

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

Axel Sommerfeldt  
[caption@sommerfee.de](mailto:caption@sommerfee.de)

2007/09/16

## 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 provides 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, like the caption package and the subfig package.

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> itself or L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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

<a href="#">caption-eng.pdf</a>	The English documentation
<a href="#">caption-rus.pdf</a>	The Russian documentation <sup>1</sup>
<a href="#">caption-deu.pdf</a>	The German documentation

---

<sup>\*</sup>This package has version number v3.1a, last revised 2007/09/16.

<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 . . . . .	11
1.6	Indentions . . . . .	13
1.7	Styles . . . . .	14
1.8	Formats . . . . .	15
1.9	Label formats . . . . .	16
1.10	Label separators . . . . .	17
1.11	Text formats . . . . .	17
1.12	Fonts . . . . .	18
1.13	Justifications . . . . .	20
1.13.1	The ragged2e package . . . . .	20
1.14	Vertical spaces before and after captions . . . . .	21
1.15	Positioning . . . . .	22
1.16	Hooks . . . . .	23
1.17	Lists . . . . .	23
1.18	Debug option . . . . .	24
1.19	Document classes & Babel support . . . . .	24
1.19.1	The standard L <sup>A</sup> T <sub>E</sub> X classes . . . . .	24
1.19.2	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes . . . . .	24
1.19.3	The beamer class . . . . .	26
1.19.4	The KOMA-Script classes . . . . .	27
1.19.5	The NTG Dutch classes . . . . .	27
1.19.6	The frenchb Babel option . . . . .	28
1.19.7	The frenchle/pro package . . . . .	28
1.20	Execution of options . . . . .	29
1.21	Typesetting the caption . . . . .	29
1.22	Bonus features . . . . .	33
1.23	subfig package adaption . . . . .	34
<b>2</b>	<b>Main package</b>	<b>36</b>
2.1	Identification . . . . .	36
2.2	Loading the kernel . . . . .	36
2.3	Check against incompatible packages . . . . .	36
2.4	Check document class . . . . .	36
2.5	Adaption to the $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes . . . . .	37

2.6	Emulation of the KOMA-Script commands	37
2.7	Option for configuration files	39
2.8	Options for figure and table	39
2.9	caption v1.x compatibility options	39
2.10	caption2 v2.x compatibility options	40
2.11	Obsolete caption v3.0 options	40
2.12	ftpage package support options	41
2.13	hyperref package support options	41
2.14	Processing of options	41
2.15	Option ‘type’, \captionof and \ContinuedFloat	41
2.16	Internal helpers	44
2.17	\caption, \@caption, and \@makecaption	46
2.18	Document class & Babel package support	52
2.18.1	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	52
2.18.2	The beamer class	53
2.18.3	The KOMA-Script classes	53
2.18.4	The frenchb Babel option	53
2.18.5	The frenchle/pro package	53
2.19	Package support	54
2.19.1	The float package	56
2.19.2	The floatflt package	59
2.19.3	The ftpage package	60
2.19.4	The hyperref package	62
2.19.5	The hypcap package	65
2.19.6	The listings package	66
2.19.7	The longtable package	66
2.19.8	The picinpar package	69
2.19.9	The picins package	70
2.19.10	The rotating package	71
2.19.11	The sidecap package	72
2.19.12	The subfigure package	74
2.19.13	The supertabular package	74
2.19.14	The threeparttable package	75
2.19.15	The wrapfig package	76

# 1 Kernel

## 1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2007/09/16 v3.1a 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` his 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  $\backslash@removeelement$  is using  $\backslashedef$  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@ifbool
\caption@undefbool \caption@setbool{\langle name\rangle}{\langle value\rangle}
(with value = false/true/no/yes/off/on/0/1)
\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>}
74 \newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}

\caption@withoptargs \caption@withoptargs{<cmd>}
75 \newcommand*\caption@withoptargs[1]{%
76   \@ifstar
77   {\def\caption@tempa{*}\caption@@withoptargs#1}%
78   {\def\caption@tempa{}\caption@@withoptargs#1}}

79 \def\caption@@withoptargs#1{%
80   \@ifnextchar[%]
81   {\caption@@@withoptargs#1}%
82   {\caption@@@withoptargs#1}}

83 \def\caption@@@withoptargs#1[#2]{%
84   \l@addto@macro\caption@tempa{[#{#2}]}%
85   \caption@@withoptargs#1}

86 \def\caption@@@withoptargs#1{%
87   \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.)

88 \newcommand\caption@DoCheckCommand[2]{%
89   \begingroup
90     \let\@tempa#1%
91     #2%
92     \ifx\@tempa#1%
93       \endgroup
94       \let\caption@CheckCommand\@gobbletwo
95     \else
96       \endgroup
97     \fi}
98 \@onlypreamble\caption@DoCheckCommand

```

```

99 \let\caption@CheckCommand\caption@DoCheckCommand
100 \@onlypreamble\caption@CheckCommand

101 \newcommand*\caption@ifCheckCommand{%
102   \ifx\caption@CheckCommand\@gobbletwo
103     \let\next\@firstoftwo
104   \else
105     \let\next\@secondoftwo
106   \fi
107   \let\caption@CheckCommand\caption@DoCheckCommand
108   \next}
109 \@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.

```

110 \let\caption@begindocumenthook\@empty
111 \let\caption@@begindocumenthook\@empty

112 \def\caption@AtBeginDocument{%
113   \caption@teststar\g@addto@macro
114     \caption@@begindocumenthook\caption@begindocumenthook}
115 \@onlypreamble\caption@AtBeginDocument

116 \AtBeginDocument{%
117   \PackageInfo{caption}{Begin \noexpand\AtBeginDocument code\@gobble}%

118   \def\caption@AtBeginDocument{%
119     \ifstar{\g@addto@macro\caption@@begindocumenthook}\@firstofone}%
120   \caption@begindocumenthook
121   \let\caption@begindocumenthook\@undefined

122   \def\caption@AtBeginDocument{%
123     \@ifstar\@firstofone\@firstofone}%
124   \caption@@begindocumenthook
125   \let\caption@@begindocumenthook\@undefined

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

```

### 1.3 Errors

`\caption@Error` `\caption@Error{<message>}`

This is mainly identical to `\PackageError{caption}{<message>}{\caption@eh}`.

```

127 \newcommand*\caption@Error[1]{%
128   \PackageError\caption@Package{#1}\caption@eh}

129 \newcommand*\caption@Package{caption}

```

`\caption@eh` At the moment we only offer this simple error message as generic helper for the user.

```

130 \newcommand*\caption@eh{%
131   If you do not understand this error, please take a closer look\MessageBreak
132   at the documentation of the '\caption@Package' package.\MessageBreak
133   \@ehc}

```

## 1.4 Using the keyval package

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

```
134 \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.

```
135 \providecommand*\undefine@key[2]{%
136   \@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>.

```
137 \providecommand*\@onlypreamble@key[2]{%
138   \expandafter\@onlypreamble\csname KV@#1@#2\endcsname}
139 \@onlypreamble\@onlypreamble@key
```

```
\DeclareCaptionOption \DeclareCaptionOption{<option>}{<code>}
\DeclareCaptionOption*{<option>}{<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.

```
140 \newcommand*\DeclareCaptionOption{%
141   \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
142 \@onlypreamble\DeclareCaptionOption
143 \newcommand*\caption@declareoption[2]{%
144   #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
145 \@onlypreamble\caption@declareoption
```

```
\captionsetup \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`.

```
146 \newcommand*\captionsetup{%
147   \caption@teststar\@captionsetup\@gobble\@firstofone}
148 \newcommand*\@captionsetup[1]{%
149   \@ifnextchar[{\caption@setup@options#1}\caption@setup}
150 \newcommand*\caption@setup{\caption@setkeys{caption}}
151 \def\caption@setup@options#1[#2]{%
152   % #1 is \@gobble in star form, and \@firstofone otherwise
153   #1{\caption@addtooptlist{#2}}%
154   \expandafter\caption@l@addto@list\csname caption@opt@#2\endcsname}
```

```
\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 a warning.

```
155 \newcommand*\clearcaptionsetup{%
156   \caption@teststar\@clearcaptionsetup\@gobble\@firstofone}
157 \newcommand*\@clearcaptionsetup[1]{%
158   \let\caption@tempa#1%
159   \@testopt\@clearcaptionsetup{}}
```

```

160 \def\@@clearcaptionsetup[#1]#2{%
161   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
162   {\caption@tempa{\PackageWarning{caption}{Option list `#2' undefined}}}%
163   {\ifx,#1,%
164     \caption@clearsetup{#2}%
165   \else
166     \caption@@removefromsetup{#1}{#2}%
167   \fi}}

168 \newcommand*\caption@clearsetup[1]{%
169   \caption@removefromoptlist{#1}%
170   \@nameundef{caption@opt@#1}}

171 \newcommand*\caption@removefromsetup{%
172   \let\caption@tempa\@gobble
173   \caption@@removefromsetup}

174 \newcommand*\caption@@removefromsetup[2]{%
175   \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
176   \expandafter\let\csname caption@opt@#2\endcsname\@undefined
177   \def\@tempb##1=##2\@nil{##1}%
178   \edef\@tempc{#1}%
179   \@for\@tempa:=\@tempa\do{%
180     \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
181     \ifx\@tempd\@tempc
182       \let\caption@tempa\@gobble
183     \else
184       \expandafter\expandafter\expandafter\caption@l@addto@list
185       \expandafter\csname caption@opt@#2\endcsname\expandafter\endcsname
186       \expandafter{\@tempa}%
187     \fi}%
188   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
189   {\caption@removefromoptlist{#2}}}%
190   \caption@tempa{%
191     \PackageWarning{caption}{%
192       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>*.

```

193 \newcommand*\showcaptionsetup[2][\@firstofone]{%
194   \GenericWarning{ }{%
195     #1 Caption Info: KV list on `#2'\MessageBreak
196     #1 Caption Data: \@ifundefined{caption@opt@#2}{%
197       -none-%
198     }{%
199       {\expandafter\expandafter\expandafter\strip@prefix
200         \expandafter\meaning\csname caption@opt@#2\endcsname}%
201     }}

202 \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.

```

203 \newcommand*\caption@setoptions[1]{%
204   \caption@Debug{options=#1}%
205   \expandafter\let\expandafter\@tempa\csname caption@opt@#1\endcsname
206   \ifx\@tempa\relax \else
207     \let\caption@iftypewarning\@firstoftwo % = true
208     \expandafter\caption@setup\expandafter{\@tempa}%
209     \caption@clearsetup{#1}%
210   \fi}

\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”).

211 \newcommand*\caption@addtooptlist[1]{%
212   \ifundefined{caption@opt@#1@lineno}{%
213     \caption@dooptlist\caption@g@addto@list{#1}%
214     \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
215   }{}}

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

219 \newcommand*\caption@dooptlist[2]{%
220   \begingroup
221     \edef\@tempa{#2}\@onelevel@sanitize\@tempa
222     \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
223   \endgroup}

224 \AtEndDocument{%
225   \caption@for@list\caption@optlist{%
226     \PackageWarningNoLine{caption}{%
227       Unused \string\captionsetup[1]
228       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.

