionmargin by -\captionwidth
1037 \divide\captionmargin by 2
1038 \captionmargin@\z@
1039 \fi

1040 \caption@Debug{%
1041 \string\hsize=\the\hsize,
1042 \string\linewidth=\the\linewidth,\MessageBreak
1043 \string\leftmargin=\the\leftmargin,
1044 \string\rightmargin=\the\rightmargin,\MessageBreak
1045 \string\margin=\the\captionmargin,
1046 \string\margin@=\the\captionmargin@,
1047 \string\width=\the\captionwidth}%
1048 }
```

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

This one does the single-line-check.

```
1049 \newcommand\caption@slc[4]{%
1050 \caption@Debug{Begin SLC}%
1051 \begingroup
1052 \caption@singleline
1053 \let\caption@hj\@empty
```



```

1054 \caption@calcmargin % calculate #3 if necessary
1055 \caption@prepareslc
1056 \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1057 \ifdim\wd\@tempboxa>#3%
1058 \endgroup
1059 \else
1060 \endgroup
1061 \caption@singleline
1062 #4%
1063 \fi
1064 \caption@Debug{End SLC}}

1065 \newcommand*\caption@singleline{%
1066 \caption@xsetup\caption@opt@singleline
1067 \let\caption@fmt\caption@slfmt}

```

`\caption@prepareslc`

`\caption@prepareslc`

**Re-define anything which would disturb the single-line-check.**

```

1068 \newcommand*\caption@prepareslc{%
1069 \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1070 \def\label{\caption@withoptargs\@gobbletwo}%
1071 \let\stepcounter\caption@l@stepcounter
1072 \let\refstepcounter\stepcounter\let\H@refstepcounter\stepcounter}

1073 \newcommand*\caption@l@stepcounter[1]{\advance\csname c@#1\endcsname\@ne\relax}

```

`\caption@parbox`

`\caption@parbox{<width>}{<contents>}`

**This macro defines the box which surrounds the caption paragraph.**

```
1074 \newcommand*\caption@parbox{\parbox[b]}
```

`\caption@@@make`

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

**This one finally typesets the caption paragraph, without margin and indentation.**

```
1075 \newcommand\caption@@@make[2]{%
```

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

```

1076 \sbox\@tempboxa{#1}%
1077 \ifdim\wd\@tempboxa=\z@
1078 \let\caption@lsep\relax
1079 % \@capbreakfalse
1080 \fi

```

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

```

1081 \caption@ifempty{#2}{%
1082 \let\caption@lsep\relax
1083 % \@capbreakfalse
1084 % \let\caption@ifstrut\@secondoftwo
1085 }%

```

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

```
1086 \@setpar{\@par\caption@@par}\caption@@par
```

**Finally typeset the caption.**

```

1087 \caption@hj\captionfont\captionsize\caption@fmt
1088 {\ifcaption@star\else{\captionlabelfont#1}\fi}%
1089 {\ifcaption@star\else{\caption@iflfl\captionlabelfont\caption@lsep}\fi}%

```

```

1090    {{\captiontextfont
1091      \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1092      \nobreak\hskip\z@skip % enable hyphenation
1093      \caption@tfmt{#2}%
1094 %      \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1095      \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1096      \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.

```

1097 \newcommand\caption@ifempty[1]{%
1098   \def\caption@tempa{#1}%
1099   \def\caption@tempb{\ignorespaces}%
1100   \ifx\caption@tempa\caption@tempb
1101     \let\caption@tempa\@empty
1102   \fi
1103   \ifx\caption@tempa\@empty
1104     \let\caption@ifempty\@secondoftwo
1105   \else
1106     \let\caption@ifempty\@gobbletwo
1107   \fi
1108   \caption@ifempty\@unused}

```

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

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

```

1109 \newcommand*\caption@@par{%
1110   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

## 1.23 Types & sub-types

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

```

1111 \newcommand*\DeclareCaptionType{%
1112   \@testopt\@DeclareCaptionType{}%
1113   \@onlypreamble\DeclareCaptionType
1114   \def\@DeclareCaptionType[#1]#2{%
1115     \def\caption@type{#2}%
1116     \caption@Debug{New type `#2'}%
1117     \newcounter{#2}\@namedef{theH#2}{}%
1118     \KV@caption@DCT@within{\@ifundefined{c@chapter}{none}{chapter}}%
1119     \KV@caption@DCT@placement{tbp}%
1120     \@ifundefined{c@float@type}%
1121       {\newcounter{float@type}%
1122        \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1123     {}%
1124     \caption@Debug{float type `#2'=\the\value{float@type}}%
1125     \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1126     \addtocounter{float@type}{\value{float@type}}%
1127     \KV@caption@DCT@fileext{lo#2}%
1128     \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%

```

```

1129 \newenvironment{#2}{\@float{#2}}{\end@float}%
1130 \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1131 \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1132 \@ifundefined{l@figure}%
1133   {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1134   {\expandafter\let\csname l@#2\endcsname\l@figure}%
1135 \expandafter\newcommand\csname #2name\endcsname{}%
1136 \edef\@tempa{%
1137   \def\noexpand\@tempa{\expandafter\noexpand\@car#2\@nil}}%
1138 \uppercase\expandafter{\@tempa}%
1139 \edef\@tempb{%
1140   \noexpand@g@addto@macro\noexpand\@tempa{\expandafter\@cdr#2\@nil}}%
1141 \@tempb
1142 \expandafter\let\csname #2name\endcsname\@tempa
1143 \expandafter\newcommand\csname list#2name\endcsname{}%
1144 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1145 \@cons\caption@typelist{#2}%
1146 \caption@setkeys[caption]{caption@DCT}{#1}%
1147 \@ifnextchar[\@@DeclareCaptionType\relax}
1148 \@onlypreamble\@@DeclareCaptionType
1149 \def\@@DeclareCaptionType[#1]{%
1150   \KV@caption@DCT@name{#1}%
1151   \@ifnextchar[\@@@DeclareCaptionType\relax}
1152 \@onlypreamble\@@@DeclareCaptionType
1153 \def\@@@DeclareCaptionType[#1]{%
1154   \KV@caption@DCT@listname{#1}}
1155 \@onlypreamble\@@@DeclareCaptionType
1156 \let\DeclareFloatingEnvironment\DeclareCaptionType % old command name

```

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

```

1157 \newcommand*\caption@listof[1]{%
1158   \begingroup
1159     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1160     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1161     \let\caption@ORI@starttoc\@starttoc
1162     \renewcommand*\@starttoc[1]{%
1163       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1164     \listoffigures
1165   \endgroup}

```

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

```
1166 \newcommand*\caption@typelist{}
```

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

```

1167 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
1168 \@onlypreamble@key{caption@DCT}{fileext}
1169 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1170 \@onlypreamble@key{caption@DCT}{listname}
1171 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1172 \@onlypreamble@key{caption@DCT}{name}
1173 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}

```

```

1174 \@onlypreamble@key{caption@DCT}{placement}
1175 \define@key{caption@DCT}{within}{%
1176   \@ifundefined{c@chapter}{}{\@removefromreset\caption@type{chapter}}%
1177   \@removefromreset\caption@type{section}%
1178   \begingroup
1179     \caption@setkeys[caption]{caption@within}{#1}%
1180   \endgroup}
1181 \@onlypreamble@key{caption@DCT}{within}
1182 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1183 \@onlypreamble@key{caption@DCT}{without}

1184 \define@key{caption@within}{none}[]{%
1185   \caption@within{}{}}
1186 \@onlypreamble@key{caption@within}{none}
1187 \@ifundefined{c@chapter}{}{%
1188   \define@key{caption@within}{chapter}[]{%
1189     \@addtoreset\caption@type{chapter}%
1190     \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
1191   \@onlypreamble@key{caption@within}{chapter}}
1192 \define@key{caption@within}{section}[]{%
1193   \@addtoreset\caption@type{section}%
1194   \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
1195 \@onlypreamble@key{caption@within}{section}

```

\caption@within \caption@within{*the code*}{*the Hcode*}

```

1196 \newcommand*\caption@within{%
1197   \expandafter\caption@within@\expandafter{\caption@type}}
1198 \@onlypreamble\caption@within
1199 \newcommand*\caption@within@[3]{%
1200   \global\@namedef{the\caption@type}{#2\arabic{#1}}%
1201   \@ifundefined{theH\caption@type}\caption@AtBeginDocument\@firstofone
1202     {\global\@namedef{theH\caption@type}{#3\arabic{#1}}}}
1203 \@onlypreamble\caption@within@

```

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

```

1204 \providecommand*\@removefromreset[2]{%
1205   \expandafter\let\csname c@#1\endcsname\@removefromreset
1206   \def\@elt##1{%
1207     \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1208     \else
1209       \noexpand\@elt{##1}%
1210     \fi}%
1211   \expandafter\xdef\csname cl@#2\endcsname{%
1212     \csname cl@#2\endcsname}}

```

\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*.

```

1213 \newcommand*\DeclareCaptionSubType{%
1214   \caption@teststar\@DeclareCaptionSubType\@firstoftwo\@secondoftwo}
1215 \@onlypreamble\DeclareCaptionSubType

```

```

1216 \newcommand*\@DeclareCaptionSubType[1]{%
1217   \@testopt{\@@DeclareCaptionSubType{#1}}{alph}}
1218 \@onlypreamble\@DeclareCaptionSubType
1219 \def\@@DeclareCaptionSubType#1[#2]#3{%
1220   \@ifundefined{c@#3}%
1221     {\caption@Error{No float type '#3' defined}}%
1222     {\@ifundefined{c@sub#3}%
1223       {\caption@Debug{New subtype 'sub#3'}%
1224         \newcounter{sub#3}%
1225         \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1226         \@ifundefined{l@chapter}%
1227           {\edef\@tempa{\expandafter\expandafter\expandafter\noexpand
1228             \expandafter\@car\l@subsubsection\@nil}%
1229             \def\@tempb{\@dottedtocline}%
1230             \ifx\@tempa\@tempb % \l@subsubsection starts with \@dottedtocline
1231               \expandafter\edef\csname l@sub#3\endcsname{%
1232                 \noexpand\@dottedtocline{2}%
1233                 \expandafter\expandafter\expandafter\noexpand
1234                 \expandafter\@gobbletwo\l@subsubsection}%
1235             \else
1236               \@namedef{l@sub#3}{\@dottedtocline{2}{3.8em}{3.2em}}%
1237               \fi}%
1238           {\expandafter\let\csname l@sub#3\endcsname\l@subsection}%
1239           \@cons\caption@subtypelist{{#3}}}%
1240     {\caption@Debug{Modify caption 'sub#3'}}%
1241   #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
1242   {\@namedef{p@sub#3}}}%
1243   \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1244   {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1245     \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1246   \@namedef{theHsub#3}{\csname theH#3\endcsname.\@arabic{sub#3}}%
1247   \@namedef{sub#3autorefname}{\csname #3name\endcsname}}
1248 \@onlypreamble\@@DeclareCaptionSubType

```

```

\caption@For \caption@For{\langle elt-list \rangle}{\langle code with #1 \rangle}
\caption@For*{\langle elt-list \rangle}{\langle code with #1 \rangle}