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

230 \long\def\caption@@setkeys[#1]#2#3{%
231   \ifx\caption@keydepth\@empty
232     \let\caption@ORI@KV@errx\KV@errx
233     \let\caption@ORI@KV@err\KV@err
234     \let\KV@errx\caption@Error
235     \let\KV@err\caption@Error
236   \fi
237   \l@addto@macro\caption@keydepth.%

238   \def\caption@Package{#1}%
239   \caption@Debug{\protect\setkeys{#2}{#3}}%
240   \setkeys{#2}{#3}%
241   \def\caption@Package{caption}%

```

```

242 \edef\caption@keydepth{\expandafter\@cdr\caption@keydepth\@nil}%
243 \ifx\caption@keydepth\@empty
244   \let\KV@errx\caption@ORI@KV@errx
245   \let\KV@err\caption@ORI@KV@err
246 \fi}
247 \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.

248 \newcommand*\caption@ExecuteOptions[2]{%
249   \@expandtwoargs\caption@setkeys{#1}{#2}}%
250 \@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.)

251 \newcommand*\caption@ProcessOptions{%
252   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
253 \@onlypreamble\caption@ProcessOptions

254 \newcommand*\caption@@ProcessOptions[2]{%
255   \let\@tempc\relax
256   \let\caption@tempa\@empty
257   #1{% \@firstofone -or- \@gobble
258     \for\CurrentOption:=\@classoptionslist\do{%
259       \ifundefined{KV@#2@\CurrentOption}}{%
260         \ifundefined{KV@#2@\CurrentOption @default}}{%
261           \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
262         }%
263         \PackageInfo{#2}{Global option '\CurrentOption' processed}%
264         \edef\caption@tempa{\caption@tempa,\CurrentOption,%
265           \@expandtwoargs\@removeelement\CurrentOption
266           \@unusedoptionlist\@unusedoptionlist
267         }%
268       }%
269     }%
270     \let\CurrentOption\@empty
271   }%
272   \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
273   \AtEndOfPackage{\let\@unprocessedoptions\relax}
274 \@onlypreamble\caption@@ProcessOptions

```

## 1.5 Margin resp. width

`\captionmargin` and `\captionwidth` contain the extra margin resp. the total width used for captions. Please never set these values in a direct way, they are just accessible in user documents to provide compatibility to `caption.sty v1.x`. Note that we can only set one value at a time, ‘margin’ or ‘width’. If `\captionwidth` is not zero we will take this value afterwards, otherwise `\captionmargin` and `\captionmargin@`.

```

275 \newdimen\captionmargin

```

```

276 \newdimen\captionmargin@
277 \newdimen\captionwidth

278 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
279 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
280 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
281 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}

282 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
283 \DeclareCaptionOption{maxmargin}{\caption@setoptcmd\caption@maxmargin{#1}}

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

284 \newcommand*\setcaptionmargin{%
285   \caption@teststar\caption@setmargin\@gobble\@firstofone}

286 \newcommand*\caption@setmargin[2]{%
287   #1{\captionwidth\z@}%
288   \caption@@setmargin#2,#2,\@nil}

289 \def\caption@@setmargin#1,#2,#3\@nil{%
290   \setlength\captionmargin@{#2}%
291   \setlength\captionmargin{#1}%
292   \addtolength\captionmargin@{-\captionmargin}}

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

293 \newcommand*\setcaptionwidth{%
294   \captionmargin\z@
295   \captionmargin@\z@
296   \setlength\captionwidth}

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

297 \newcommand*\caption@count{0}

\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.)

298 \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.

299 \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.

300 \newcommand*\caption@label[1]{%
301   \protected@write\@auxout{\let\caption@thepage\relax}%
302   {\string\caption@newlabel{#1}{\caption@thepage}}

```

`\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`.

```

303 \newcommand*\caption@pageref[1]{%
304   \expandafter\ifx\csname caption@r@#1\endcsname\relax
305     \G@refundefinedtrue % => 'There are undefined references.'
306     \PackageWarning{caption}{Reference on page \thepage \space undefined}%
307   \else
308     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
309   \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.

```

310 \newcommand*\caption@ifoddpage{%
311   \caption@iftwoside{%
312     \begingroup
313       \@tempcnta\caption@count
314       \advance\@tempcnta\@ne
315       \xdef\caption@count{\the\@tempcnta}%
316     \endgroup
317     \caption@label\caption@count
318     \caption@pageref\caption@count
319     \ifodd\caption@thepage
320       \let\caption@ifoddpage\@firstoftwo
321     \else
322       \let\caption@ifoddpage\@secondoftwo
323     \fi
324   }{\let\caption@ifoddpage\@firstoftwo}%
325   \caption@ifoddpage}

```

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

```

326 \newcommand*\caption@setoptcmd[2]{%
327   \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` 328 \newdimen\caption@indent

`\caption@hangindent` 329 \newdimen\caption@parindent

330 \newdimen\caption@hangindent

```

331 \DeclareCaptionOption{indent}[\leftmargini]{%
332   \setlength\caption@indent{#1}}% obsolete!
333 \DeclareCaptionOption{indentation}[\leftmargini]{%
334   \setlength\caption@indent{#1}}
335 \DeclareCaptionOption{parindent}{%
336   \setlength\caption@parindent{#1}}
337 \DeclareCaptionOption{hangindent}{%
338   \setlength\caption@hangindent{#1}}
339 \DeclareCaptionOption{parskip}{%
340   \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>

```

341 \ifundefined{scr@caption}{}{%
342   \let\caption@KV@parindent\KV@caption@parindent
343   \DeclareCaptionOption{parindent}[]{%
344     \ifx,#1,%
345       \caption@Debug{Option 'parindent' ignored}%
346     \else
347       \caption@KV@parindent{#1}%
348     \fi}%
349   \let\caption@KV@parskip\KV@caption@parskip
350   \DeclareCaptionOption{parskip}[]{%
351     \ifx,#1,%
352       \caption@Debug{Option 'parskip' ignored}%
353     \else
354       \caption@KV@parskip{#1}%
355     \fi}%
356 }

```

## 1.7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
357 \newcommand*\DeclareCaptionStyle[1]{%
358   \@testopt{\caption@declarestyle{#1}}{}%
359   \@onlypreamble\DeclareCaptionStyle
360   \def\caption@declarestyle#1[#2]#3{%
361     \global\@namedef{caption@sls@#1}{#2}%
362     \global\@namedef{caption@sty@#1}{#3}%
363     \@onlypreamble\caption@declarestyle
364   }
364   \DeclareCaptionOption{style}{\caption@setstyle{#1}}
365   \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.

```

366 \newcommand*\caption@setstyle{%
367   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

```

---

<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 otherwise would still collide with the current version 1.3 of the subfig package (Sigh!)

```

368 \newcommand*\caption@@setstyle[2]{%
369   \@ifundefined{caption@sty@#2}%
370     {#1{\caption@Error{Undefined style `#2'}}}%
371     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
372       \ifx\caption@setstyle@flag\@undefined
373         \let\caption@setstyle@flag\caption@setstyle
374         \let\caption@iftypewarning\@firstoftwo % = true
375         \caption@resetstyle
376         \expandafter\caption@setup\expandafter{\caption@sty}%
377         \let\caption@setstyle@flag\@undefined
378       }else
379         \expandafter\caption@setup\expandafter{\caption@sty}%
380       \fi
381     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
382     \expandafter\caption@l@addto@list\expandafter\caption@opt@SingleLine
383     \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!*

```

384 \newcommand*\caption@resetstyle{%
385   \caption@setup{%
386     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
387     justification=justified,font=,labelfont=,textfont=,%
388     margin=0pt,minmargin=0,maxmargin=0,%
389     indent=0pt,parindent=0pt,hangindent=0pt,%
390     singlelinecheck=1,strut=1}%
391   \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.

```

392 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
393 \DeclareCaptionStyle{default}[indent=0pt,justification=centering,format=plain]{%
394   format=default,labelsep=default,textformat=default,%
395   justification=default,font=default,labelfont=default,textfont=default}

```

## 1.8 Formats

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

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

```

396 \newcommand\DeclareCaptionFormat{%
397   \caption@teststar\caption@declareformat\@gobble\@firstofone}
398 \@onlypreamble\DeclareCaptionFormat

399 \newcommand\caption@declareformat[3]{%
400   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
401   \global\long\@namedef{caption@fmt@#2}##1##2##3{#3}}
402 \@onlypreamble\caption@declareformat

403 \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`).

```
404 \newcommand*\caption@setformat[1]{%
405   \ifundefined{caption@fmt@#1}%
406     {\caption@Error{Undefined format '#1'}}%
407     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
408       \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}
```

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

```
409 \DeclareCaptionFormat{plain}{#1#2#3\par}

410 \DeclareCaptionFormat{hang}{%
411   \caption@ifin@list\caption@lsepclist\caption@lsepname
412   {\caption@Error{%
413     The option 'labelsep=\caption@lsepname' does not work\MessageBreak
414     with 'format=hang'}}%
415   {\@hangfrom{#1#2}%
416     \advance\caption@parindent\hangindent
417     \advance\caption@hangindent\hangindent
418     \caption@@par#3\par}}
```

‘default’ usually maps to ‘plain’.

```
419 \def\caption@fmt@default{\caption@fmt@plain}
420 \def\caption@ifh@default{\caption@ifh@plain}
```

## 1.9 Label formats

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

421 \newcommand*\DeclareCaptionLabelFormat[2]{%
422   \global\@namedef{caption@lfmt@#1}##1##2##3{#2}}
423 \onlypreamble\DeclareCaptionLabelFormat

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

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

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

```
425 \newcommand*\caption@setlabelformat[1]{%
426   \ifundefined{caption@lfmt@#1}%
427     {\caption@Error{Undefined label format '#1'}}%
428     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}
```

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

```
429 \DeclareCaptionLabelFormat{empty}{}
430 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
431 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace} (#2)}
```

‘default’ usually maps to ‘simple’.

```
432 \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.
433 \newcommand\DeclareCaptionLabelSeparator{%
434   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
435 \@onlypreamble\DeclareCaptionLabelSeparator
436 \newcommand\caption@declarelabelseparator[3]{%
437   \global\expandafter\let\csname caption@iflf@#2\endcsname#1%
438   \global\long\@namedef{caption@lsep@#2}{#3}%
439   \caption@@declarelabelseparator{#2}#3\\@nil}
440 \@onlypreamble\caption@declarelabelseparator
441 \long\def\caption@@declarelabelseparator#1#2\\#3@nil{%
442   \def\@tempa{#3}\ifx\@tempa\@empty \else
443     \caption@gaddto@list\caption@lsep@crlist{#1}%
444   \fi}
445 \@onlypreamble\caption@@declarelabelseparator
446 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
447 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

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

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

```
448 \newcommand*\caption@setlabelseparator[1]{%
449   \@ifundefined{caption@lsep@#1}%
450   {\caption@Error{Undefined label separator `#1'}}%
451   {\edef\caption@lsepname{#1}%
452     \expandafter\let\expandafter\caption@iflf\csname caption@iflf@#1\endcsname
453     \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’.

```
454 \DeclareCaptionLabelSeparator{none}{}
455 \DeclareCaptionLabelSeparator{colon}{: }
456 \DeclareCaptionLabelSeparator{period}{. }
457 \DeclareCaptionLabelSeparator{space}{ }
458 \DeclareCaptionLabelSeparator*{quad}{\quad}
459 \DeclareCaptionLabelSeparator*{newline}{\\}
460 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}
```

‘default’ usually maps to ‘colon’.

```
461 \def\caption@lsep@default{\caption@lsep@colon}
462 \def\caption@iflf@default{\caption@iflf@colon}
```

## 1.11 Text formats

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

```

466 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
467 \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`).

```

468 \newcommand*\caption@settextformat[1]{%
469   \ifundefined{caption@tfmt@#1}%
470     {\caption@Error{Undefined text format `#1'}}%
471     {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

472 \DeclareCaptionTextFormat{simple}{#1}
473 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

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

```

## 1.12 Fonts

```

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

```

```

DeclareDefaultCaptionFont \DeclareDefaultCaptionFont{<name>}{<code>}
478 \newcommand*\DeclareDefaultCaptionFont[2]{%
479   \global\@namedef{caption#1@default}{#2}}
480 \@onlypreamble\DeclareDefaultCaptionFont

481 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
482 \DeclareDefaultCaptionFont{font}{}

483 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
484 \DeclareDefaultCaptionFont{labelfont}{}

485 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
486 \DeclareDefaultCaptionFont{textfont}{}

```

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

Selecting a caption font means saving all the code snippets (in `\caption#1`). Because we use `\setkeys` recursive here we need to do this inside an extra group and collect all the code snippets in `\caption@tempa` first.

```

487 \newcommand*\caption@setfont[2]{%
488   \begingroup
489     \let\caption@fnt\@empty
490     \define@key{caption@fnt}{default}[]{%
491       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
492     \caption@setkeys[caption]{caption@fnt}{#2}%
493     \global\let\caption@tempa\caption@fnt
494   \endgroup
495   \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`.

```

496 \newcommand*\caption@font{%
497   \caption@teststar\caption@@font\@firstofone
498     {\caption@setkeys[caption]{caption@fnt}}}%
499 \newcommand*\caption@@font[2]{%
500   \begingroup
501   \def\caption@fnt{\endgroup}%
502   #1{#2}%
503   \caption@fnt}

```

These are the pre-defined font code snippets.

```

504 \DeclareCaptionFont{normalcolor}{\normalcolor}
505 \DeclareCaptionFont{color}{\color{#1}}

506 \DeclareCaptionFont{normalfont}{\normalfont}
507 \DeclareCaptionFont{up}{\upshape}
508 \DeclareCaptionFont{it}{\itshape}
509 \DeclareCaptionFont{sl}{\slshape}
510 \DeclareCaptionFont{sc}{\scshape}
511 \DeclareCaptionFont{md}{\mdseries}
512 \DeclareCaptionFont{bf}{\bfseries}
513 \DeclareCaptionFont{rm}{\rmfamily}
514 \DeclareCaptionFont{sf}{\sffamily}
515 \DeclareCaptionFont{tt}{\ttfamily}

516 \DeclareCaptionFont{scriptsize}{\scriptsize}
517 \DeclareCaptionFont{footnotesize}{\footnotesize}
518 \DeclareCaptionFont{small}{\small}
519 \DeclareCaptionFont{normalsize}{\normalsize}
520 \DeclareCaptionFont{large}{\large}
521 \DeclareCaptionFont{Large}{\Large}

522 \DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespacing}% normally 1
523 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
524 \DeclareCaptionFont{doublespacing}{\doublespacing}
525 \DeclareCaptionFont{stretch}{\setstretch{#1}}
526 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }

527 %\DeclareCaptionFont{normal}{%
528 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
529 %\DeclareCaptionFont{normal}{%
530   \caption@font*{%
531     \KV@caption@fnt@normalcolor\@unused
532     \KV@caption@fnt@normalfont\@unused
533     \KV@caption@fnt@normalsize\@unused
534     \KV@caption@fnt@singlespacing\@unused}}

```

`\captionsize` 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.

```

535 \providecommand\captionsize{}

536 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
537 \DeclareDefaultCaptionFont{size}{}

```

## 1.13 Justifications

```

\DeclareCaptionJustification <name>{<code>}
538 \newcommand*\DeclareCaptionJustification[2]{%
539   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
540   \DeclareCaptionFont{#1}{#2}}
541 \@onlypreamble\DeclareCaptionJustification

\DeclareDefaultCaptionJustification{<code>}
542 \newcommand*\DeclareDefaultCaptionJustification[1]{%
543   \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
544   \DeclareDefaultCaptionFont{@hj}{#1}}
545 \@onlypreamble\DeclareDefaultCaptionJustification

546 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
547 \DeclareDefaultCaptionJustification{}

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

```

These are the pre-defined justification code snippets.

```

549 \DeclareCaptionJustification{justified}{}
550 \DeclareCaptionJustification{centering}{\centering}
551 \DeclareCaptionJustification{centerfirst}{\centerfirst}
552 \DeclareCaptionJustification{centerlast}{\centerlast}
553 \DeclareCaptionJustification{raggedleft}{\raggedleft}
554 \DeclareCaptionJustification{raggedright}{\raggedright}

\centerfirst Please blame Frank Mittelbach for the code of \caption@centerfirst :-)
555 \providecommand\centerfirst{%
556   \let\\\@centercr
557   \edef\caption@normaladjust{%
558     \leftskip\the\leftskip
559     \rightskip\the\rightskip
560     \parfillskip\the\parfillskip\relax}%
561   \leftskip\z@\@plus -1fil%
562   \rightskip\z@\@plus 1fil%
563   \parfillskip\z@skip
564   \noindent\hskip\z@\@plus 2fil%
565   \@setpar{\@@par\@restorepar\caption@normaladjust}}

\centerlast This is based on code from Anne Brüggemann-Klein[22]
566 \providecommand\centerlast{%
567   \let\\\@centercr
568   \leftskip\z@\@plus 1fil%
569   \rightskip\z@\@plus -1fil%
570   \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.

```

571 \DeclareCaptionJustification{Centering}{%
572   \caption@ragged\Centering\centering}
573 \DeclareCaptionJustification{RaggedLeft}{%
574   \caption@ragged\RaggedLeft\raggedleft}
575 \DeclareCaptionJustification{RaggedRight}{%
576   \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.

```

577 \newcommand*\caption@ragged{%
578   \caption@Debug{We need ragged2e}%
579   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
580   \global\let\caption@ragged\caption@@ragged
581   \caption@ragged}

582 \caption@AtBeginDocument{%
583   \@ifundefined{caption@r@ragged2e}{%
584     \newcommand*\caption@@ragged{%
585       \PackageWarning{caption}{%
586         'ragged2e' support has been changed.\MessageBreak
587         Rerun to get captions right}%
588       \global\let\caption@ragged\@secondoftwo % suppress further warnings
589       \caption@ragged}%
590   }{%
591     \caption@Debug{We load ragged2e}%
592     \IfFileExists{ragged2e.sty}{%
593       \RequirePackage{ragged2e}%
594       \let\caption@@ragged\@firstoftwo
595     }{%
596       \newcommand*\caption@@ragged[2]{%
597         \@ifundefined{caption\string#1}{%
598           \PackageWarning{caption}{%
599             'ragged2e' package not loaded, therefore\MessageBreak
600             substituting \string#2 for \string#1\MessageBreak}%
601           \global\@namedef{caption\string#1}{}%
602           #2}%
603     }%
604   }}

```

## 1.14 Vertical spaces before and after captions

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

```

605 \@ifundefined{abovecaptionskip}{%
606   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
607 \@ifundefined{belowcaptionskip}{%
608   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

```

```

609 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
610 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
611 \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.

```

612 \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.*

```

613 \newcommand*\caption@setposition[1]{%
614   \caption@ifinlist{#1}{d,default}{%
615     \let\caption@position\caption@defaultpos
616   }{\caption@ifinlist{#1}{t,top,above}{%
617     \let\caption@position\@firstoftwo
618   }{\caption@ifinlist{#1}{b,bottom,below}{%
619     \let\caption@position\@secondoftwo
620   }{\caption@ifinlist{#1}{a,auto}{%
621     \let\caption@position\@undefined
622   }{%
623     \caption@Error{Undefined position `#1'}%
624   }}}}%

```

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

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

```

625 %\caption@setdefaultpos{a}% default = auto
626 \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.)

```

627 \newcommand*\caption@iftop{%
628   \ifx\caption@position\@undefined
629     \let\caption@position\@secondoftwo
630   %   = \caption@setposition b%
631   \fi
632   \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.

```

633 \newcommand*\caption@fixposition{%
634   \ifx\caption@position\@undefined

```

```

635 \caption@autoposition
636 \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.

```

637 \newcommand*\caption@autoposition{%
638   \ifvmode
639     \edef\caption@tempa{\the\prevdepth}%
640     \caption@Debug{\protect\prevdepth=\caption@tempa}%
641     \ifdim\prevdepth>-\p@
642       \let\caption@position\@secondoftwo
643     \else
644       \let\caption@position\@firstoftwo
645     \fi
646 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
647   \else
648     \caption@Debug{no \protect\prevdepth}%
649     \let\caption@position\@secondoftwo
650 %   = \caption@setposition b%
651   \fi}

```

```
\caption@setautoposition \caption@setautoposition{<position>}
```

replaces the above algorithm by a different one (or a fixed position setting).

```

652 \newcommand*\caption@setautoposition[1]{%
653   \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`.

```

654 \newcommand*\caption@beginhook{}
655 \newcommand*\caption@endhook{}
656 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
657 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

## 1.17 Lists

```

658 \DeclareCaptionOption{listof}{\caption@set@bool\caption@iflof{#1}}
\DeclareCaptionListFormat{<name>}{<code with #1 and #2>}
659 \newcommand*\DeclareCaptionListFormat[2]{%
660   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
661 \@onlypreamble\DeclareCaptionListFormat

662 \DeclareCaptionOption{listofformat}{\caption@setlistofformat{#1}}

```

```
\caption@setlistofformat \caption@setlistofformat{<name>}
Selecting a caption list-of format simply means saving the code (in \caption@lstfmt).
663 \newcommand*\caption@setlistofformat[1]{%
664   \@ifundefined{caption@lstfmt@#1}%
665   {\caption@Error{Undefined list-of format `#1'}}%
666   {\expandafter\let\expandafter\caption@lstfmt
667     \csname caption@lstfmt@#1\endcsname}}
```

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

```
668 \DeclareCaptionListOfFormat{empty}{}
669 \DeclareCaptionListOfFormat{simple}{#1#2}
670 \DeclareCaptionListOfFormat{parens}{#1(#2)}
671 \DeclareCaptionListOfFormat{subsimple}{#2}
672 \DeclareCaptionListOfFormat{subparens}{(#2)}
673 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}
```

1.18 Debug option

```
674 \DeclareCaptionOption{debug}[1]{%
675   \caption@set@bool\caption@ifdebug{#1}%
676   \let\caption@iftypewarning\caption@ifdebug
677   \caption@ifdebug
678   {\def\caption@Debug{\PackageInfo{caption}}}%
679   {\let\caption@Debug@gobble}}
680 \DeclareOption{debug}{\setkeys{caption}{debug}}
681 \setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

```
682 \caption@CheckCommand\@makecaption{\let\@makecaption\@undefined}
```

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

```
683 \caption@CheckCommand\@makecaption{%
684   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
685   \long\def\@makecaption#1#2{%
686     \vskip\abovecaptionskip
687     \sbox\@tempboxa{#1: #2}%
688     \ifdim \wd\@tempboxa >\hsize
689       #1: #2\par
690     \else
691       \global \@minipagefalse
692       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
693     \fi
694     \vskip\belowcaptionskip}}
```

1.19.2 The A<sub>M</sub>S & SMF classes