1250 \newcommand*\caption@For{\caption@withoptargs\caption@@For}
1251 \@onlypreamble\caption@For

1252 \newcommand\caption@@For[3]{%
1253   \caption@AtBeginDocument#1{%
1254     \def\@elt##1{#3}%
1255     \@nameuse{caption@#2}%
1256     \let\@elt\relax}%
1257 \@onlypreamble\caption@@For

```

```

1258 \caption@AtBeginDocument{%
1259   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1260   \ifx\subref\@tempa
1261     \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1262     \let\caption@setfloattype\@gobble
1263     \let\@dottedxxxline\sf@NEW@dottedxxxline
1264     \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.

```

1265   \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1266   \fi
1267   \let\sf@NEW@dottedxxxline\@undefined
1268   \let\sf@NEW@subfloat\@undefined}
1269 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1270   \begingroup
1271     \caption@setfloattype{#1}%
1272     \caption@setoptions{subfloat}%
1273     \caption@setoptions{sub#1}%
1274     \ifnum #3>\@nameuse{c@#2depth}\else
1275       \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1276     \fi
1277   \endgroup}
1278 \def\sf@NEW@subfloat{%
1279   \begingroup
1280     \caption@setfloattype\@capttype
1281     \sf@ifpositiontop{%
1282       \maincaptiontoptrue
1283     }{%
1284       \maincaptiontopfalse
1285     }%
1286     \caption@setoptions{subfloat}%
1287     \caption@setoptions{sub\@capttype}%
1288     \let\sf@oldlabel=\label
1289     \let\label=\subfloat@label
1290     \ifmaincaptiontop\else
1291       \advance\@nameuse{c@\@capttype}\@ne
1292     \fi
1293     \refstepcounter{sub\@capttype}%
1294     \setcounter{sub\@capttype}{\value{sub\@capttype}}%
1295     \@ifnextchar [% %] match left bracket
1296       {\sf@@subfloat}%
1297       {\sf@@subfloat[\@empty]}}

```

## 2 Main package

### 2.1 Identification

```
1298 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1299 \ProvidesPackage{caption}[2007/12/23 v3.1f Customizing captions (AR)]
1300 %\@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.

1301 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1302 \@onlypreamble\caption@Info
```

### 2.2 Loading the kernel

```
1303 \RequirePackage{caption3}[2007/12/21] % needs v3.1f or newer
```

### 2.3 Check against incompatible packages

```
1304 \@ifpackageloaded{caption2}{%
1305   \caption@Error{%
1306     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1307     the (current) caption package}%
1308   \endinput
1309 }{}

1310 \caption@AtBeginDocument{%
1311   \@ifpackageloaded{ftcap}{%
1312     \caption@IncompatiblePackage{ftcap}{}%
1313   }{}%
1314   \@ifpackageloaded{nonfloat}{%
1315     \caption@IncompatiblePackage{nonfloat}{.\MessageBreak
1316       Furthermore you could use \noexpand\captionof instead of \MessageBreak
1317       \noexpand\figcaption or \string\tabcaption}%
1318   }{}%
1319   \@ifpackageloaded{topcapt}{%
1320     \caption@IncompatiblePackage{topcapt}{}%
1321   }{}%
1322 }

\caption@Incompatible Issue an 'incompatible' warning.

1323 \newcommand*\caption@IncompatiblePackage[2]{%
1324   \PackageWarningNoLine{caption}{%
1325     Usage of the #1 package together with the \MessageBreak
1326     caption package is strongly not recommended.\MessageBreak
1327     Instead of loading the #1 package you should \MessageBreak
1328     use the caption package option \MessageBreak
1329     \space\space `tableposition=top' #2}}

1330 \@onlypreamble\caption@IncompatiblePackage
```

### 2.4 Check document class

```
1331 \caption@ifbool{documentclass}{}{}%
1332   \PackageWarningNoLine{caption}{%
1333     Unsupported document class (or package) detected, \MessageBreak
```

```

1334     usage of the caption package is not recommended}%
1335 \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1336 }

```

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

```

1337 \@ifundefined{@captionheadfont}{}{}%
1338 \caption@Info{AMS or SMF document class}%
1339 \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1340 }

```

## 2.6 Emulation of the KOMA-Script commands

```

1341 \@ifundefined{scr@caption}{}{}%
1342 \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 1343 \def\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1344 \def\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1345 \if@tablecaptionabove
1346 \caption@AtBeginDocument{%
1347 \def\caption@tempa#1{%
1348 \ifpackageloaded{#1}{\caption@Error{%
1349 You can't use the global option\MessageBreak
1350 \space\space'tablecaptionabove'\MessageBreak
1351 and use the #1 package, too}}}%
1352 \caption@tempa{ftcap}%
1353 \caption@tempa{nonfloat}%
1354 \caption@tempa{topcapt}}%
1355 \@tablecaptionabovetrue
1356 \else
1357 \@tablecaptionabovefalse
1358 \fi

\onelinecaptionstrue
\onelinecaptionfalse 1359 \def\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1360 \def\onelinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1361 \ifonelinecaptions
1362 \onelinecaptionstrue
1363 \else
1364 \onelinecaptionfalse
1365 \fi

\@captionabovetrue
\@captionabovefalse 1366 \def\@captionabovetrue{\let\caption@position\@firstoftwo}
1367 \def\@captionabovefalse{\let\caption@position\@secondoftwo}

\setcapindent 1368 \let\caption@KOMA@setcapindent\@setcapindent
1369 \renewcommand*\@setcapindent[1]{%
1370 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

```



```

1371 \let\caption@KOMA@@setcapindent\@@setcapindent
1372 \renewcommand*\@@setcapindent[1]{%
1373   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1374 \newcommand*\caption@setcapindent{%
1375   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1376 \@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.)
1377 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1378   \csname\string\setcapwidth\endcsname
1379 \@namedef{\string\setcapwidth}[1]#2{%
1380   \caption@KOMA@setcapwidth{#1}{#2}\caption@setcapwidth{#1}}
1381 \newcommand*\caption@setcapwidth[1]{%
1382   \ifx\#1\\\else
1383     \@ifundefined{cap@margin}{}{%
1384       \def\@tempa{captionbeside}%
1385       \ifx\@tempa\@currenvir\else\caption@Warning{%
1386         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1387       \fi}%
1388   \fi
1389   \captionsetup{width=\cap@width}}

1390 \def\caption@tempa{\hsize}%
1391 \ifx\caption@tempa\cap@width \else
1392   \caption@setcapwidth{?}
1393 \fi

\setcapmargin
1394 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1395   \csname\string\@setcapmargin\endcsname
1396 \@namedef{\string\@setcapmargin}[1]#2{%
1397   \caption@KOMA@setcapmargin{#1}{#2}\caption@setcapmargin}
1398 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1399   \csname\string\@@setcapmargin\endcsname
1400 \@namedef{\string\@@setcapmargin}[1]#2{%
1401   \caption@KOMA@@setcapmargin{#1}{#2}\caption@setcapmargin}
1402 \newcommand*\caption@setcapmargin{%
1403   \begingroup
1404     \let\onelinecaptionsfalse\relax
1405     \def\@twoside{0}%
1406     \def\if@twoside{\def\@twoside{1}\iffalse}%
1407     \cap@margin
1408     \def\@tempa{\endgroup}%
1409     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1410       \def\hspace##1##{\@firstofone}%
1411       \edef\@tempa{\endgroup
1412         \noexpand\captionsetup{%
1413           twoside=\@twoside,singlelinecheck=0,%
1414           margin={\cap@left,\cap@right}}}%
1415     \fi\fi
1416   \@tempa}

```

```

1417 \ifx\cap@margin\relax \else
1418 \caption@setcapmargin
1419 \fi
1420 }

```

## 2.7 Declaration of options

### 2.7.1 Options for figure and table

```

1421 \DeclareCaptionOption{figureposition}{%
1422 \captionsetup*[figure]{position=#1}}
1423 \@onlypreamble@key{caption}{figureposition}

1424 \DeclareCaptionOption{tableposition}{%
1425 \caption@AtBeginDocument{%
1426 \def\caption@tempa##1{%
1427 \ifpackageloaded{##1}{\caption@Error{%
1428 You can't use the caption package option\MessageBreak
1429 \space\space`tableposition=#1'\MessageBreak
1430 and use the ##1 package, too)}}}%
1431 \caption@tempa{ftcap}%
1432 \caption@tempa{nonfloat}%
1433 \caption@tempa{topcapt}}%
1434 \captionsetup*[table]{position=#1}}
1435 \@onlypreamble@key{caption}{tableposition}

1436 \DeclareCaptionOption{figurename}{\caption@SetFloatName{figure}{#1}}
1437 \DeclareCaptionOption{tablename}{\caption@SetFloatName{table}{#1}}
1438 \DeclareCaptionOption{name}{\caption@setfloatname\@captive{#1}}

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

1439 \newcommand*\caption@SetFloatName[2]{%
1440 \begingroup
1441 \def\@tempa{\endgroup\@firstofone}%
1442 \ifundefined{language\@tempa}{%
1443 \ifundefined{captions\language\@tempa}{%
1444 \def\@tempa{\endgroup
1445 \expandafter\addto\csname captions\language\@tempa\endcsname}}%
1446 \@tempa{\caption@setfloatname{#1}{#2}}}%
1447 \newcommand*\caption@setfloatname[2]{\@namedef{#1name}{#2}}
1448 \caption@AtBeginDocument{\let\caption@SetFloatName\caption@setfloatname}

1449 \DeclareCaptionOption{figurewithin}{\caption@Within{figure}{#1}}
1450 \@onlypreamble@key{caption}{figurewithin}
1451 \DeclareCaptionOption{figurewithout}{\caption@Within{figure}{none}}
1452 \@onlypreamble@key{caption}{figurewithout}

1453 \DeclareCaptionOption{tablewithin}{\caption@Within{table}{#1}}
1454 \@onlypreamble@key{caption}{tablewithin}
1455 \DeclareCaptionOption{tablewithout}{\caption@Within{table}{none}}
1456 \@onlypreamble@key{caption}{tablewithout}

\caption@within

1457 \newcommand*\caption@Within[1]{\def\caption@type{#1}\KV@caption@DCT@within}
1458 \@onlypreamble\caption@Within

```

## 2.7.2 Miscellaneous options

```
1459 \DeclareCaptionOption*{config}[caption]{%
1460   \InputIfFileExists{#1.cfg}%
1461   {\typeout{*** Local configuration file #1.cfg used ***}}%
1462   {\caption@Warning{Configuration file #1.cfg not found}}}%

1463 \DeclareCaptionOption{@minipage}{%
1464   \caption@ifinlist{#1}{auto,default}%
1465   {\let\caption@if@minipage\@gobbletwo}%
1466   {\caption@set@bool\caption@if@minipage{#1}}}%
1467 \captionsetup{@minipage=default}
```