```
695 \@ifundefined{captionheadfont}{}{}%
696 \caption@CheckCommand\@makecaption{%
697   % amsart|amsproc|amsbook [2004/08/06 v2.20]
698   \long\def\@makecaption#1#2{%
699     \setbox\@tempboxa\vbox{\color@setgroup
700       \advance\hsize-2\captionindent\noindent
```

```

701         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
702         {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
703         \unskip\kern-2\captionindent\par
704         \global\setbox\@ne\lastbox\color@endgroup}%
705     \ifhbox\@ne % the normal case
706         \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
707     \fi
708     \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
709         \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
710     \else % tempboxa contained more than one line
711         \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
712             \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
713     \fi
714     \ifnum\@tempcnta<64 % if the float IS a figure...
715         \addvspace\abovecaptionskip
716         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
717     \else % if the float IS NOT a figure...
718         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
719         \nobreak
720         \vskip\belowcaptionskip
721     \fi
722     \relax
723 }}

724 \caption@CheckCommand\@makecaption{%
725     % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
726     \long\def\@makecaption#1#2{%
727         \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
728         \setbox\@tempboxa\vbox{\color@setgroup
729             \advance\hsize-2\captionindent\noindent
730             \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
731             {\@cdr#2\@nil}{.\@addpunct{.}\@captionfont\upshape\enspace#2}%
732             \unskip\kern-2\captionindent\par
733             \global\setbox\@ne\lastbox\color@endgroup}%
734         \ifhbox\@ne % the normal case
735             \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
736         \fi
737         \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
738             \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
739             \@tempdima\wd\@ne\advance\@tempdima-\captionindent
740             \wd\@ne\@tempdima
741         \else % tempboxa contained more than one line
742             \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
743                 \unvbox\@tempboxa\parskip\z@skip
744                 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
745         \fi
746         \ifnum\@tempcnta<64 % if the float IS a figure...
747             \addvspace\abovecaptionskip
748             \noindent\kern\captionindent\box\@ne
749         \else % if the float IS NOT a figure...
750             \noindent\kern\captionindent\box\@ne
751             \nobreak
752             \vskip\belowcaptionskip
753         \fi
754     \relax

```

```

755     }}
756 \let\captionmargin\captionindent
757 \begingroup\edef\@tempa{\endgroup
758   \noexpand\caption@g@addto@list\noexpand\caption@sty@default
759     {margin=\the\captionmargin
760       \@ifundefined{smf@makecaption}}{{},maxmargin=.1\linewidth}}
761 \@tempa
762 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
763 \DeclareCaptionLabelSeparator{default}{.\enspace}
764 \DeclareDefaultCaptionFont{font}{\@captionfont}
765 \DeclareDefaultCaptionFont{labelfont}{\@captionheadfont}
766 \DeclareDefaultCaptionFont{textfont}{\@captionfont\upshape}
767 \captionsetup*[figure]{position=b}
768 \captionsetup*[table]{position=t}
769 }

```

### 1.19.3 The beamer class

```

770 \@ifclassloaded{beamer}{%
771   \caption@CheckCommand\beamer@makecaption{%
772     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
773     \long\def\beamer@makecaption#1#2{%
774       \def\insertcaptionname{\csname#1name\endcsname}%
775       \def\insertcaptionnumber{\csname the#1\endcsname}%
776       \def\insertcaption{#2}%
777       \nobreak\vskip\abovecaptionskip\nobreak
778       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
779       \ifdim \wd\@tempboxa >\hsize
780         \usebeamertemplate**{caption}\par
781       \else
782         \global \@minipagefalse
783         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
784       \fi
785       \nobreak\vskip\belowcaptionskip\nobreak}}
786 \DeclareCaptionLabelFormat{default}{#1}
787 \DeclareDefaultCaptionJustification{\raggedright}
788 \DeclareDefaultCaptionFont{font}{%
789   \usebeamerfont*{caption}%
790   \usebeamercolor[fg]{caption}}
791 \DeclareDefaultCaptionFont{labelfont}{%
792   \usebeamercolor[fg]{caption name}%
793   \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.)

```

794 \defbeamertemplate{caption}{caption3}{%
795   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
796 \DeclareOption{beamer}{%
797   % \usebeamertemplate**{caption} will set font
798   \DeclareDefaultCaptionFont{font}{}%
799   \setbeamertemplate{caption}[caption3]}
800 {}

```

### 1.19.4 The KOMA-Script classes

```
801 \@ifundefined{scr@caption}{}{%
802   \caption@CheckCommand\@makecaption{%
803     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
804     \long\def\@makecaption#1#2{%
805       \if@captionabove
806         \vskip\belowcaptionskip
807       \else
808         \vskip\abovecaptionskip
809       \fi
810       \@@makecaption\@firstofone{#1}{#2}%
811       \if@captionabove
812         \vskip\abovecaptionskip
813       \else
814         \vskip\belowcaptionskip
815       \fi}}
816 \DeclareCaptionFormat{default}{%
817   \ifdofullc@p
818     \caption@ifin@list\caption@lsep@crlist\caption@lsep@name
819     {\caption@Error{%
820       The option 'labelsep=\caption@lsep@name' does not work\MessageBreak
821       with \noexpand\setcaphanging (which is set by default)}}%
822     {\caption@fmt@hang{#1}{#2}{#3}}%
823   \else
824     #1#2%
825     \ifdim\cap@indent<\z@
826       \par
827       \noindent\hspace*{-\cap@indent}%
828     \else\if@capbreak
829       \par
830     \fi\fi
831     #3\par
832   \fi}
833 \DeclareCaptionLabelSeparator{default}{\captionformat}
834 \DeclareDefaultCaptionFont{font}{\scr@fnt@caption}
835 \DeclareDefaultCaptionFont{labelfont}{\scr@fnt@captionlabel}
836 }
```

### 1.19.5 The NTG Dutch classes

```
837 \@ifundefined{CaptionFonts}{}{%
838   \caption@CheckCommand\@makecaption{%
839     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
840     \long\def\@makecaption#1#2{%
841       \vskip\abovecaptionskip
842       \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
843       \ifdim \wd\@tempboxa >\hsize
844         {\CaptionLabelFont#1:} \CaptionTextFont#2\par
845       \else
846         \global \@minipagefalse
847         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
848       \fi
849       \vskip\belowcaptionskip}}}
```

```

850 \DeclareDefaultCaptionFont{labelfont}{\CaptionLabelFont}
851 \DeclareDefaultCaptionFont{textfont}{\CaptionTextFont}
852 }

```

### 1.19.6 The frenchb Babel option

```

853 \@ifundefined{FB@makecaption}{}{%
854 \caption@CheckCommand\@makecaption{%
855 % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
856 \long\def\@makecaption#1#2{%
857 \vskip\abovecaptionskip
858 \sbox\@tempboxa{#1\CaptionSeparator #2}%
859 \ifdim \wd\@tempboxa >\hsize
860 #1\CaptionSeparator #2\par
861 \else
862 \global \@minipagefalse
863 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
864 \fi
865 \vskip\belowcaptionskip}}
866 \ifx\@makecaption\STD@makecaption
867 \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
868 \def\caption@frenchb{% supress frenchb warning
869 \let\STD@makecaption\@makecaption
870 \let\FB@makecaption\@makecaption}
871 \else
872 \ifx\@makecaption\@undefined\else
873 \PackageInfo{caption}{%
874 The definition of \protect\@makecaption\space
875 has been changed,\MessageBreak
876 frenchb will NOT customize it}%
877 \fi
878 \fi
879 }

```

### 1.19.7 The frenchle/pro package

```

880 \@ifundefined{frenchTeXmods}{}{%
881 \caption@CheckCommand\@makecaption{%
882 % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
883 % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
884 \def\@makecaption#1#2{%
885 \ifFTY%
886 \def\@secondofmany##1##2\void{##2}%
887 \def\@tempa{\@secondofmany#2\void}%
888 \ifx\@tempa\empty%
889 \let\captionseparator\empty%
890 \fi%
891 \@mcORI{#1}{\relax\captionfont{#2}}%
892 \else
893 \@mcORI{#1}{#2}%
894 \fi}}
895 \caption@CheckCommand\@makecaption{%
896 % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]

```

```

897 \def\@makecaption#1#2{%
898 \ifFTY%
899 \def\@secondofmany##1##2\void{##2}%
900 \protected@edef\@tempa{\@secondofmany#2\void}%
901 \ifx\@tempa\empty%
902 \let\captionseparator\empty%
903 \fi%
904 \@mcORI{#1}{\relax\captionfont{#2}}%
905 \else
906 \@mcORI{#1}{#2}%
907 \fi}}
908 \DeclareDefaultCaptionFont{textfont}{\itshape}%
909 \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
910 }

```

## 1.20 Execution of options

```

911 \captionsetup{style=default,position=default,listof=1,listofformat=default,%
912             twoside=\if@twoside 1\else 0\fi}
913 \ProcessOptions*
914 \caption@ifcheckcommand{%
915 \caption@setbool{documentclass}{1}%
916 }{%
917 \caption@setbool{documentclass}{0}%
918 \PackageInfo{caption}{%
919     Unknown document class (or package), \MessageBreak
920     standard defaults will be used}%
921 \caption@debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
922 }

```

## 1.21 Typesetting the caption

```

\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.

```

923 \newcommand*\caption@addcontentsline[2]{%
924 \caption@iflof%
925 {\def\@tempa{#2}}%
926 {\let\@tempa\@empty}%
927 \ifx\@tempa\@empty \else
928 {\let\\space
929 \addcontentsline{\csname ext@#1\endcsname}{#1}%
930                 {\protect\numberline
931                  {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
932                  {\ignorespaces #2}}}%
933 \fi}

```

`\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`.)

```

934 \newif\ifcaption@star

```

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

```

935 \newcommand*\caption@fnum[1]{%
936   \caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}{\theContinuedFloat}}

\caption@make \caption@make{\float name}{\ref. number}{\text}
Typesets the caption.
937 \newcommand\caption@make[2]{%
938   \caption@@make{\caption@lfmt{#1}{#2}}{\theContinuedFloat}}

\caption@@make \caption@@make{\caption label}{\caption text}
939 \newcommand\caption@@make[2]{%
940   \beginngroup
941   \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
942   \ifx\caption@maxmargin\undefined \else
943     \ifdim\captionmargin>\caption@maxmargin\relax
944       \captionmargin\caption@maxmargin\relax
945     \fi
946   \fi
947   \ifx\caption@minmargin\undefined \else
948     \ifdim\captionmargin<\caption@minmargin\relax
949       \captionmargin\caption@minmargin\relax
950     \fi
951   \fi

Special single-line treatment (option singlelinecheck=)
952   \caption@ifslc{\caption@ifempty@list\caption@opt@SingleLine}{\%
953     \caption@slc{#1}{#2}\captionwidth\relax}}{\%

Typeset the left margin (option margin=)
954   \caption@calcmargin
955   \@tempdima\captionmargin
956   \ifdim\captionmargin@=\z@ \else
957     \caption@ifoddpagel{\advance\@tempdima\captionmargin@}%
958   \fi
959   \caption@ifh{\advance\@tempdima\caption@indent}%
960   \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*.
961   \@tempdima\captionwidth
962   \caption@ifh{\advance\@tempdima-\caption@indent}%
963   \captionbox\@tempdima{%

Typeset the indentation (option indentation=)
Bugfix 04-05-05: \hskip-\caption@indent replaced by \ifdim\caption@indent=\z@...
964   \caption@ifh{%
965     \ifdim\caption@indent=\z@
966       \leavevmode
967     \else
968       \hskip-\caption@indent
969     \fi}%

Typeset the caption itself and close the \captionbox
970   \caption@@@make{#1}{#2}}%

```

Typeset the right margin (option margin=)

```

971 \@tempdima\captionmargin
972 \ifdim\captionmargin@=\z@ \else
973   \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
974   \fi
975   \hspace\@tempdima
976 \caption@endhook
977 \endgroup
978 \global\caption@starfalse}

```

\caption@calcmargin

\caption@calcmargin

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

```

979 \newcommand*\caption@calcmargin{%
980   \ifdim\captionwidth=\z@
981     \captionwidth\linewidth
982     \advance\captionwidth by -2\captionmargin
983     \advance\captionwidth by -\captionmargin@
984   \else
985     \captionmargin\linewidth
986     \advance\captionmargin by -\captionwidth
987     \divide\captionmargin by 2
988     \captionmargin@\z@
989   \fi
990   \caption@Debug{%
991     \string\hsize=\the\hsize,
992     \string\linewidth=\the\linewidth,\MessageBreak
993     \string\leftmargin=\the\leftmargin,
994     \string\rightmargin=\the\rightmargin,\MessageBreak
995     \string\margin=\the\captionmargin,
996     \string\margin@=\the\captionmargin@,
997     \string\width=\the\captionwidth}%
998 }

```

\caption@slc

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

This one does the single-line-check.

```

999 \newcommand\caption@slc[4]{%
1000   \caption@Debug{Begin SLC}%
1001   \begingroup
1002   \expandafter\caption@setup\expandafter{\caption@opt@SingleLine}%
1003   \let\caption@hj\@empty
1004   \caption@calcmargin % calculate #3 if necessary
1005   \caption@prepareslc
1006   \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1007   \ifdim\wd\@tempboxa>#3%
1008     \endgroup
1009   \else
1010     \endgroup
1011     \expandafter\caption@setup\expandafter{\caption@opt@SingleLine}%
1012     #4%
1013   \fi
1014   \caption@Debug{End SLC}}

```

```

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1015 \newcommand*\caption@prepareslc{%
1016 \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1017 \def\label{\caption@withoptargs\@gobbletwo}%
1018 \def\stepcounter##1{\advance\cname c@##1\endcname\@ne\relax}%
1019 \let\refstepcounter\stepcounter\let\H@refstepcounter\stepcounter}

\captionbox \captionbox{\langle width \rangle \{\langle contents \rangle\}}
This macro defines the box which surrounds the caption paragraph.
1020 \newcommand*\captionbox{\parbox[b]}

\caption@@@make \caption@@@make{\langle caption label \rangle \{\langle caption text \rangle\}}
This one finally typesets the caption paragraph, without margin and indentation.
1021 \newcommand\caption@@@make[2]{%
If the label is empty, we use no caption label separator.
1022 \sbox\@tempboxa{#1}%
1023 \ifdim\wd\@tempboxa=\z@
1024 \let\caption@lsep\relax
1025 % \@capbreakfalse
1026 \fi
If the text is empty, we use no caption label separator, too.
1027 \caption@ifempty{#2}{%
1028 \let\caption@lsep\relax
1029 % \@capbreakfalse
1030 % \let\caption@ifstrut\@secondoftwo
1031 }%
Take care that \caption@parindent and \caption@hangindent will be used
to typeset the paragraph.
1032 \@setpar{\@@par\caption@@par}\caption@@par
Finally typeset the caption.
1033 \caption@hj\captionsize\captionfont\caption@fmt
1034 {\ifcaption@star\else{\captionlabelfont#1}\fi}%
1035 {\ifcaption@star\else{\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1036 {\captiontextfont
1037 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1038 \nobreak\hskip\z@skip % enable hyphenation
1039 \caption@tfmt{#2}%
1040 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1041 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1042 \par}}

\caption@ifempty \caption@ifempty{\langle text \rangle \{\langle true \rangle\} (no \langle false \rangle)}
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.
1043 \newcommand\caption@ifempty[1]{%
1044 \def\caption@tempa{#1}%
1045 \def\caption@tempb{\ignorespaces}%

```

```

1046 \ifx\caption@tempa\caption@tempb
1047 \let\caption@tempa\@empty
1048 \fi
1049 \ifx\caption@tempa\@empty
1050 \let\caption@ifempty\@secondoftwo
1051 \else
1052 \let\caption@ifempty\@gobbletwo
1053 \fi
1054 \caption@ifempty\@unused}

\caption@@par \caption@@par
This command will be executed with every \par inside the caption.
1055 \newcommand*\caption@@par{%
1056 \parindent\caption@parindent\hangindent\caption@hangindent}%

```

## 1.22 Bonus features

```

\DeclareFloatingEnvironment [⟨file ext⟩] {⟨environment⟩} [⟨name⟩] [⟨list name⟩]
1057 \providecommand*\DeclareFloatingEnvironment{%
1058 \@ifnextchar[%]
1059 \caption@@DeclareFloatingEnvironment\caption@DeclareFloatingEnvironment}
1060 \@onlypreamble\DeclareFloatingEnvironment

1061 \newcommand*\caption@DeclareFloatingEnvironment[1]{%
1062 \caption@@DeclareFloatingEnvironment[lo#1]{#1}}
1063 \@onlypreamble\caption@DeclareFloatingEnvironment

1064 \def\caption@@DeclareFloatingEnvironment[#1]#2{%
1065 \ifundefined{c@chapter}%
1066 {\newcounter{#2}%
1067 \@namedef{the#2}{\arabic{#2}}}%
1068 \@namedef{theH#2}{\arabic{#2}}}%
1069 {\newcounter{#2}[chapter]%
1070 \@namedef{the#2}{\ifnum\c@chapter>\z@ \thechapter.\fi \arabic{#2}}}%
1071 \@namedef{theH#2}{\theHchapter.\arabic{#2}}}%
1072 \@namedef{fps@#2}{tbp}%
1073 \ifundefined{c@float@type}%
1074 {\newcounter{float@type}%
1075 \setcounter{float@type}{\ifundefined{c@figure}14}}%
1076 {}%
1077 \caption@Debug{float type `#2'=\the\value{float@type}}%
1078 \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1079 \addtocounter{float@type}{\value{float@type}}%
1080 \@namedef{ext@#2}{#1}%
1081 \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1082 \newenvironment{#2}{\@float{#2}}{\end@float}%
1083 \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1084 \expandafter\newcommand\csname listof#2s\endcsname{\begingroup
1085 \expandafter\let\expandafter\listfigurename\csname list#2name\endcsname
1086 \expandafter\let\expandafter\ext@figure\csname ext@#2\endcsname
1087 \let\caption@ORI@starttoc\@starttoc
1088 \renewcommand*\@starttoc[1]{%
1089 \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1090 \listoffigures
1091 \endgroup}%

```

```

1092 \@ifundefined{l@figure}%
1093   {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1094   {\expandafter\let\csname l@#2\endcsname l@figure}%
1095 \expandafter\newcommand\csname #2name\endcsname{}%
1096 \edef\@tempa{%
1097   \def\noexpand\@tempa{\expandafter\noexpand\@car#2\@nil}}%
1098 \uppercase\expandafter{\@tempa}%
1099 \edef\@tempb{%
1100   \noexpand\g@addto@macro\noexpand\@tempa{\expandafter\@cdr#2\@nil}}%
1101 \@tempb
1102 \expandafter\let\csname #2name\endcsname\@tempa
1103 \expandafter\newcommand\csname list#2name\endcsname{}%
1104 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1105 \@cons\caption@floatlist{{#2}}%
1106 \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#2}}\relax}
1107 \@onlypreamble\caption@@DeclareFloatingEnvironment
1108 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1109   \@namedef{#1name}{#2}%
1110   \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#1}}\relax}
1111 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1112 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1113   \@namedef{list#1name}{#2}}
1114 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1115 \newcommand*\caption@floatlist{}

```

```

\ForFloatingEnvironments \ForFloatingEnvironments{<code with #1>}
1116 \providecommand\ForFloatingEnvironments[1]{%
1117   \caption@AtBeginDocument{%
1118     \def\@elt##1{#1}%
1119     \caption@floatlist
1120     \let\@elt\relax}}%
1121 \@onlypreamble\ForFloatingEnvironments

```

### 1.23 subfig package adaption

Unfortunately the author of the subfig package does not seem to maintain his package anymore, so we have to make several adaption here.

```

1122 \caption@AtBeginDocument{%
1123   \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1124   \ifx\subref\@tempa
1125     \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1126     \let\caption@setfloattype\@gobble
1127 %   \let\sf@ifinlist\caption@ifinlist
1128   \let\sf@ifpositiontop\caption@iftop
1129   \def\@dottedxxxline#1#2#3#4#5#6#7{%
1130     \begingroup
1131     \caption@setfloattype{#1}%
1132     \caption@setoptions{subfloat}%
1133     \caption@setoptions{sub#1}%
1134     \ifnum #3>\@nameuse{c@#2depth}\else
1135       \@dottedtocline{\z@}{#4}{#5}{#6}{#7}%

```

```

1136         \fi
1137         \endgroup}%
1138         \let\sf@subfloat\sf@NEW@subfloat

```

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

```

1139         \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1140         \caption@removefromoptlist{subfloat}%
1141         \caption@removefromoptlist{subfigure}%
1142         \caption@removefromoptlist{subtable}%

```

```

1143     \fi
1144     \let\sf@NEW@subfloat\@undefined}

```

```

1145 \def\sf@NEW@subfloat{%
1146   \begingroup
1147     \caption@setfloattype\@capttype
1148     \sf@ifpositiontop{%
1149       \maincaptiontoptrue
1150     }{%
1151       \maincaptiontopfalse
1152     }%
1153     \caption@setoptions{subfloat}%
1154     \caption@setoptions{sub\@capttype}%
1155     \let\sf@oldlabel=\label
1156     \let\label=\subfloat@label
1157     \ifmaincaptiontop\else
1158       \advance\@nameuse{c@\@capttype}\@ne
1159     \fi
1160     \refstepcounter{sub\@capttype}%
1161     \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1162     \@ifnextchar [% %] match left bracket
1163       {\sf@@subfloat}%
1164       {\sf@@subfloat[\@empty]}}%

```

## 2 Main package

### 2.1 Identification

```
1165 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1166 \ProvidesPackage{caption}[2007/09/16 v3.1a Customising captions (AR)]

\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.

1167 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1168 \@onlypreamble\caption@Info
```

### 2.2 Loading the kernel

```
1169 \RequirePackage{caption3}[2007/09/16] % needs v3.1a or newer
```

### 2.3 Check against incompatible packages

```
1170 \@ifpackageloaded{caption2}{%
1171   \caption@Error{%
1172     You can't use both, the (obsolete) caption2 *and* \MessageBreak
1173     the (current) caption package}%
1174   \endinput
1175 }{}

1176 \caption@AtBeginDocument{%
1177   \@ifpackageloaded{ftcap}{%
1178     \caption@IncompatiblePackage{ftcap}{}%
1179   }{}%
1180   \@ifpackageloaded{nonfloat}{%
1181     \caption@IncompatiblePackage{nonfloat}{. %
1182       Furthermore you could use \string\captionof\MessageBreak
1183       instead of \noexpand\figcaption or \string\tabcaption}%
1184   }{}%
1185   \@ifpackageloaded{topcapt}{%
1186     \caption@IncompatiblePackage{topcapt}{}%
1187   }{}%
1188 }

\caption@Incompatible Issue an ‘incompatible’ warning.