## 2.7.3 caption v1.x compatibility options

```
1468 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1469 \@onlypreamble@key{caption}{compatibility}

1470 \DeclareCaptionOptionNoValue*{normal}{%
1471   \caption@setformat{plain}%
1472   \caption@setjustification{justified}}%
1473 \DeclareCaptionOptionNoValue*{isu}{%
1474   \caption@setformat{hang}%
1475   \caption@setjustification{justified}}%
1476 \DeclareCaptionOptionNoValue*{hang}{%
1477   \caption@setformat{hang}%
1478   \caption@setjustification{justified}}%
1479 \DeclareCaptionOptionNoValue*{center}{%
1480   \caption@setformat{plain}%
1481   \caption@setjustification{centering}}%
1482 \DeclareCaptionOptionNoValue*{anne}{%
1483   \caption@setformat{plain}%
1484   \caption@setjustification{centerlast}}%
1485 \DeclareCaptionOptionNoValue*{centerlast}{%
1486   \caption@setformat{plain}%
1487   \caption@setjustification{centerlast}}%

1488 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1489 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1490 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1491 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1492 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1493 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1494 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1495 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1496 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1497 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1498 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1499 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1500 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1501 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1502 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1503 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}

1504 \caption@setbool{ruled}{0}
1505 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}
```

## 2.7.4 caption2 v2.x compatibility options

```
1506 \DeclareCaptionOptionNoValue*{flushleft}{%
1507   \caption@setformat{plain}%
1508   \caption@setjustification{raggedright}}
1509 \DeclareCaptionOptionNoValue*{flushright}{%
1510   \caption@setformat{plain}%
1511   \caption@setjustification{raggedleft}}
1512 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}
1513 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
1514   \caption@WarningNoLine{Obsolete option 'ignoreLTcapwidth' ignored}}
```

## 2.7.5 Obsolete caption v3.0 options

```
1515 \DeclareCaptionOption*{caption}{%
1516   \caption@setbool{temp}{#1}%
1517   \caption@ifbool{temp}{}{%
1518     \caption@Error{%
1519       The package option 'caption=#1' is obsolete.\MessageBreak
1520       Please pass this option to the subfig package instead.\MessageBreak
1521       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.

```
1522 \DeclareCaptionOption{FPlist}[1]{\caption@setFPOption{list}{#1}}
1523 \DeclareCaptionOption{FPref}[1]{\caption@setFPOption{ref}{#1}}
1524 \@onlypreamble@key{caption}{FPlist}
1525 \@onlypreamble@key{caption}{FPref}
1526 \newcommand*\caption@setFPOption[2]{%
1527   \edef\@tempa{\@car#2\@nil}%
1528   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1529 \@onlypreamble\caption@setFPOption
1530 \captionsetup{FPlist=caption,FPref=figure}
```

## 2.7.7 hyperref package support options

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

```
1531 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1532 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}
1533 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

## 2.8 Processing of options

```
1534 \caption@ProcessOptions*{caption}
```

## 2.9 \captionof and \ContinuedFloat

```
1535 \caption@AtBeginDocument{%
1536   \DeclareCaptionOption{type}{\caption@settype{#1}}%
1537   \DeclareCaptionOption{type*}{\caption@settype*{#1}}%
```

```

1538 \DeclareCaptionOption{subtype}[sub\@capttype]{\caption@setsubtype{#1}}%
1539 \DeclareCaptionOption{subtype*}[sub\@capttype]{\caption@setsubtype*{#1}}%
1540 }

```

*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).

```

1541 \newcommand*\caption@settype{%
1542   \caption@@settype{}}

1543 \newcommand*\caption@setsubtype{%
1544   \caption@iftype
1545     {\caption@@settype{sub}}%
1546     {\caption@Error{Option 'subtype=' outside float}}}%

1547 \newcommand*\caption@@settype[1]{%
1548   \caption@teststar{\caption@@@settype{#1}}{\@firstoftwo\@secondoftwo}

1549 \newcommand*\caption@@@settype[3]{%
1550   % #1 = "" or "sub"
1551   % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1552   % #3 = <type>, e.g. "figure" or "table"
1553   \ifundefined{c@#3}%
1554     {\caption@Error{No float type '#3' defined}}%
1555     {\caption@Debug{#1type=#3}}%
1556     \caption@checkgrouplevel{#1}{%
1557       \captionsetup{#1type#2*\@empty=...}#2{ or
1558         \@backslashchar#1captionof}{}}%

1559   \edef\@tempa{#3}%
1560   \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1561     \@nameuse{caption@#1type@warning}%
1562   \fi
1563   \expandafter\let\csname @#1capttype\endcsname\@tempa
1564   \@nameuse{caption@#1typehook}%

1565   \caption@setoptions{#3}%
1566   \ifx\caption@opt\relax
1567     \@nameundef{caption@#1type@warning}%
1568   \else
1569     \@namedef{caption@#1type@warning}{\caption@Warning{%
1570       The #1caption type was already set to
1571       '\csname @#1capttype\endcsname'\MessageBreak}}%
1572   \fi

1573   \let\caption@ifrefstepcounter\@secondoftwo
1574   #2{}{%
1575     \let\@currentlabel\caption@undefinedlabel

```

```

1576 %      \let\@currentHlabel\@undefined
1577      \ifx\caption@ORI@label\@undefined
1578          \let\caption@ORI@label\label
1579          \let\label\caption@xlabel
1580      \fi
1581      \caption@start}}

```

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

```

1582 \newcommand*\caption@typehook{}

```

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

```

1583 \newcommand*\caption@iftype{%
1584   \@ifundefined{\capytype}{\let\@capytype\@undefined\@secondoftwo}\@firstoftwo}

```

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

```

1585 \begingroup\expandafter\expandafter\expandafter\endgroup
1586 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1587   \caption@Debug{TeX engine: TeX}
1588   \let\caption@checkgrouplevel\@gobbletwo
1589 \else
1590   \caption@Debug{TeX engine: e-TeX}
1591   \newcommand*\caption@checkgrouplevel[2]{%
1592     \@ifundefined{#1caption@grouplevel}{%
1593       \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}{}%
1594       \ifnum\currentgrouplevel>\caption@grouplevel\relax
1595         \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1596           \the\currentgrouplevel}%
1597       \else
1598         \caption@Warning{\string#2\MessageBreak outside box or environment}%
1599       \fi
1600     }{} }
1601 \fi

```

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

```

1602 \newcommand*\caption@undefinedlabel{??}

```

`\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.

```

1603 \newcommand*\caption@xlabel{%
1604   \ifx\@currentlabel\caption@undefinedlabel
1605     \caption@Warning{\noexpand\label before \noexpand\caption ignored}%
1606     \expandafter\@gobble
1607   \else
1608     \expandafter\caption@ORI@label
1609   \fi}

```

```

\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*)

```
1610 \caption@AtBeginDocument{%
1611   \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}
1612 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}
```

```
\captionlistentry \captionlistentry[<float type>]{<list entry>}
1613 \newcommand*\captionlistentry{%
1614   \@testopt\caption@listentry\@capttype}

1615 \def\caption@listentry[#1]#2{%
1616   \@bsphack
1617   \caption@refstepcounter{#1}%
1618   \caption@makecurrent{#1}{#2}%
1619   \caption@addcontentsline{#1}{#2}%
1620   \@esphack}
```

```
\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 optional argument.

```
1621 \def\ContinuedFloat{%
1622   \@ifnextchar[\@ContinuedFloat{%
1623     \caption@iftype
1624     {\addtocounter\@capttype\m@ne
1625       \caption@ContinuedFloat\@capttype}%
1626     {\caption@Error{\noexpand\ContinuedFloat outside float}}}}

1627 \def\caption@ContinuedFloat#1{%
1628   \edef\@tempa{#1}%
1629   \ifx\@tempa\caption@CFtype \else
1630     \caption@Error{Continued `#1' after `\caption@CFtype'}%
1631   \fi
1632   \stepcounter{ContinuedFloat}%
1633   \caption@@ContinuedFloat{#1}%
1634   \caption@setoptions{ContinuedFloat}%
1635   \caption@setoptions{continued#1}%
1636   \let\caption@resetContinuedFloat\@gobble}

1637 \def\@ContinuedFloat[#1]{\addtocounter{#1}\m@ne}

1638 \newcounter{ContinuedFloat}
1639 \newcommand*\caption@CFtype{??}
```

```
\caption@@ContinuedFloat \caption@@ContinuedFloat{<type>}
```

Usually this do nothing but this changes if the `hyperref` package is loaded. (See `hyperref` package support for details.)

```
1640 \newcommand*\caption@@ContinuedFloat[1]{}%
```

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

```

1641 \newcommand*\caption@resetContinuedFloat[1]{%
1642   \@stpelt{ContinuedFloat}\xdef\caption@CFtype{#1}}

```

## 2.10 Internal helpers

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

```

1643 \newcommand*\caption@refstepcounter[1]{%
1644   \caption@resetContinuedFloat{#1}%
1645   \caption@@refstepcounter{#1}%
1646   \let\caption@ifrefstepcounter\@firstoftwo}

1647 \newcommand*\caption@@refstepcounter{\refstepcounter}
1648 \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[]{}).`

```

1649 \@ifundefined{kernel@ifnextchar}%
1650   {\newcommand\caption@dblarg[1]{\@ifnextchar[{\#1}{\caption@xdblarg{#1}}}}%
1651   {\newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}}%
1652 \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`)

```

1653 \newcommand*\caption@begin[1]{%
1654   \begingroup
1655     \caption@setfnum{#1}%
1656     \caption@fixposition
1657     \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>`.

```

1658 \newcommand\caption@beginex[3]{%
1659   \caption@begin{#1}%
1660   \caption@addcontentsline{#1}{#2}%
1661   \caption@ifempty{#3}{} }

```



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

```
1662 \newcommand*\caption@end{%
1663   \endgroup
1664   \let\caption@position\caption@fixedposition}
```

`\caption@setfnum` `\caption@setfnum{<type>}`

redefines `\fnum{<type>}` according to the caption label format set with `labelformat=`.  
But if `labelformat=default` is set, `\fnum{<type>}` will not be overwritten by us.

```
1665 \newcommand*\caption@setfnum[1]{%
1666   \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1667   \@namedef{fnum@#1}{\caption@fnum{#1}}%
1668   \fi}
```

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

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

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

```
1669 \newcommand*\caption@boxrestore{%
1670   \let\if@nobreak\iffalse
1671   \let\if@noskipsec\iffalse
1672   \let\par\@par
1673   \let\-\@dischyph
1674   \let'\@acci\let'\@accii\let\=\@acciii
1675   \parindent\z@ \parskip\z@skip
1676   \everypar{}%
1677   \linewidth\hsize
1678   \@totalleftmargin\z@
1679   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1680   \parfillskip\@flushglue \lineskip\normallineskip
1681   \baselineskip\normalbaselineskip
1682   \sloppy
1683   \let\\\@normalcr
1684 }
```

`\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).

```
1685 \newcommand*\caption@normalsize{%
1686   \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.

```
1687 \let\caption@setfloatcapt\@firstofone
```

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

```
\caption@@anchor 1688 \newcommand*\caption@makecurrent[2]{%
\caption@start 1689 \let\caption@@anchor\@firstofone
\caption@@start 1690 \let\caption@start\relax
\caption@freezeHref 1691 \let\caption@@start\relax
\caption@defrostHref 1692 \let\caption@freezeHref\relax
1693 \let\caption@defrostHref\relax
```

## 2.11 `\caption`, `\@caption`, and `\@makecaption`

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

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

```
1694 \caption@AtBeginDocument{%
1695   \caption@setbool{incompatible}{0}%
1696   \caption@CheckCommand\caption{%
1697     % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1698     \def\caption{%
1699       \ifx\@capttype\@undefined
1700         \@latex@error{\noexpand\caption outside float}\@ehd
1701         \expandafter\@gobble
1702       \else
1703         \refstepcounter\@capttype
1704         \expandafter\@firstofone
1705       \fi
1706       {\@dblarg{\@caption\@capttype}}%
1707     }%
1708   \caption@CheckCommand\caption{%
1709     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1710     \def\caption{
1711       \ifx\@capttype\@undefined
1712         \@latex@error{\noexpand\caption outside figure or table}\@ehd
1713         \expandafter\@gobble
1714       \else
1715         \refstepcounter\@capttype
1716         \expandafter\@firstofone
1717       \fi
```

```

1718     {\@dblarg{\@caption\@capttype}}%
1719   }}%

1720 \caption@CheckCommand\caption{%
1721   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1722   \renewcommand\caption{%
1723     \ifx\@capttype\@undefined
1724       \@latex@error{\noexpand\caption outside float}\@ehd
1725       \expandafter\@gobble
1726     \else
1727       \refstepcounter\@capttype
1728       \let\@tempf\@caption
1729       \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1730         \expandafter\expandafter\let
1731         \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1732       \fi
1733     \fi
1734     \@dblarg{\@tempf\@capttype}}}%

1735 \caption@CheckCommand\caption{%
1736   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1737   % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1738   % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1739   \def\caption{%
1740     \ifx\@capttype\@undefined
1741       \@latex@error{\noexpand\caption outside float}\@ehd
1742       \expandafter\@gobble
1743     \else
1744       \H@refstepcounter\@capttype
1745       \@ifundefined{fst@\@capttype}{%
1746         \let\Hy@tempa\@caption
1747       }{%
1748         \let\Hy@tempa\Hy@float@caption
1749       }%
1750       \expandafter\@firstofone
1751     \fi
1752     {\@dblarg{\Hy@tempa\@capttype}}}%
1753   }}%

1754 \caption@CheckCommand\caption{%
1755   % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1756   \def\caption{%
1757     \ifx\@capttype\@undefined
1758       \@latex@error{\noexpand\caption outside float}\@ehd
1759       \expandafter\@gobble
1760     \else
1761       \H@refstepcounter\@capttype
1762       \let\Hy@tempa\@caption
1763       \@ifundefined{float@caption}{%
1764       }{%
1765         \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
1766         \let\Hy@tempa\Hy@float@caption
1767       \fi
1768       }%
1769       \expandafter\@firstofone
1770     \fi

```

```

1771     {\@dblarg{\Hy@tempa\@capttype}}%
1772   }}%

1773 \caption@ifcheckcommand{}{%
1774   \caption@info{%
1775     Incompatible package detected (regarding \string\caption).\MessageBreak
1776     \string\caption\space=\space\meaning\caption}%
1777   \caption@setbool{incompatible}{1}}%

1778 \caption@checkcommand\@caption{%
1779   % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1780   \long\def\@caption#1[#2]#3{%
1781     \par
1782     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1783     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1784     \begingroup
1785       \@parboxrestore
1786       \if@minipage
1787         \@setminipage
1788       \fi
1789       \normalsize
1790       \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1791     \endgroup}}%

1792 \caption@checkcommand\@caption{%
1793   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1794   \long\def\@caption#1[#2]#3{% second argument ignored
1795     \par\nobreak
1796     \begingroup
1797       \@parboxrestore
1798       \if@minipage
1799         \@setminipage
1800       \fi
1801       \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
1802     \endgroup}}%

1803 % \caption@checkcommand\float@caption{%
1804 %   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1805 %   \long\def\float@caption#1[#2]#3{%
1806 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%
1807 %     {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
1808 %     \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
1809 %     \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
1810 %     \@ifnextchar[{\float@ccon}{\egroup}}%
1811 %   \long\def\float@ccon[#1]{#1\par\egroup}}%

1812 \caption@checkcommand\@caption{%
1813   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1814   \long\def\@caption#1[#2]#3{%
1815     \hyper@makecurrent{\@capttype}%
1816     \def\@currentlabelname{#2}%
1817     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1818       \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1819     }%
1820     \begingroup
1821       \@parboxrestore
1822       \if@minipage

```

```

1823         \@setminipage
1824     \fi
1825     \normalsize
1826     \@makecaption{\csname fnum@#1\endcsname}{%
1827         \ignorespaces
1828         \ifHy@nesting
1829             \hyper@@anchor{\@currentHref}{#3}%
1830         \else
1831             \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1832         \fi
1833     }%
1834     \par
1835     \endgroup
1836 }}%

1837 \caption@CheckCommand\@caption{%
1838 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1839 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1840 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1841 \long\def\@caption#1[#2]#3{%
1842     \expandafter\ifx\csname if@capstart\expandafter\endcsname
1843         \csname iftrue\endcsname
1844         \global\let\@currentHref\hc@currentHref
1845     \else
1846         \hyper@makecurrent{\@cuptype}%
1847     \fi
1848     \def\@currentlabelname{#2}%
1849     \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1850         \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1851     }%
1852     \begingroup
1853         \@parboxrestore
1854         \if@minipage
1855             \@setminipage
1856         \fi
1857         \normalsize
1858         \expandafter\ifx\csname if@capstart\expandafter\endcsname
1859             \csname iftrue\endcsname
1860             \global\@capstartfalse
1861             \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1862         \else
1863             \@makecaption{\csname fnum@#1\endcsname}{%
1864                 \ignorespaces
1865                 \ifHy@nesting
1866                     \hyper@@anchor{\@currentHref}{#3}%
1867                 \else
1868                     \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1869                 \fi
1870             }%
1871         \fi
1872         \par
1873     \endgroup
1874 }}%

1875 \caption@CheckCommand\@caption{%

```

```

1876 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1877 \long\def\@caption#1[#2]{%
1878   \def\@currentlabelname{#2}%
1879   \NR\@caption{#1}[{#2}]%
1880 }%
1881 \caption@CheckCommand\@caption{%
1882   % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1883   \long\def\@caption#1[#2]#3{%
1884     \ifundefined{if#1topcap}%
1885       {\subfig@oldcaption{#1}[{#2}]{#3}}%
1886       {\@nameuse{if#1topcap}%
1887         \@listsubcaptions{#1}%
1888         \subfig@oldcaption{#1}[{#2}]{#3}%
1889       }%
1890       \subfig@oldcaption{#1}[{#2}]{#3}%
1891       \@listsubcaptions{#1}%
1892     \fi}}%
1893 \caption@CheckCommand\@caption{%
1894   % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1895   \def\@caption{\caption}%
1896 % \long\def\@caption#1[#2]#3{%
1897 %   \ifundefined{caption@setfloattype}%
1898 %     \caption@settype
1899 %     \caption@setfloattype
1900 %     \@captype
1901 %     \sf@ifpositiontop{%
1902 %       \@listsubcaptions{#1}%
1903 %       \sf@old@caption{#1}[{#2}]{#3}%
1904 %     }%
1905 %     \sf@old@caption{#1}[{#2}]{#3}%
1906 %     \@listsubcaptions{#1}%
1907 %   }%
1908 % }%
1909 \caption@ifCheckCommand{}{%
1910   \caption@Info{%
1911     Incompatible package detected (regarding \string\@caption).\MessageBreak
1912     \string\@caption\space=\space meaning\@caption}%
1913   \caption@setbool{incompatible}{1}}%

```

The option `compatibility=` will override the compatibility mode.

```

1914 \ifundefined{caption@ifcompatibility}%
1915 {\let\caption@ifcompatibility\caption@ifincompatible
1916  \let\@tempa\caption@WarningNoLine}%
1917 {\let\@tempa\@gobble}% suppress warning
1918 \caption@ifcompatibility{%
1919   \@tempa{%
1920     \noexpand\caption will not be redefined since it's already\MessageBreak
1921     redefined by a document class or package which is\MessageBreak
1922     unknown to the caption package}%
1923   \renewcommand*\caption@ContinuedFloat[1]{%
1924     \caption@Error{Not available in compatibility mode}}%

```

`\ContinuedFloat` is not supported in compatibility mode.

```

\caption@start is not supported in compatibility mode.
1925 \caption@AtBeginDocument*{%
1926 \let\caption@start\relax
1927 \ifundefined{caption@ORI@capstart}}{%
1928 \caption@Debug{%
1929 Restore hypcap definition of \string\capstart\@gobble}%
1930 \let\capstart\caption@ORI@capstart}%
1931 \ifundefined{caption@ORI@float@makebox}}{%
1932 \caption@Debug{%
1933 Restore hyperref redefinition of \string\float@makebox\@gobble}%
1934 \let\float@makebox\caption@ORI@float@makebox}%
1935 }%

\caption@star We define \caption@star here, too, so it's defined but does not make any harm.
1936 \newcommand*\caption@star[2]{#1#2}%

1937 }{%
1938 \caption@ifincompatible{%
1939 \caption@WarningNoLine{%
1940 Forced redefinition of \noexpand\caption since the\MessageBreak
1941 unsupported(!) package option 'compatibility=false'\MessageBreak
1942 was given}%
1943 }{}%

\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.
1944 \def\caption{%
1945 \caption@iftype
1946 {\caption@checkgrouplevel\@empty\caption
1947 \caption@star
1948 {\caption@refstepcounter\@captype}%
1949 {\caption@dblarg{\@caption\@captype}}}%
1950 {\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.
1951 \newcommand*\caption@star[2]{%
1952 \@ifstar{\global\caption@startrue#2[]}{#1#2}}%

\@caption As above, our version has been adapted to the packages we support. Addition-
ally our code is nested by \caption@beginex & \caption@end instead of
\begin group & \end group. Furthermore we use \caption@boxrestore in-
stead of \@parboxrestore so this code also works correctly inside list-based environ-
ments like wide & addmargin. (This, and the fact that we use \linewidth instead
of \hsize inside \@makecaption, solves LATEX PR latex/2472.)
1953 \long\def\@caption#1[#2]#3{%
1954 \ifcaption@star \else
1955 \caption@makecurrent{#1}{#2}%
1956 \caption@ifhypcap{\caption@start}}{%
1957 \fi

```

```

1958     \par
1959     \caption@beginex{#1}{#2}{#3}%
1960         \caption@setfloatcapt{%
1961             \caption@boxrestore
1962             \if@minipage
1963                 \@setminipage
1964             \fi
1965             \caption@normalsize
1966             \ifcaption@star
1967                 \let\caption@@anchor\@firstofone
1968             \fi
1969             \@makecaption{\csname fnum@#1\endcsname}%
1970                 {\ignorespaces\caption@@anchor{#3}}\par
1971             \caption@if@minipage\@minipagetrue\@minipagefalse}%
1972     \caption@end}%

1973 }%

1974 \caption@AtBeginDocument*{%
1975     \let\caption@ORI@capstart\@undefined
1976     \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.

1977 \let\caption@ORI@xfloat\@xfloat
1978 \def\@xfloat#1[#2]{%
1979     \caption@ORI@xfloat{#1}[#2]%
1980     \caption@settype{#1}}%

1981 }

\@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.

1982 \long\def\@makecaption#1#2{%
1983     \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
1984     \caption@@make{#1}{#2}%
1985     \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}}