1189 \newcommand*\caption@IncompatiblePackage[2]{%
1190   \PackageWarningNoLine{caption}{%
1191     Usage of the #1 package together with the\MessageBreak
1192     caption package is strongly *not* recommended.\MessageBreak
1193     Instead of loading the #1 package you should\MessageBreak
1194     use the caption package option\MessageBreak
1195     \space\space`tableposition=top'\MessageBreak
1196     #2}}

1197 \@onlypreamble\caption@IncompatiblePackage
```

### 2.4 Check document class

```
1198 \caption@ifbool{documentclass}{}{}%
1199   \PackageWarningNoLine{caption}{%
1200     Unsupported document class (or package) detected,\MessageBreak
```

```

1201     usage of the caption package is *not* recommended}%
1202   \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1203 }

```

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

```

1204 \@ifundefined{@captionheadfont}{}{}%
1205   \caption@Info{AMS or SMF document class}%
1206   \setlength\belowcaptionskip{0pt}% set to 3pc by AMS class
1207 }

```

## 2.6 Emulation of the KOMA-Script commands

```

1208 \@ifundefined{scr@caption}{}{}%
1209   \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 1210   \def\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1211   \def\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1212   \if@tablecaptionabove
1213     \caption@AtBeginDocument{%
1214       \def\caption@tempa#1{%
1215         \@ifpackageloaded{#1}{\caption@Error{%
1216           You can't use the global option\MessageBreak
1217           \space\space'tablecaptionabove'\MessageBreak
1218           and use the #1 package, too}}{}}}%
1219       \caption@tempa{ftcap}%
1220       \caption@tempa{nonfloat}%
1221       \caption@tempa{topcapt}}}%
1222   \@tablecaptionabovetrue
1223 \else
1224   \@tablecaptionabovefalse
1225 \fi

\onelinecaptionstrue
\onelinecaptionfalse 1226   \def\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1227   \def\onelinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1228   \ifonelinecaptions
1229     \onelinecaptionstrue
1230   \else
1231     \onelinecaptionfalse
1232   \fi

\@captionabovetrue
\@captionabovefalse 1233   \def\@captionabovetrue{\let\caption@position\@firstoftwo}
1234   \def\@captionabovefalse{\let\caption@position\@secondoftwo}

\setcapindent 1235   \let\caption@KOMA@setcapindent\@setcapindent
1236   \renewcommand*\@setcapindent[1]{%
1237     \caption@KOMA@setcapindent{#1}\caption@setcapindent}

```

```

1238 \let\caption@KOMA@@setcapindent\@@setcapindent
1239 \renewcommand*\@@setcapindent[1]{%
1240   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1241 \newcommand*\caption@setcapindent{%
1242   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1243 \@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.)
1244 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1245   \csname\string\setcapwidth\endcsname
1246 \@namedef{\string\setcapwidth}[1]#2{%
1247   \caption@KOMA@setcapwidth[#1]{#2}\caption@setcapwidth{#1}}
1248 \newcommand*\caption@setcapwidth[1]{%
1249   \ifx\#1\\\else
1250     \@ifundefined{cap@margin}{}{%
1251       \def\@tempa{captionbeside}%
1252       \ifx\@tempa\@currentenv\else\PackageWarning{caption}{%
1253         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1254       \fi}%
1255   \fi
1256   \captionsetup{width=\cap@width}}

1257 \def\caption@tempa{\hsize}%
1258 \ifx\caption@tempa\cap@width \else
1259   \caption@setcapwidth{?}
1260 \fi

\setcapmargin
1261 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1262   \csname\string\@setcapmargin\endcsname
1263 \@namedef{\string\@setcapmargin}[1]#2{%
1264   \caption@KOMA@setcapmargin[#1]{#2}\caption@setcapmargin}
1265 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1266   \csname\string\@@setcapmargin\endcsname
1267 \@namedef{\string\@@setcapmargin}[1]#2{%
1268   \caption@KOMA@@setcapmargin[#1]{#2}\caption@setcapmargin}
1269 \newcommand*\caption@setcapmargin{%
1270   \begingroup
1271     \let\onelinecaptionsfalse\relax
1272     \def\@twoside{0}%
1273     \def\if@twoside{\def\@twoside{1}\iffalse}%
1274     \cap@margin
1275     \def\@tempa{\endgroup}%
1276     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1277       \def\hspace##1##{\@firstofone}%
1278       \edef\@tempa{\endgroup
1279         \noexpand\captionsetup{%
1280           twoside=\@twoside,singlelinecheck=0,%
1281           margin={\cap@left,\cap@right}}}%
1282     \fi\fi
1283   \@tempa}

```

```

1284 \ifx\cap@margin\relax \else
1285 \caption@setcapmargin
1286 \fi
1287 }

```

## 2.7 Option for configuration files

```

1288 \DeclareCaptionOption*{config}[caption]{%
1289 \InputIfFileExists{#1.cfg}%
1290 {\typeout{*** Local configuration file #1.cfg used ***}}%
1291 {\PackageWarning{caption}{Configuration file #1.cfg not found}}}

```

## 2.8 Options for figure and table

```

1292 \DeclareCaptionOption{figureposition}{%
1293 \captionsetup*[figure]{position=#1}}
1294 \@onlypreamble@key{caption}{figureposition}

1295 \DeclareCaptionOption{tableposition}{%
1296 \caption@AtBeginDocument{%
1297 \def\caption@tempa##1{%
1298 \ifpackageloaded{##1}{\caption@Error{%
1299 You can't use the caption package option\MessageBreak
1300 \space\space`tableposition=#1'\MessageBreak
1301 and use the ##1 package, too}}{}}}%
1302 \caption@tempa{ftcap}%
1303 \caption@tempa{nonfloat}%
1304 \caption@tempa{topcapt}}}%
1305 \captionsetup*[table]{position=#1}}
1306 \@onlypreamble@key{caption}{tableposition}

1307 \DeclareCaptionOption{figurename}{\caption@SetFloatName{figure}{#1}}
1308 \@onlypreamble@key{caption}{figurename}
1309 \DeclareCaptionOption{tablename}{\caption@SetFloatName{table}{#1}}
1310 \@onlypreamble@key{caption}{tablename}

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

1311 \newcommand*\caption@SetFloatName[2]{%
1312 \begingroup
1313 \def\@tempa{\endgroup\@firstofone}%
1314 \ifundefined{language\@tempa}{\@tempa}%
1315 \ifundefined{captions\@tempa}{\@tempa}%
1316 \def\@tempa{\endgroup
1317 \expandafter\addto\csname captions\@tempa\endcsname}}}%
1318 \@tempa{\@namedef{#1name}{#2}}}%
1319 \@onlypreamble\caption@SetFloatName

```

## 2.9 caption v1.x compatibility options

```

1320 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1321 \@onlypreamble@key{caption}{compatibility}

1322 \DeclareCaptionOption*{normal}[]{%
1323 \caption@setformat{normal}%
1324 \caption@setjustification{justified}}
1325 \DeclareCaptionOption*{isu}[]{%

```

```

1326 \caption@setformat{hang}%
1327 \caption@setjustification{justified}}
1328 \DeclareCaptionOption*{hang}[]{%
1329 \caption@setformat{hang}%
1330 \caption@setjustification{justified}}
1331 \DeclareCaptionOption*{center}[]{%
1332 \caption@setformat{normal}%
1333 \caption@setjustification{centering}}
1334 \DeclareCaptionOption*{anne}[]{%
1335 \caption@setformat{normal}%
1336 \caption@setjustification{centerlast}}
1337 \DeclareCaptionOption*{centerlast}[]{%
1338 \caption@setformat{normal}%
1339 \caption@setjustification{centerlast}}

1340 \DeclareCaptionOption*{scriptsize}[]{\def\captionfont{\scriptsize}}
1341 \DeclareCaptionOption*{footnotesize}[]{\def\captionfont{\footnotesize}}
1342 \DeclareCaptionOption*{small}[]{\def\captionfont{\small}}
1343 \DeclareCaptionOption*{normalsize}[]{\def\captionfont{\normalsize}}
1344 \DeclareCaptionOption*{large}[]{\def\captionfont{\large}}
1345 \DeclareCaptionOption*{Large}[]{\def\captionfont{\Large}}

1346 \DeclareCaptionOption*{up}[]{\l@addto@macro\captionlabelfont\upshape}
1347 \DeclareCaptionOption*{it}[]{\l@addto@macro\captionlabelfont\itshape}
1348 \DeclareCaptionOption*{sl}[]{\l@addto@macro\captionlabelfont\slshape}
1349 \DeclareCaptionOption*{sc}[]{\l@addto@macro\captionlabelfont\scshape}
1350 \DeclareCaptionOption*{md}[]{\l@addto@macro\captionlabelfont\mdseries}
1351 \DeclareCaptionOption*{bf}[]{\l@addto@macro\captionlabelfont\bfseries}
1352 \DeclareCaptionOption*{rm}[]{\l@addto@macro\captionlabelfont\rmfamily}
1353 \DeclareCaptionOption*{sf}[]{\l@addto@macro\captionlabelfont\sffamily}
1354 \DeclareCaptionOption*{tt}[]{\l@addto@macro\captionlabelfont\ttfamily}

1355 \DeclareCaptionOption*{nooneline}[]{\caption@setbool{slc}{0}}

1356 \caption@setbool{ruled}{0}
1357 \DeclareCaptionOption*{ruled}[]{\caption@setbool{ruled}{1}}

```

## 2.10 caption2 v2.x compatibility options

```

1358 \DeclareCaptionOption*{flushleft}[]{%
1359 \caption@setformat{normal}%
1360 \caption@setjustification{raggedright}}
1361 \DeclareCaptionOption*{flushright}[]{%
1362 \caption@setformat{normal}%
1363 \caption@setjustification{raggedleft}}

1364 \DeclareCaptionOption*{oneline}[]{\caption@setbool{slc}{1}}

1365 \DeclareCaptionOption*{ignoreLTcapwidth}[]{%
1366 \PackageWarning{caption}{Obsolete option 'ignoreLTcapwidth' ignored}}

```

## 2.11 Obsolete caption v3.0 options

```

1367 \DeclareCaptionOption*{caption}{%
1368 \caption@setbool{temp}{#1}%
1369 \caption@ifbool{temp}{}{%
1370 \caption@Error{%
1371 The package option 'caption=#1' is obsolete.\MessageBreak
1372 Please pass this option to the subfig package instead\MessageBreak

```

```
1373         and do *not* load the caption package anymore}}}
```

## 2.12 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.