```

## 2.12 Support for sub-captions

```

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

1986 \def\caption@DeclareSubType sub#1\@nil{%
1987     \caption@Debug{Initializing subtype for `#1'\@gobble}%
1988     \@namedef{subcaption@value#1}{0}%
1989     \@namedef{caption@beginsub#1}{\caption@beginsubfloat{#1}}
1990 \@onlypreamble\caption@DeclareSubType

Initialize the sub-captions defined with \DeclareCaptionSubType...
1991 \caption@For*(subtypelist){\caption@DeclareSubType sub#1\@nil}

Initialize the sub-captions defined with \newsfloat[18]...
1992 \caption@AtBeginDocument*{%

```



```

1993 \@ifundefined{sf@counterlist}{}{%
1994   \@for\sf@temp:=\sf@counterlist\do{%
1995     \expandafter\caption@DeclareSubType\sf@temp\@nil}}}

```

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

```

1996 \newcommand*\caption@subtypehook{%
1997   \ifx\caption\caption@subcaption \else
1998     \caption@ifrefstepcounter{}{%
1999       % no \caption or \subcaption in this (floating) environment yet
2000       \caption@Debug{Increment \@captive\ counter =\the\value\@captive}%
2001       \caption@l@stepcounter\@captive
2002       \let\addcontentsline\subcaption@addcontentsline}%
2003     \ifnum\csname subcaption@value\@captive\endcsname=\value\@captive \else
2004       \caption@Debug{Reset sub\@captive\ counter}%
2005       \expandafter\xdef\csname subcaption@value\@captive\endcsname{%
2006         \the\value\@captive}%
2007       \@stpelt\@subcaptive
2008     \fi
2009     \c@ContinuedFloat=0\relax
2010     \let\caption@resetContinuedFloat\@gobble
2011     \let\caption@addcontentsline\caption@kernel@addcontentsline
2012     \let\caption@setfloatcapt\@firstofone
2013     \caption@clearmargin
2014     \caption@setoptions{sub}%
2015     \caption@setoptions{subfloat}% for subfig-package compatibility
2016     \let\caption\caption@subcaption
2017   \fi}%

```

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

```

2018 \newcommand*\caption@subcaption{%
2019   \caption@iftype
2020     {\caption@checkgrouplevel{sub}\subcaption
2021     \caption@star
2022     {\caption@refstepcounter\@subcaptive}%
2023     {\caption@dblarg{\@caption\@subcaptive}}}%
2024     {\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.

```

2025 \let\caption@kernel@addcontentsline\caption@addcontentsline
2026 \renewcommand*\caption@addcontentsline[2]{%
2027   \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
2028   \caption@kernel@addcontentsline{#1}{#2}%
2029   \sf@ifpositiontop{}{\@listsubcaptions{#1}}%
2030   \caption@addsubcontentslines{#1}}
2031 \newcommand*\caption@addsubcontentslines[1]{%
2032   \begingroup
2033     \subcaption@addcontentslines
2034   \endgroup
2035   \subcaption@clearcontentslines}%

```

```

2036 \caption@AtBeginDocument*{%
2037   \@ifundefined{sf@ifpositiontop}{\let\sf@ifpositiontop@gobbletwo}{}%
2038   \subcaption@clearcontentslines
2039   \g@addto@macro\caption@typehook{\subcaption@checkcontentslines}%
2040   \AtEndDocument{\subcaption@checkcontentslines}}%

ubcaption@addcontentsline Add a pending sub-caption list entry.
2041 \newcommand*\subcaption@addcontentsline[3]{%
2042   \protected@edef\@tempa{%
2043     \noexpand\g@addto@macro\noexpand\subcaption@addcontentslines{%
2044       \noexpand\@namedef{the#2}{\csname the#2\endcsname}%
2045       \ifx\@currentHref\@undefined \else
2046         \noexpand\def\noexpand\@currentHref{\@currentHref}%
2047       \fi
2048       \protect\addcontentsline{#1}{#2}{#3}}}%
2049   \@tempa}

aption@checkcontentslines Checks if the list of pending sub-captions is empty, if not, a warning will be issued.
2050 \newcommand*\subcaption@checkcontentslines{%
2051   \ifx\subcaption@addcontentslines\@empty \else
2052     \subcaption@Error{%
2053       Something's wrong--perhaps a missing \protect\caption\MessageBreak
2054       in the last figure or table}%
2055     \subcaption@clearcontentslines
2056     \fi}

aption@clearcontentslines Clear pending sub-caption list entries.
2057 \newcommand*\subcaption@clearcontentslines{%
2058   \global\let\subcaption@addcontentslines\@empty}

```

## 2.13 Document class & Babel package support

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

```
2059 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}
```

### 2.13.2 The beamer class

```
2060 \@ifclassloaded{beamer}{%
2061   \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.

```

2062 \captionsetup{list=false}
2063 \DeclareCaptionOption{list}{}
2064 \DeclareCaptionOption{listof}{}

```

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

```

2065 \expandafter\let\expandafter\caption@ORI@figure
2066   \csname\string\figure\endcsname
2067 \@namedef{\string\figure}[#1]{%
2068   \caption@ORI@figure[#1]%
2069   \caption@settype{figure}}

```

```

2070 \expandafter\let\expandafter\caption@ORI@table
2071 \csname\string\table\endcsname
2072 \@namedef{\string\table}[#1]{%
2073 \caption@ORI@table[#1]%
2074 \caption@settype{table}}
2075 }{}

```

### 2.13.3 The KOMA-Script classes

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

```

2076 \@ifundefined{scr@caption}{}{%
2077 \caption@AtBeginDocument{\let\scr@caption\caption}}

```

### 2.13.4 The frenchb Babel option

Suppress “Package frenchb.1df Warning: The definition of `\@makecaption` has been changed, frenchb will NOT customize it.” (but only if we emulate this customization)

```

2078 \@ifundefined{caption@frenchb}{}{\caption@frenchb}

```

### 2.13.5 The frenchle/pro package

```

2079 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
2080 \caption@Info{frenchle/pro package is loaded}%
2081 \let\captionfont@ORI\captionfont
2082 \let\captionlabelfont@ORI\captionlabelfont
2083 \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`.

```

2084 \@ifundefined{GOfrench}%
2085 {\let\@tempa\@firstofone}%
2086 {\def\@tempa{\g@addto@macro\GOfrench}}%
2087 \@tempa{%
2088 \let\captionfont\captionfont@ORI
2089 \let\captionfont@ORI\@undefined
2090 \let\captionlabelfont\captionlabelfont@ORI
2091 \let\captionlabelfont@ORI\@undefined
2092 \let\@makecaption\@makecaption@ORI
2093 \let\@makecaption@ORI\@undefined

```

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

```

2094 \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.

```

2095 \let\caption@tcORI\@tablescaption
2096 \def\@tablescaption{\caption@star\relax\caption@tcORI}%

```

```

\ffrench \ffrench and \tfrench reflect \fnum@figure and \fnum@table when
\tfrench used in French mode. These contain additional code which typesets the caption separator \captionseparator instead of the usual colon. Because this breaks with our \makecaption code we have to remove this additional code here.

2097 \let\eatDP\undefined
2098 \let\tempa\empty
2099 \ifx\ffrench\fnum@figure
2100 \l@addto@macro\tempa{\let\fnum@figure\ffrench}%
2101 \fi
2102 \ifx\tfrench\fnum@table
2103 \l@addto@macro\tempa{\let\fnum@table\tfrench}%
2104 \fi
2105 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
2106 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
2107 \tempa

2108 }%
2109 }}

```

## 2.14 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.

2110 \newcommand\caption@ifpackageloaded[1]{%
2111 \testopt{\caption@ifpackageloaded{#1}}{}}
2112 \@onlypreamble\caption@ifpackageloaded

2113 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
2114 \ifpackageloaded{#1}\@firstofone{%
2115 \caption@debug{#1 package is not loaded (yet)\@gobble}%
2116 \caption@atbegindocument}{%
2117 \caption@ifpackageloaded{#1}[#2][#3][#4]}
2118 \@onlypreamble\caption@ifpackageloaded

2119 \newcommand\caption@ifpackageloaded[1]{%
2120 \testopt{\caption@ifpackageloaded{#1}}{}}
2121 \@onlypreamble\caption@ifpackageloaded

2122 \long\def\caption@ifpackageloaded#1[#2]{%
2123 \ifpackageloaded{#1}{%
2124 \caption@info{#1 package is loaded}%
2125 \ifpackagelater{#1}[#2]\@firstoftwo{%
2126 \caption@error{%
2127 For a successful cooperation we need at least version\MessageBreak
2128 `#2' of package #1,\MessageBreak
2129 but only version\MessageBreak
2130 \csname ver@#1.\@pkgextension\endcsname'\MessageBreak
2131 is available}%
2132 \@secondoftwo}%
2133 }{\@secondoftwo}}
2134 \@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.

```

2135 \newcommand*\caption@clearmargin{%
2136   \setcaptionmargin\z@
2137   \let\caption@minmargin\@undefined}

2138 \caption@setbool{needfreeze}{0}
2139 \caption@AtBeginDocument*{%
2140   \caption@ifneedfreeze{%

```

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

```

2141 \newcommand*\caption@freeze{%
2142   \caption@teststar\caption@@freeze\@gobble\@firstofone}%

2143 \newcommand*\caption@@freeze[1]{%
2144   \global\let\caption@SCcontinued\relax
2145   \global\let\caption@SCsetup\@undefined
2146   \global\let\caption@SClentry\@undefined
2147   \global\let\caption@SCtext\@undefined
2148   \global\let\caption@SClabel\@undefined

2149   \let\caption@ORI@ContinuedFloat\ContinuedFloat
2150   \def\ContinuedFloat{%
2151     \let\caption@ORI@setcounter\setcounter
2152     \let\caption@ORI@addtocounter\addtocounter
2153     \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
2154     \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
2155     \caption@ORI@ContinuedFloat
2156     \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
2157     \let\setcounter\caption@ORI@setcounter
2158     \let\addtocounter\caption@ORI@addtocounter}%
2159   \def\captionsetup##1{%
2160     \caption@g@addto@list\caption@SCsetup{##1}%
2161     \caption@setup{##1}}%
2162   \let\caption@ORI\caption
2163   \def\caption{%
2164     \def\caption{\caption@Error{%
2165       Only one \noexpand\caption can be placed in this environment}}%
2166     \let\captionsetup\caption@setup
2167     \let\caption@@refstepcounter\caption@l@stepcounter
2168     \caption@ORI}%
2169   \long\def\@caption##1[##2]##3{%
2170     \@bsphack
2171     \gdef\caption@SClentry{##2}%
2172     \gdef\caption@SCtext{##3}%
2173     \@esphack}%
2174   #1{% is \@gobble in star form, and \@firstofone otherwise
2175     \def\label##1{\@bsphack\gdef\caption@SClabel{##1}\@esphack}}%

2176 }%

```

`\caption@defrost` `\caption@defrost`

```

2177 \newcommand*\caption@defrost{%
2178   \ifx\caption@ORI@ContinuedFloat\@undefined

```

```

2179 \caption@SCcontinued
2180 \ifx\caption@SCsetup\@undefined \else
2181 \expandafter\captionsetup\expandafter{\caption@SCsetup}%
2182 \fi
2183 \ifx\caption@SCtext\@undefined \else
2184 \expandafter\expandafter\expandafter\caption
2185 \expandafter\expandafter\expandafter[%
2186 \expandafter\expandafter\expandafter{%
2187 \expandafter\caption@SClentry\expandafter}\expandafter]%
2188 \expandafter{\caption@SCtext}%
2189 \fi
2190 \ifx\caption@SClabel\@undefined \else
2191 \expandafter\label\expandafter{\caption@SClabel}%
2192 \fi
2193 \else
2194 \caption@Error{Internal Error:\MessageBreak
2195 \noexpand\caption@defrost in same group as \string\caption@freeze}%
2196 \fi}%

2197 }{}%
2198 \caption@undefbool{needfreeze}}

```

### 2.14.1 The float package

The float package usually do not use the L<sup>A</sup>T<sub>E</sub>X 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.)

```

2199 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
2200 \@ifpackageloaded{floatrow}{%
2201 \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
2202 }{}%

```

`\@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>]`.

```

2203 \let\caption@ORI@float@setevery\@float@setevery
2204 \def\@float@setevery#1{%
2205 \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.