```
1374 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1375 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1376 \@onlypreamble@key{caption}{FPlist}
1377 \@onlypreamble@key{caption}{FPref}

1378 \newcommand*\caption@setFPoption[2]{%
1379   \edef\@tempa{\@car#2\@nil}%
1380   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1381 \@onlypreamble\caption@setFPoption

1382 \captionsetup{FPlist=caption,FPref=figure}
```

## 2.13 hyperref package support options

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

```
1383 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1384 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}
1385 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

## 2.14 Processing of options

```
1386 \caption@ProcessOptions*{caption}
```

## 2.15 Option ‘type’, `\captionof` and `\ContinuedFloat`

```
1387 \caption@AtBeginDocument{%
1388   \DeclareCaptionOption{type}{\caption@settype{#1}}
1389   \DeclareCaptionOption{type*}{\caption@settype*{#1}}}
```

*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@settypehook` (which will be used by our float package support).

The non-starred version sets a `hyperref` anchor additionally (if `hycap=true` and the `hycap` package is not loaded).

```
1390 \newcommand*\caption@settype{%
1391   \caption@@settype{}}

1392 \newcommand*\caption@@settype[1]{%
1393   \caption@teststar{\caption@@settype{#1}}{\@gobble\@firstofone}}
```

```

1394 \newcommand*\caption@@@settype[3]{%
1395 % #1 = "" or "sub"
1396 % #2 = \@gobble in star form, \@firstofone otherwise
1397 % #3 = <type>, e.g. "figure" or "table"
1398 \caption@Debug{type=#3}%
1399 \caption@checkgrouplevel{#1}{\captionsetup{#1type#2\@gobble*=#3}}%
1400 \@ifundefined{@#1capttype}{}{%
1401 \edef\@tempa{#3}%
1402 \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1403 \caption@iftypewarning{%
1404 \PackageWarning{caption}{%
1405 Caption type was already set to `#3'\MessageBreak}}{}%
1406 \fi}%
1407 \expandafter\edef\csname @#1capttype\endcsname{#3}%
1408 \let\caption@iftypewarning\caption@ifdebug
1409 \@nameuse{#1caption@settypehook}%
1410 \expandafter\caption@setoptions\csname @#1capttype\endcsname
1411 \let\caption@ifrefstepcounter\@secondoftwo
1412 #2{% \@gobble or \@firstofone
1413 \let\@currentlabel\caption@undefinedlabel
1414 \let\@currentHlabel\@undefined
1415 \caption@start}}

```

`\caption@settypehook` Hook, will be extended by our float package support.

```
1416 \newcommand*\caption@settypehook{}
```

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

```

1417 \newcommand*\caption@iftype{%
1418 \@ifundefined{@capttype}{\let\@capttype\@undefined\@secondoftwo}\@firstoftwo}

```

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

```

1419 \begingroup\expandafter\expandafter\expandafter\endgroup
1420 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1421 \caption@Debug{TeX engine: TeX}
1422 \let\caption@checkgrouplevel\@gobbletwo
1423 \else
1424 \caption@Debug{TeX engine: e-TeX}
1425 \newcommand*\caption@checkgrouplevel[2]{%
1426 \@ifundefined{#1caption@grouplevel}{%
1427 \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}}{}%
1428 \ifnum\currentgrouplevel>\caption@grouplevel\relax
1429 \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1430 \the\currentgrouplevel}%
1431 \else
1432 \caption@Error{\string#2 outside group or environment}%
1433 \fi
1434 }{}%
1435 \fi

```

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

*Note:* All commands used have to be robust here.

```
1436 \newcommand*\caption@undefinedlabel{%
1437   \emph{Please place \caption@cs{label} \textbf{after} \caption@cs{caption}}}%
1438 \DeclareRobustCommand\caption@cs[1]{\texttt{\char`\\#1}}
```

```
\captionof \captionof{<type>}[<lst.entry>]{<heading>}
\captionof* [<lst.entry>]{<heading>}
1439 \def\captionof{\caption@teststar\caption@of{\caption*}\caption}
1440 \newcommand*\caption@of[2]{%
1441   \let\caption@setfloatcapt\@firstofone
1442   \caption@settype*{#2}#1}
```

```
\captionlistentry \captionlistentry[<float type>]{<list entry>}
1443 \newcommand*\captionlistentry{%
1444   \@testopt\caption@listentry\@capttype}
1445 \def\caption@listentry[#1]#2{%
1446   \caption@refstepcounter{#1}%
1447   \caption@makecurrent{#1}{#2}%
1448   \caption@addcontentsline{#1}{#2}}
```

```
\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.
Note: The definition of \ContinuedFloat itself is compatible to the one inside the subfig pack-
age.
```

```
1449 \def\ContinuedFloat{%
1450   \caption@iftype
1451     {\addtocounter\@capttype\m@ne
1452     \caption@ContinuedFloat\@capttype}%
1453     {\caption@Error{\noexpand\ContinuedFloat outside float}}}%
1454 \def\caption@ContinuedFloat#1{%
1455   \edef\@tempa{#1}%
1456   \ifx\@tempa\caption@CFtype \else
1457     \caption@Error{Continued `#1' after '\caption@CFtype'}%
1458   \fi
1459   \stepcounter{ContinuedFloat}%
1460   \caption@@ContinuedFloat{#1}%
1461   \caption@setoptions{ContinuedFloat}%
1462   \caption@setoptions{continued#1}%
1463   \let\caption@resetContinuedFloat\@gobble}
1464 \newcounter{ContinuedFloat}
1465 \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.)
1466 \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`.)

```

1467 \newcommand*\caption@resetContinuedFloat[1]{%
1468   \@stpett{ContinuedFloat}\xdef\caption@CFtype{#1}}

```

## 2.16 Internal helpers

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

```

1469 \newcommand*\caption@refstepcounter[1]{%
1470   \caption@resetContinuedFloat{#1}%
1471   \caption@@refstepcounter{#1}%
1472   \let\caption@ifrefstepcounter\@firstoftwo}
1473 \newcommand*\caption@@refstepcounter{\refstepcounter}
1474 \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[]{}`).

```

1475 \newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1476 \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`)

```

1477 \newcommand*\caption@begin[1]{%
1478   \begingroup
1479   \caption@setfnum{#1}%
1480   \caption@fixposition
1481   \global\let\caption@fixedposition\caption@position}

```

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

5. Make an entry in the list-of-whatever.
6. Set `\caption@ifempty` according argument `<heading>`.

```

1482 \newcommand\caption@beginex[3]{%
1483   \caption@begin{#1}%
1484   \caption@addcontentsline{#1}{#2}%
1485   \caption@ifempty{#3}{}%

```

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

```

1486 \newcommand*\caption@end{%
1487   \endgroup
1488   \let\caption@position\caption@fixedposition}

```

```

\caption@setfnum \caption@setfnum{<type>}
redefines \fnum@<type> according the caption label format set with labelformat=.
But if labelformat=default is set, \fnum@<type> will not be overwritten by us.
1489 \newcommand*\caption@setfnum[1]{%
1490   \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1491   \@namedef{fnum@#1}{\caption@fnum{#1}}%
1492   \fi}

```

`\caption@addcontentsline` If `\sf@ifpositiontop` is defined, we extend `\caption@addcontentsline` so it handles sub-captions, too.

```

1493 \caption@AtBeginDocument*{%
1494   \@ifundefined{sf@ifpositiontop}{}{%
1495     \let\caption@kernel@addcontentsline\caption@addcontentsline
1496     \renewcommand*\caption@addcontentsline[2]{%
1497       \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
1498       \caption@kernel@addcontentsline{#1}{#2}%
1499       \sf@ifpositiontop{}{\@listsubcaptions{#1}}}%
1500   }}

```

`\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`.

```

1501 \newcommand*\caption@boxrestore{%
1502   \let\if@nobreak\iffalse
1503   \let\if@noskipsec\iffalse
1504   \let\par\@par
1505   \let\-\@dischyph
1506   \let'\@acci\let'\@accii\let\=\@acciii
1507   \parindent\z@ \parskip\z@skip
1508   \everypar{}%
1509   \linewidth\hsize
1510   \@totalleftmargin\z@
1511   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1512   \parfillskip\@flushglue \lineskip\normallineskip
1513   \baselineskip\normalbaselineskip
1514   \sloppy
1515   \let\\\@normalcr
1516 }

```

`\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).

```

1517 \newcommand*\caption@normalsize{%
1518   \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.

```

1519 \let\caption@setfloatcapt\@firstofone

```

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

```

\caption@makestart 1520 \newcommand*\caption@makecurrent[2]{}
\caption@anchor    1521 \let\caption@makestart\@gobble
\caption@start     1522 \let\caption@anchor\@firstofone
\caption@@start    1523 \let\caption@start\relax
\caption@freezeHref 1524 \let\caption@@start\relax
\caption@defrostHref 1525 \let\caption@freezeHref\relax
                  1526 \let\caption@defrostHref\relax
                  1527 \g@addto@macro\caption@prepareslc{\let\caption@anchor\@firstofone}

```

## 2.17 `\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.

```

1528 \caption@AtBeginDocument{%
1529   \caption@setbool{incompatible}{0}%
1530   \caption@CheckCommand\caption{%
1531     % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1532     \def\caption{%
1533       \ifx\@capttype\@undefined
1534         \@latex@error{\noexpand\caption outside float}\@ehd
1535         \expandafter\@gobble
1536       \else
1537         \refstepcounter\@capttype
1538         \expandafter\@firstofone
1539       \fi
1540       {\@dblarg{\@caption\@capttype}}%
1541     }%
1542   \caption@CheckCommand\caption{%
1543     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1544     \def\caption{

```

```

1545     \ifx\@capttype\@undefined
1546       \@latex@error{\noexpand\caption outside figure or table}\@ehd
1547       \expandafter\@gobble
1548     \else
1549       \refstepcounter\@capttype
1550       \expandafter\@firstofone
1551     \fi
1552     {\@dblarg{\@caption\@capttype}}%
1553   }}%

1554 \caption@CheckCommand\caption{%
1555   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1556   \renewcommand\caption{%
1557     \ifx\@capttype\@undefined
1558       \@latex@error{\noexpand\caption outside float}\@ehd
1559       \expandafter\@gobble
1560     \else
1561       \refstepcounter\@capttype
1562       \let\@tempf\@caption
1563       \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1564         \expandafter\expandafter\let
1565           \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1566       \fi
1567     \fi
1568     \@dblarg{\@tempf\@capttype}}}%

1569 \caption@CheckCommand\caption{%
1570   % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1571   % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1572   % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1573   \def\caption{%
1574     \ifx\@capttype\@undefined
1575       \@latex@error{\noexpand\caption outside float}\@ehd
1576       \expandafter\@gobble
1577     \else
1578       \H@refstepcounter\@capttype
1579       \@ifundefined{fst@\@capttype}{%
1580         \let\Hy@tempa\@caption
1581       }{%
1582         \let\Hy@tempa\Hy@float@caption
1583       }%
1584       \expandafter\@firstofone
1585     \fi
1586     {\@dblarg{\Hy@tempa\@capttype}}%
1587   }}%

1588 \caption@CheckCommand\caption{%
1589   % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1590   \def\caption{%
1591     \ifx\@capttype\@undefined
1592       \@latex@error{\noexpand\caption outside float}\@ehd
1593       \expandafter\@gobble
1594     \else
1595       \H@refstepcounter\@capttype
1596       \let\Hy@tempa\@caption
1597       \@ifundefined{float@caption}{%