```
2206 \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.

```
2207 \renewcommand\caption@setfloatcapt{%
2208 \let\@makecaption\caption@make
2209 \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.

```
2210 \float@getstyle\float@style{#1}%
2211 \caption@setstyle*\float@style
2212 \caption@setoptions\float@style
2213 }{}%
2214 \caption@freezeHref % will be defrosted in \float@makebox
2215 \caption@ORI@float@setevery{#1}}%
```

`\caption@typehook` L<sup>A</sup>T<sub>E</sub>X 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.

```
2216 \g@addto@macro\caption@typehook{%
2217 \expandafter\ifx\csname #1name\endcsname\relax
2218 \expandafter\let\csname #1name\expandafter\endcsname
2219 \csname fname@#1\endcsname
2220 \fi}%
```

`\fs@plaintop` `\fs@boxed` Since the float styles `plaintop` and `boxed` don't use `\abovecaptionskip` which could be set with `skip=` (`plaintop` uses `\belowcaptionskip` instead of `\abovecaptionskip`, and `boxed` uses a fixed space of 2pt) we patch the according float style macros here to change this.

```
2221 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2222 \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).

```
2223 \providecommand*\float@ifstyle[1]{%
2224 \expandafter\ifx\csname fst@#1\endcsname\relax
2225 \expandafter\@secondoftwo
2226 \else
```

```

2227     \expandafter\@firstoftwo
2228     \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.

```

2229 \providecommand*\float@getstyle[2]{%
2230     \edef#1{%
2231         \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2232         \expandafter\expandafter\expandafter\noexpand
2233         \csname fst@#2\endcsname}%
2234     \edef#1{#1}%
2235     \caption@Debug{floatstyle{#2} = '#1'}}%

```

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

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

```

2236 \providecommand*\float@setstyle[2]{%
2237     \expandafter\edef\csname fst@#1\endcsname{%
2238         \expandafter\noexpand\csname fs@#2\endcsname}}%

```

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

```

2239 \providecommand*\float@dostyle[1]{%
2240     \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.)

```

2241 \providecommand*\float@ifcaption[1]{%
2242     \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2243     \expandafter\@firstoftwo
2244     \else
2245     \expandafter\@secondoftwo
2246     \fi}%

```

```

2247 }{%
2248     \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2249     \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2250 % \clearcaptionsetup{boxed}% used by the floatrow package?
2251 }

```



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

```
2252 \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’.

```
2253 \caption@ifbool{ruled}{%
2254   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
2255 }{%
2256   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2257 }
2258 \caption@undefbool{ruled}
```

### 2.14.2 The floatflt package

```
2259 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

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

```
2260 \let\caption@ORI@floatingfigure\floatingfigure
2261 \def\floatingfigure{%
2262   \caption@floatflt{figure}%
2263   \caption@ORI@floatingfigure}%

```

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

```
2264 \let\caption@ORI@floatingtable\floatingtable
2265 \def\floatingtable{%
2266   \caption@floatflt{table}%
2267 %   \caption@setautoposition b%
2268   \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.

```
2269 \newcommand*\caption@floatflt[1]{%
2270   \caption@settype{#1}%
2271   \caption@clearmargin
2272   \caption@setoptions{floating#1}%
2273   \let\caption@boxrestore\@parboxrestore}%
2274 }{ }
```

### 2.14.3 The fltpage package

```
2275 \caption@ifPackageLoaded{fltpage}[1998/10/29 v.0.3]{%
2276   \caption@setbool{needfreeze}{1}%

```

\FP@helpNote **Original code:**

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

2277 \renewcommand\FP@helpNote[2]{%
2278   \begingroup % save \caption@thepage
2279     \caption@pageref{#2}%
2280     \typeout{FP#1 is inserted on page \caption@thepage!}%
2281   \endgroup}%

```

\FP@floatBegin **Original code:**

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

2282 \renewcommand*\FP@floatBegin[1]{%
2283   \def\@capttype{#1}%
2284   \let\FP@LabelText\@empty
2285   \begin{lrbox}{\FP@floatCorpusBOX}%
2286     \caption@ifFPrefcap
2287       {\caption@freeze\relax}%
2288       {\def\label##1{\@bsphack\gdef\FP@LabelText{##1}\@esphack}%
2289         \caption@freeze*}}%

```

\FP@floatEnd **Original code:**

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

```

```

{\addtocounter{\@capttype}{-1}}
{\begin{\@capttype}[b!]}%
  \ifthenelse{\equal{\FP@guide}{\@empty}}%
  {}{\ifthenelse{\equal{\@capttype}{figure}}%
    {\renewcommand{\fnum@figure}{\old@Fnum\ {\FP@guide}}}%
    {\renewcommand{\fnum@table}{\old@Fnum\ {\FP@guide}}}%
  \setlength{\abovecaptionskip}{2pt plus2pt minus 1pt} % length above caption
  \setlength{\belowcaptionskip}{2pt plus2pt minus 1pt} % length above caption
  \FP@separatorCaption%
  \ifthenelse{\equal{\FP@optionalCaptionText}{\@empty}}%
  {\FP@savedCaptionCommand{\expandafter\protect\FP@CaptionText}}%
  {\FP@savedCaptionCommand[\expandafter\protect\FP@optionalCaptionText]%
    {\expandafter\protect\FP@CaptionText}}%
\end{\@capttype}}%
}%

2290 \renewcommand*\FP@floatEnd{%
2291   \end{lrbox}%

2292   \stepcounter{FP@\@capttype C}%
2293   \caption@label\FP@positionLabel
2294   \FP@helpNote\@capttype\FP@positionLabel

2295   \let\FP@ContinuedFloat\relax
2296   \FP@float
2297   {\FP@positionLabel}% location label test
2298   {\begin{\@capttype[p!]}%
2299     \FP@ContinuedFloat
2300     \usebox\FP@floatCorpusBOX
2301     \caption@ifFPlistcap
2302       {\caption@refstepcounter\@capttype
2303         \expandafter\caption@makecurrent\expandafter\@capttype
2304           \expandafter{\caption@SClentry}}%
2305       {\expandafter\captionlistentry\expandafter{\caption@SClentry}}%
2306     \caption@@anchor\relax
2307     \ifx\FP@LabelText\@empty \else
2308       \expandafter\label\expandafter{\FP@LabelText}%
2309     \fi
2310   \end{\@capttype}}%
2311   {\let\FP@ContinuedFloat\ContinuedFloat
2312     \caption@ifcompatibility
2313       {\let\caption@ContinuedFloat\@gobble}%
2314       {\caption@clearsetup{\ContinuedFloat}%
2315         \caption@clearsetup{continued\@capttype}}}%
2316   {\begin{\@capttype[b!]}%
2317     \FP@ContinuedFloat
2318     \let\FP@savedSetfnumCommand\caption@setfnum
2319     \def\caption@setfnum##1{%
2320       \FP@savedSetfnumCommand{##1}%
2321       \ifx\FP@guide\@empty \else
2322         \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ {\FP@guide}}%
2323       \fi}%
2324     \setlength{\abovecaptionskip}{2pt plus 2pt minus 1pt}% length above caption
2325     \setlength{\belowcaptionskip}{2pt plus 2pt minus 1pt}% length below caption
2326     \caption@setoptions{FP\@capttype}%

```

```

2327         \FP@separatorCaption
2328         \caption@ifFPlistcap{}{\caption@set@bool\caption@iflof0}%
2329         \caption@defrost
2330         \end\@capttype}%
2331     }%

2332 \caption@For{typelist}{%
2333     \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2334 }{%
2335 \let\caption@ifFPlistcap\@undefined
2336 \let\caption@ifFPrefcap\@undefined
2337 }

```

#### 2.14.4 The hyperref package

```

2338 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2339 \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2340 \caption@WarningNoLine{%
2341     Hyperref support is turned off\MessageBreak
2342     because hyperref has stopped early}%
2343 }{%
2344 \g@addto@macro\caption@prepareslc{\measuring@true}%

```

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

```

2345 \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

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

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

```

2346 \renewcommand*\caption@makecurrent[2]{%
2347 \caption@makecurrentHref{#1}%
2348 \caption@Debug{hyperref current=\@currentHref}%
2349 \def\@currentlabelname{#2}}%

2350 \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

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

```

2351 \renewcommand\caption@@anchor[1]{%
2352 \caption@Debug{hyperref anchor: \@currentHref}%
2353 % If we cannot have nesting, the anchor is empty.
2354 \ifHy@nesting
2355 \hyper@@anchor{\@currentHref}{#1}%
2356 \else
2357 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2358 \fi}%

2359 \g@addto@macro\caption@prepareslc{\let\caption@@anchor\@firstofone}%

```

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

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

will be used instead of

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

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

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

```
2360 \renewcommand*\caption@@ContinuedFloat[1]{%
2361 \ifundefined{theH#1}{}{}%
2362 \expandafter\l@addto@macro\csname theH#1\endcsname{%
2363 \@alpha\c@ContinuedFloat}}}%
```

### The `hycap` option

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

```
2364 \newif\if@capstart
```

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

```
2365 \def\caption@start{\caption@ifhycap{%
```

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

```
2366 \caption@makestart\@captype
2367 \caption@startanchor\@currentHref
```

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

```
2368 \global\@capstarttrue
2369 \let\hc@currentHref\@currentHref
2370 \def\caption@makecurrentHref##1{%
2371 \global\@capstartfalse
2372 \global\let\@currentHref\hc@currentHref}%
```

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

```
2373 \let\caption@@anchor\@firstofone
2374 }{}%}
```

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

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

-or-

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

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

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

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

```
2375 \newcommand*\caption@makestart[1]{%
2376 \begingroup
2377 \Hy@hypertextnamesfalse
2378 % \gdef\@currentHlabel{}%
2379 \hyper@makecurrent{#1.caption}%
2380 \endgroup
2381 \caption@Debug{hycap start=\@currentHref}}%
```

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

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

```
2382 \newcommand*\caption@startanchor[1]{%
2383 \caption@Debug{hycap anchor: #1}%
2384 \ifvmode\begingroup
2385 \@tempdima\prevdepth
2386 \nointerlineskip
2387 \vspace*{-\caption@hycapSPACE}%
2388 \caption@anchor{#1}%
2389 \vspace*{\caption@hycapSPACE}%
2390 \prevdepth\@tempdima
2391 \endgroup\else
2392 \caption@Warning{Internal Warning: Vertical TeX mode expected}%
2393 \caption@anchor{#1}%
2394 \fi}%
```

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

```
2395 \newcommand*\caption@anchor[1]{%
2396 \ifmeasuring@ \else
2397 \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2398 \fi}%
```

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

```
2399 \ifx\HyperRaiseLinkLength\@tempdima
2400 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2401 \else
2402 \let\caption@raisedlink\Hy@raisedlink
2403 \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.

```
2404 \def\caption@@start{%
2405 \@ifundefined{hc@currentHref}{%
```

```

2406         \caption@Warning{%
2407             The option 'hycap=true' will be ignored for this\MessageBreak
2408             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.

2409     \def\caption@freezeHref{%
2410         \let\caption@ORI@start\caption@start
2411         \def\caption@start{\let\caption@start\caption@ORI@start}%

2412 %     \let\caption@ORI@@start\caption@@start
2413 %     \l@addto@macro\caption@subtypehook{%
2414 %         \let\caption@@start\caption@ORI@@start}%

2415     \global\let\caption@currentHref\@undefined
2416     \def\caption@@start{\global\let\caption@currentHref\@currentHref}%

2417     \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2418     \renewcommand*\caption@setfloatcapt{%
2419         \ifx\caption@currentHref\@undefined \else
2420             \let\caption@@anchor\@firstofone
2421         \fi
2422         \caption@ORI@setfloatcapt}}%

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

2423     \def\caption@defrostHref{%
2424         \ifx\caption@currentHref\@undefined \else
2425             \caption@startanchor\caption@currentHref
2426             \global\let\caption@currentHref\@undefined
2427         \fi}%

\float@makebox Do our own redefinition of \float@makebox, if it was redefined by the hyperref pack-
age.

2428     \@ifundefined{HyOrg@float@makebox}{}{%
2429         \caption@Debug{%
2430             Redefining \noexpand\float@makebox (again)\@gobble}%
2431         \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
2432         \renewcommand\float@makebox[1]{%
2433             \HyOrg@float@makebox{#1}\relax \caption@defrostHref}}%
2434     }%

2435 }}{}

```

## 2.14.5 The hycap package

```

2436 \caption@IfPackageLoaded{hycap}{% v1.0
2437     \ifx\caption@start\relax \else % hyperref hasn't stopped early

```

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

```

\capstart We do this simply by mapping \capstart to \caption@start, although our code
          does not behave exactly like the original one: The original \capstart has an effect on

```

the next `\caption` only but our version affects *all* `\captions` in the same environment, at least unless a new `\capstart` will be placed.

```
2438 \let\caption@ORI@capstart\capstart % save for compatibility mode
2439 \let\capstart\caption@start
2440 \let\caption@start\relax
2441 \let\caption@@start\relax
```

`\caption@hyppcapspace` Furthermore we map our `\caption@hyppcapspace` to `\hyppcapspace` offered by the `hyppcap` package.

```
2442 \caption@set@bool\caption@ifhyppcap1%
2443 \renewcommand*\caption@hyppcapspace{\hyppcapspace}%

2444 \fi}{}}
```

### 2.14.6 The listings package

```
2445 \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`.

```
2446 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2447 \def\lst@MakeCaption#1{%
2448 \caption@setoptions{lstlisting}%
```

If the `position=` is set to `auto`, we take over the `captionpos=` setting from the `listings` package. Note that we won't do this otherwise, so `listings` settings like `abovecaptionskip=0pt`, `belowcaptionskip=10pt`, `captionpos=t` will *not* cause different outputs with or without the `caption` package loaded.

```
2449 \caption@setautoposition{#1}%
2450 \caption@begin{lstlisting}%
2451 \caption@ORI@lst@MakeCaption{#1}%
2452 \caption@end}%
```

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

```
2453 \providecommand*\ext@lstlisting{lol}%

2454 }{}}
```

### 2.14.7 The longtable package

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

```
2455 \providecommand*\LTcapttype{table}

2456 \caption@IfPackageLoaded{longtable}[1995/05/24 v3.14]{%
2457 \RequirePackage{ltcaption}[2007/09/01]%
2458 \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`.



```

2459 \caption@AtBeginDocument{%
2460 \let\caption@ORI@LT@array\LT@array
2461 \renewcommand*\LT@array{%

\captionsetup for longtable:
2462 \global\let\caption@opt@@longtable\@undefined
2463 \def\captionsetup##1{\noalign{\LT@captionsetup{##1}}}%
2464 \def\LT@captionsetup##1{%
2465 \@captionsetup\@gobble[@longtable]{##1}%
2466 \global\let\caption@opt@@longtable\caption@opt@@longtable}%

\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2467 \def\@captionabovetrue{\LT@captionsetup{position=t}}%
2468 \def\@captionabovefalse{\LT@captionsetup{position=b}}%

\captionlistentry for longtable:
2469 \def\captionlistentry##1{\noalign{%
2470 \caption@listentry[\LTcaption]{##1}}}%

\ContinuedFloat for longtable:
(Commented out, since it's not deeply tested and quite useless anyway)
Note: hyperref versions < v6.76j uses 2x \hyper@makecurrent
2471 % \caption@ifhypcap{%
2472 % \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2473 % \def\hyper@makecurrent##1{%
2474 % \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2475 % \caption@makestart{##1}%
2476 % \let\Hy@LT@currentHlabel\@currentHlabel
2477 % \let\Hy@LT@currentHref\@currentHref
2478 % \def\hyper@makecurrent###1{%
2479 % \let\@currentHlabel\Hy@LT@currentHlabel
2480 % \let\@currentHref\Hy@LT@currentHref}}%
2481 % \let\caption@ORI@ContinuedFloat\ContinuedFloat
2482 % \def\ContinuedFloat{\noalign{%
2483 % \gdef\caption@setContinuedFloat{%
2484 % \let\caption@resetContinuedFloat\@gobble}%
2485 % \def\caption@setoptions####1{%
2486 % \g@addto@macro\caption@setContinuedFloat{%
2487 % \caption@setoptions{####1}}}%
2488 % \let\@captiontype\LTcaptiontype
2489 % \caption@ORI@ContinuedFloat}}%
2490 % }{%
2491 % \def\ContinuedFloat{\noalign{%
2492 % \caption@Error{%
2493 % \noexpand\ContinuedFloat inside longtables\MessageBreak
2494 % is only available with 'hypcap=true'}}}%
2495 % }%
2496 % \global\let\caption@setContinuedFloat\@empty
2497 \def\ContinuedFloat{\noalign{%
2498 \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2499 \caption@ORI@LT@array}}%

```

\LT@c@ption The original implementation:

```

\def\LT@c@ption#1[#2]#3{%
\LT@makecaption#1\fnum@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}`:

```

2500 \def\LT@caption#1[#2]#3{%
2501   \LT@makecaption#1{\csname fnum@\LTcapttype\endcsname}{#3}%
2502   \def\@currentlabelname{#2}%
2503   \caption@addcontentsline\LTcapttype{#2}}%

```

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

The original definition:

```

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

```

Our definition:

```

2504 \renewcommand\LT@makecaption[3]{%
2505   \caption@LT@make{%

```

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

```

2506   \caption@startrue#1\caption@starfalse

```

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

```

2507   \caption@settype*\LTcapttype
2508   \ifdim\LTcapwidth=4in \else
2509     \setcaptionwidth\LTcapwidth
2510   \fi
2511   \caption@setoptions{longtable}%
2512 %   \caption@setContinuedFloat
2513   \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!

```

2514   \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%

```

```

2515      \caption@resetContinuedFloat\LTcapttype
2516      \caption@begin\LTcapttype
2517      \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...)

```

2518      \vskip-\ht\strutbox

```

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

```

2519      \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2520      \caption@@make{#2}{#3}\endgraf
2521      \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2522      \caption@end}}%
2523 }{}

```

## 2.14.8 The picinpar package

```

2524 \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.

```

2525 \long\def\figwindow[#1,#2,#3,#4] {%
2526   \caption@window{figure}%
2527   \caption@setoptions{figwindow}%
2528   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
2529 \long\def\tabwindow[#1,#2,#3,#4] {%
2530   \caption@window{table}%
2531   \caption@setoptions{tabwindow}%
2532   \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.

```

2533 \newcommand*\caption@window[1]{%
2534   \let\caption@boxrestore\@parboxrestore
2535   \let\@makecaption\caption@@make
2536   \caption@setautoposition b%
2537   \caption@settype{#1}%
2538   \caption@clearmargin}%

```

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

```

2539 \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`.

```

2540   \ifdim\picwd=\z@

```

```

2541      \let\caption@makecurrent\@gobbletwo
2542      \let\caption@@start\relax
2543      \caption@prepareslc
2544      \fi

```

The argument #1 could contain simply the caption text (e.g. A figure caption), but it could also contain an optional argument, the *<lst\_entry>* (e.g. [An entry to the LOF] {A figure caption}). Therefore we have to test if #1 begins with [ or not; furthermore we support a starred variant – as in \caption\* – so we test for \*, too.

```

2545      \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2546      \if\@tempa*%
2547          \let\@tempa\@firstofone
2548      \else\if\@tempa[%
2549          \let\@tempa\@firstofone
2550      \else
2551          \let\@tempa\@empty
2552      \fi\fi
2553      \expandafter\caption\@tempa{#1}}%
2554  {} }

```

#### 2.14.9 The picins package

\piccaptiontype \piccaptiontype{<type>}

We offer this macro for changing the *<type>* of the caption, so the user doesn't have to redefine \caption, as proposed in the picins documentation.

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

```

2555 \newcommand*\piccaptiontype[1]{\def\@piccaptiontype{#1}}
2556 \caption@ifPackageLoaded{picins}{%

```

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

```

2557 \ifundefined{@piccaptiontype}{%
2558     \caption@iftype{%
2559         \let\@piccaptiontype\@captiontype
2560     }{%
2561         \def\@piccaptiontype{figure}%
2562     }%
2563 }{}%
2564 \let\@captiontype\undefined

```

\piccaption The original code:

```

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

```

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

```

2565 \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.

```
2566 \let\caption@ORI@make@piccaption\make@piccaption
2567 \def\make@piccaption{%
2568   \let\caption@ORI\caption
2569   \long\def\caption[##1]##2{%
2570     \caption@freezeHref % will be defrosted in \ivparpic
2571     \caption@settype\@piccapttype
2572 % \ifnum\c@piccaptionpos>2\relax
2573     \caption@clearmargin
2574 % \else
2575 %   \captionwidth\z@ % do not use "width=" setting
2576 % \fi
2577     \caption@setoptions{parpic}%
2578     \let\caption@boxrestore\@parboxrestore
2579     \caption@setautoposition b%
2580     \expandafter\expandafter\expandafter\caption@ORI
2581     \expandafter\expandafter\expandafter[%
2582     \expandafter\expandafter\expandafter{%
2583       \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
2584 \caption@ORI@make@piccaption
2585 \let\caption\caption@ORI%
```

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

```
2586 \let\caption@ORI@ivparpic\ivparpic
2587 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
```

```

2588 \let\caption@ORI@noindent\noindent
2589 \def\noindent{%
2590 \caption@defrostHref
2591 \let\noindent\caption@ORI@noindent
2592 \noindent}%
2593 \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2594 \let\noindent\caption@ORI@noindent}%

2595 }{%
2596 \let\piccaptiontype\@undefined
2597 }

```

#### 2.14.10 The rotating package

```

2598 \caption@ifPackageLoaded{rotating}[1995/08/22 v2.10]{%

\rotcaption Make \rotcaption* work.

2599 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
2600 % \let\@rotcaption\@undefined

\rotcaptionof Make \rotcaptionof(*) work.

2601 \def\rotcaptionof{%
2602 \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

\@makerotcaption Original (bugfixed) code:

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

```

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

```

2603 \long\def\@makerotcaption#1#2{%
2604 \ifdim\captionwidth=\z@
2605 \setcaptionwidth{.8\textheight}%
2606 \caption@slc{#1}{#2}{.8\vsizel}%
2607 \let\caption@makerot\caption@@make
2608 \caption@clearmargin
2609 % \long\def\caption@parbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2610 % (not needed because \rotatebox uses an \hbox anyway)
2611 \let\caption@parbox\@secondoftwo}%
2612 \caption@set@bool\caption@ifslc0% been there, done that
2613 \fi
2614 \rotatebox{90}{\caption@makerot{#1}{#2}}%
2615 \nobreak\hspace{12pt}}%

```

```

2616 \newcommand\caption@makerot[2]{%
2617   \begin{minipage}\captionwidth\caption@make{#1}{#2}\end{minipage}}%

2618 \caption@For{typelist}{%
2619   \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
2620   \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
2621 {}{}

```

### 2.14.11 The sidecap package

```

2622 \caption@IfPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
2623   \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.)

```

2624 \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.

```

2625 \let\caption@ORI@SC@zfloat\SC@zfloat
2626 \def\SC@zfloat#1#2#3[#4]{%

```

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

```

2627   \let\caption@ORI\caption
2628   \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
2629   \let\caption\caption@ORI

```

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

```

2630   \caption@settype*{#2}%
2631   \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.

```

2632 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
2633 \def\endSC@FLOAT{%

```

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

```

2634   \let\caption@ORI@settype\caption@settype
2635   \def\caption@settype##1{% will be done in \@xfloat
2636     \caption@ORI@settype*{##1}% do not change \@currentlabel
2637     \caption@setSC@justify
2638   }%% \caption@setoptions{SCfloat}%
2639   \caption@setoptions{SC\@capttype}%
2640   \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

```
2641 \def\caption@setSC@justify{%
2642   \caption@clearmargin
2643   \ifundefined{SC@justify}{}{%
2644     \ifx\SC@justify\@empty \else
2645       \let\caption@hj\SC@justify
2646       \let\SC@justify\@empty
2647     \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!

```
2648 \let\caption\SC@orig@caption
2649 \def\SC@orig@caption[#1]##2{\caption@defrost}%
```

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

```
2650 \caption@setSC@justify % for compatibility mode
2651 \caption@ORI@endSC@FLOAT}%

2652 \newcommand*\caption@For@SC[2]{%
2653   \def#1{b}% = \sidecaptionvpos{#2}{b} (v1.6)
2654   \newenvironment{SC#2}%
2655     {\SC@float[#1]{#2}}{\endSC@float}%
2656   \newenvironment{SC#2*}%
2657     {\SC@dblfloat[#1]{#2}}{\endSC@dblfloat}}%
2658 \@onlypreamble\caption@For@SC
2659 \caption@For{typelist}{%
2660   \expandafter\caption@For@SC\csname SC@#1@vpos\endcsname{#1}}%
2661 }{ }
```

## 2.14.12 The subfigure package

```
2662 \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.

```
2663 \def\sfc@positiontop{%
2664   \ifx\@c@type\@undefined
2665     \expandafter\@gobbletwo
2666   \else\ifx\@c@type\relax
2667     \expandafter\expandafter\expandafter\@gobbletwo
2668   \else
2669     \expandafter\expandafter\expandafter\sfc@positiontop
2670   \fi\fi}

2671 \def\sfc@if@position@top{%
2672   \ifundefined{if\@c@type topcap}%
2673     {\@gobbletwo}%
2674     {\@nameuse{if\@c@type topcap}}%
2675     \expandafter\@firstoftwo
2676   \else
```



```

2677         \expandafter\@secondoftwo
2678         \fi}}
2679 }{}

```

### 2.14.13 The supertabular package

```

2680 \caption@ifpackageloaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.
2681 \let\caption@ORI@tablecaption\tablecaption
2682 \def\tablecaption{\caption@star\relax\caption@ORI@tablecaption}%

```

\ST@caption The original code:

```

\long\def\ST@caption#1[#2]#3{\par%
\addcontentsline{\csname ext@#1\endcsname}{#1}%
{\protect\numberline{%
\csname the#1\endcsname}{\ignorespaces #2}}
\begingroup
\@parboxrestore
\normalsize
\if@topcaption \vskip -10\p@ \fi
\@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
\if@topcaption \vskip 10\p@ \fi
\endgroup}

2683 \long\def\ST@caption#1[#2]#3{\par%
2684 \caption@settype*{#1}%
2685 \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.

```

2686 \def\caption@fixposition{%
2687 \caption@setposition{\if@topcaption t\else b\fi}}%

2688 \caption@beginex{#1}{#2}{#3}%
2689 \caption@boxrestore
2690 \caption@normalsize
2691 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2692 \caption@end}%

2693 }{}%
2694 \let\caption@setSTposition\@undefined
2695 }

```

### 2.14.14 The threeparttable package

```

2696 \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.
2697 \let\caption@ORI@threeparttable\threeparttable
2698 \renewcommand*\threeparttable{%
2699 \caption@settype{table}%

```

```

2700     \caption@setposition a% ?
2701     \caption@clearmargin
2702     \caption@setoptions{threeparttable}%
2703     \caption@ORI@threeparttable}%

```

\measuredfigure Same here...

```

2704 \let\caption@ORI@measuredfigure\measuredfigure
2705 \renewcommand*\measuredfigure{%
2706     \caption@settype{figure}%
2707     \caption@setposition a% ?
2708     \caption@clearmargin
2709     \caption@setoptions{measuredfigure}%
2710     \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}

2711 \def\TPT@caption#1[#2]#3{%
2712     \gdef\TPT@docapt{%
2713         \global\let\TPT@docapt\@undefined
2714         \caption@setautoposition\caption@TPT@position
2715         \TPT@LA@caption{#1}[#2][#3]}%
2716     \ifx\TPT@hsize\@empty
2717         \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
2718         \gdef\caption@TPT@position{t}%
2719         \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2720     \else
2721         \def\caption@TPT@position{b}%
2722         \TPT@docapt
2723     \fi
2724     \ignorespaces}%

2725 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2726 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%

2727 }{}

```

## 2.14.15 The wrapfig package

```

2728 \caption@IfPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)

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

2729 \providecommand*\float@ifstyle[1]{%
2730     \expandafter\ifx\csname fst@#1\endcsname\relax
2731         \expandafter\@secondoftwo
2732     \else
2733         \expandafter\@firstoftwo
2734     \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.

```

2735 \newcommand*\caption@restylewrapfloat[1]{%
2736   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2737     \csname wrap#1\endcsname
2738   \caption@Error{%
2739     For a successful cooperation of the 'wrapfig' package\MessageBreak
2740     with the 'float' package you should load the 'wrapfig'\MessageBreak
2741     package *after* the 'float' package}%
2742   \else
2743     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2744       \csname wrap#1\endcsname
2745     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
2746     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2747       \csname wrap#1\endcsname
2748   \fi}%

```

`\caption@wrapfloat`

```

2749 \newcommand*\caption@wrapfloat[1]{%
2750   \caption@settype*{#1}%
2751   \float@ifstyle{#1}{%
2752     \ifx\WF@floatstyhook\@undefined
2753       \caption@Error{%
2754         For a successful cooperation of the 'wrapfig' package\MessageBreak
2755         with the 'float' package you should use at least\MessageBreak
2756         'wrapfig' version 3.6}%
2757     \else
2758       \float@dostyle{#1}%
2759     \fi}}%
2760   \caption@clearmargin
2761   %%% \caption@setoptions{wrapfloat}%
2762   \caption@setoptions{wrap#1}%
2763   \@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`.)

```

2764 \caption@restylewrapfloat{figure}%
2765 \caption@restylewrapfloat{table}%
2766 \caption@For{typelist}{%
2767   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
2768   \caption@restylewrapfloat{#1}}%
2769 \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.

```

2770 \ifundefined{float@restyle}{}{}%
2771 \toks@=\expandafter{\float@restyle{#1}% (env may or may not be defined)
2772 \caption@restylewrapfloat{#1}}%
2773 \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2774 \@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`.

```

2775 \let\caption@ORI@wrapfloat\wrapfloat
2776 \def\wrapfloat#1{%
2777 \float@ifstyle{#1}{%
2778 \caption@Error{%
2779 For a successful cooperation of the 'wrapfig' package\MessageBreak
2780 with the 'float' package you should load the 'wrapfig'\MessageBreak
2781 package *right after* the 'float' package}}{}%
2782 \caption@ORI@wrapfloat{#1}}%

```

2783 `\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:

```

2784 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2785 \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2786 \global\setbox\WF@box\top\bggroup \setlength\hsize{#2}%
2787 \caption@start
2788 \ifdim\hsize>\z@ \@parboxrestore \else
2789 \setbox\z@\hbox\bggroup \let\wf@@caption\caption \let\caption\wf@caption
2790 \ignorespaces \fi}%
2791 {}

```

## References

- [1] Frank Mittelbach and Michel Goossens:  
*The L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> bundle*,  
2007-01-09
- [4] Victor Eijkhout:  
*An introduction to the Dutch L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X*,  
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  $\text{\LaTeX}$* ,  
1997/09/26
- [16] Rolf Niepraschk & Hubert Gäßlein:  
*The sidecap package*,  
2003/06/06
- [17] Steven D. Cochran:  
*The subfigure package*,  
2002/07/02
- [18] Steven D. Cochran:  
*The subfig package*,  
2005/07/05
- [19] Johannes Braams and Theo Jurriens:  
*The supertabular environment*,  
2002/07/19
- [20] Donald Arseneau:  
*Three part tables: title, tabular environment, notes*,  
2003/06/13
- [21] Donald Arseneau:  
*WRAPFIG.STY ver 3.6*,  
2003/01/31
- [22] Anne Brüggemann-Klein:  
*Einführung in die Dokumentverarbeitung*,  
B.G. Teubner, Stuttgart, 1989
- [23] Heiko Oberdiek:  
*The refcount package*,  
2006/02/20