```

```

1598         }{%
1599         \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
1600             \let\Hy@tempa\Hy@float@caption
1601         \fi
1602     }%
1603     \expandafter\@firstofone
1604     \fi
1605     {\@dblarg{\Hy@tempa\@capttype}}%
1606 }}%

1607 \caption@ifcheckcommand{}{%
1608     \caption@info{%
1609         Incompatible package detected (regarding \string\caption).\MessageBreak
1610         \string\caption\space=\space\meaning\caption}%
1611     \caption@setbool{incompatible}{1}}%

1612 \caption@checkcommand\@caption{%
1613     % ltfloating.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1614     \long\def\@caption#1[#2]#3{%
1615         \par
1616         \addcontentsline{\csname ext@#1\endcsname}{#1}%
1617             {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1618         \begingroup
1619             \@parboxrestore
1620             \if@minipage
1621                 \@setminipage
1622             \fi
1623             \normalsize
1624             \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1625         \endgroup}}%

1626 \caption@checkcommand\@caption{%
1627     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1628     \long\def\@caption#1[#2]#3{% second argument ignored
1629         \par\nobreak
1630         \begingroup
1631             \@parboxrestore
1632             \if@minipage
1633                 \@setminipage
1634             \fi
1635             \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
1636         \endgroup}}%

1637 % \caption@checkcommand\float@caption{%
1638 %     % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1639 %     \long\def\float@caption#1[#2]#3{%
1640 %         \addcontentsline{\@nameuse{ext@#1}}{#1}%
1641 %             {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
1642 %         \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
1643 %             \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
1644 %             \@ifnextchar[{\float@ccon}{\egroup}}%
1645 %     \long\def\float@ccon[#1]{#1\par\egroup}}%

1646 \caption@checkcommand\@caption{%
1647     % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1648     \long\def\@caption#1[#2]#3{%
1649         \hyper@makecurrent{\@capttype}%

```

```

1650 \def\@currentlabelname{#2}%
1651 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1652 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1653 }%
1654 \begingroup
1655 \let\parboxrestore
1656 \if@minipage
1657 \let\setminipage
1658 \fi
1659 \normalsize
1660 \@makecaption{\csname fnum@#1\endcsname}{%
1661 \ignorespaces
1662 \ifHy@nesting
1663 \hyper@@anchor{\@currentHref}{#3}%
1664 \else
1665 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1666 \fi
1667 }%
1668 \par
1669 \endgroup
1670 }%

1671 \caption@CheckCommand\@caption{%
1672 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1673 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1674 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1675 \long\def\@caption#1[#2]#3{%
1676 \expandafter\ifx\csname if@capstart\expandafter\endcsname
1677 \csname iftrue\endcsname
1678 \global\let\@currentHref\hc@currentHref
1679 \else
1680 \hyper@makecurrent{\@captype}%
1681 \fi
1682 \def\@currentlabelname{#2}%
1683 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1684 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1685 }%
1686 \begingroup
1687 \let\parboxrestore
1688 \if@minipage
1689 \let\setminipage
1690 \fi
1691 \normalsize
1692 \expandafter\ifx\csname if@capstart\expandafter\endcsname
1693 \csname iftrue\endcsname
1694 \global\@capstartfalse
1695 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1696 \else
1697 \@makecaption{\csname fnum@#1\endcsname}{%
1698 \ignorespaces
1699 \ifHy@nesting
1700 \hyper@@anchor{\@currentHref}{#3}%
1701 \else
1702 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1703 \fi

```

```

1704         }%
1705     \fi
1706     \par
1707 \endgroup
1708 } }%

1709 \caption@CheckCommand\@caption{%
1710 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1711 \long\def\@caption#1[#2]{%
1712     \def\@currentlabelname{#2}%
1713     \NR@@caption{#1}[{#2}]%
1714 } }%

1715 \caption@CheckCommand\@caption{%
1716 % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1717 \long\def\@caption#1[#2]#3{%
1718     \ifundefined{if#1topcap}%
1719     {\subfig@oldcaption{#1}[{#2}]{#3}}%
1720     {\@nameuse{if#1topcap}%
1721     \@listsubcaptions{#1}%
1722     \subfig@oldcaption{#1}[{#2}]{#3}%
1723     \else
1724     \subfig@oldcaption{#1}[{#2}]{#3}%
1725     \@listsubcaptions{#1}%
1726     \fi}}}%

1727 \caption@CheckCommand\@caption{%
1728 % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1729 \def\@caption{\caption@}%
1730 \long\def\caption@#1[#2]#3{%
1731 % \ifundefined{caption@setfloattype}%
1732 % \caption@settype
1733 % \caption@setfloattype
1734 % \@captype
1735 % \sf@ifpositiontop{%
1736 % \@listsubcaptions{#1}%
1737 % \sf@old@caption{#1}[{#2}]{#3}%
1738 % }%
1739 % \sf@old@caption{#1}[{#2}]{#3}%
1740 % \@listsubcaptions{#1}%
1741 % } }%
1742 }%

1743 \caption@IfCheckCommand{}{%
1744     \caption@Info{%
1745         Incompatible package detected (regarding \string\@caption).\MessageBreak
1746         \string\@caption\space=\space\meaning\@caption}%
1747     \caption@setbool{incompatible}{1}}%

```

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

```

1748 \ifundefined{caption@ifcompatibility}%
1749 {\let\caption@ifcompatibility\caption@ifincompatible
1750 \let\@tempa\@firstofone}%
1751 {\let\@tempa\@gobble}% suppress warning
1752 \caption@ifcompatibility{%
1753     \@tempa{%

```

```

1754     \PackageWarning{caption}{%
1755         *****\MessageBreak
1756         \noexpand\caption will not be redefined since it's already\MessageBreak
1757         redefined by a document class or package which is\MessageBreak
1758         unknown to the caption package.\MessageBreak
1759         As a result some new features of the caption\MessageBreak
1760         package will not work as expected.\MessageBreak
1761         *****\@gobbletwo}%
1762     }%

\ContinuedFloat is not supported in compatibility mode.

1763     \renewcommand*\caption@ContinuedFloat[1]{%
1764         \caption@Error{Not available in compatibility mode}}%

\caption@start is not supported in compatibility mode.

1765     \caption@AtBeginDocument*{%
1766         \let\caption@start\relax
1767         \@ifundefined{caption@ORI@capstart}{}{%
1768             \caption@Debug{%
1769                 Undo redefinition of \string\capstart\@gobble}%
1770             \let\capstart\caption@ORI@capstart
1771             \let\caption@ORI@capstart\@undefined}%
1772         \@ifundefined{Hy@float@makebox}{}{%
1773             \caption@Debug{%
1774                 Restore hyperref redefinition of \string\float@makebox\@gobble}%
1775             \let\float@makebox\Hy@float@makebox
1776             \let\Hy@float@makebox\@undefined}%
1777         }%

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

1778     \newcommand*\caption@star[2]{\#1\#2}%

1779 }{%

1780     \caption@ifincompatible{%
1781         \PackageWarning{caption}{%
1782             *****\MessageBreak
1783             Forced redefinition of \noexpand\caption since the\MessageBreak
1784             unsupported(!) package option 'compatibility=false'\MessageBreak
1785             was given. Please fasten your seat belts, as we may\MessageBreak
1786             experience turbulence!\MessageBreak
1787             *****\@gobbletwo}%
1788         }{}%

\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.

1789     \def\caption{%
1790         \caption@iftype
1791         {\caption@checkgrouplevel\@empty\caption
1792         \caption@star
1793         {\caption@refstepcounter\@capttype}%
1794         {\caption@dblarg{\@caption\@capttype}}}%
1795         {\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.

```

1796 \newcommand*\caption@star[2]{%
1797 \ifstar{\global\caption@startrue#2[]}{#1#2}}%

```

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

```

1798 \long\def\@caption#1[#2]#3{%
1799 \ifcaption@star \else
1800 \caption@makecurrent{#1}{#2}%
1801 \caption@ifhypcap{\caption@@start}{}%
1802 \fi
1803 \par
1804 \caption@beginex{#1}{#2}{#3}%
1805 \caption@setfloatcapt{%
1806 \caption@boxrestore
1807 \caption@normalsize
1808 \ifcaption@star
1809 \let\caption@anchor\@firstofone
1810 \fi
1811 \@makecaption{\csname fnum@#1\endcsname}%
1812 {\ignorespaces\caption@anchor{#3}}\par}%
1813 \caption@end}%

```

`\@xfloat` We redefine `\@xfloat` so inside floating environments our type-specific options will be used, a `hyperref` anchor will be set etc.

```

1814 \let\caption@ORI@xfloat\@xfloat
1815 \def\@xfloat#1[#2]{%
1816 \caption@ORI@xfloat{#1}[#2]%
1817 \caption@settype{#1}}%
1818 }%
1819 }

```

`\@makecaption` `\@makecaption{<label>}{<text>}`  
We do basically the same as the original code (from the standard L<sup>A</sup>T<sub>E</sub>X document classes), but take care of the `position=` setting and use `\caption@@make` from the `caption` kernel to finally typeset the caption.

```

1820 \long\def\@makecaption#1#2{%
1821 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
1822 \caption@@make{#1}{#2}%
1823 \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}}

```

## 2.18 Document class & Babel package support

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

```
1824 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}
```

### 2.18.2 The beamer class

```
1825 \@ifclassloaded{beamer}{%
1826   \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.

```
1827   \captionsetup{listof=false}
1828   \undefine@key{caption}{listof}
1829   \undefine@key{caption}{listofformat}
1830 }{ }
```

### 2.18.3 The KOMA-Script classes

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

```
1831 \@ifundefined{scr@caption}{}{%
1832   \caption@AtBeginDocument{\let\scr@caption\caption}}
```

### 2.18.4 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)

```
1833 \@ifundefined{caption@frenchb}{}{\caption@frenchb}
```

### 2.18.5 The frenchle/pro package

```
1834 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
1835   \caption@Info{frenchle/pro package is loaded}%

1836   \let\captionfont@ORI\captionfont
1837   \let\captionlabelfont@ORI\captionlabelfont
1838   \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`.

```
1839   \@ifundefined{GOfrench}%
1840     {\let\@tempa\@firstofone}%
1841     {\def\@tempa{\g@addto@macro\GOfrench}}}%
1842   \@tempa{%

1843     \let\captionfont\captionfont@ORI
1844     \let\captionfont@ORI\undefined
1845     \let\captionlabelfont\captionlabelfont@ORI
1846     \let\captionlabelfont@ORI\undefined
1847     \let\@makecaption\@makecaption@ORI
1848     \let\@makecaption@ORI\undefined
```

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

```
1849     \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.
1850 \let\caption@tcORI\@tablescaption
1851 \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.
1852 \let\@eatDP\@undefined
1853 \let\@tempa\@empty
1854 \ifx\ffrench\figure
1855 \l@addto@macro\@tempa{\let\fnum@figure\ffrench}%
1856 \fi
1857 \ifx\tfrench\table
1858 \l@addto@macro\@tempa{\let\fnum@table\tfrench}%
1859 \fi
1860 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
1861 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
1862 \@tempa

1863 }%
1864 }}

```

## 2.19 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.
1865 \newcommand\caption@ifpackageloaded[1]{%
1866 \testopt{\caption@ifpackageloaded{#1}}{}%
1867 \onlypreamble\caption@ifpackageloaded
1868 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
1869 \ifpackageloaded{#1}\@firstofone{%
1870 \caption@Debug{#1 package is not loaded (yet)\@gobble}%
1871 \caption@AtBeginDocument}{%
1872 \caption@ifpackageloaded{#1}[#2]{#3}{#4}}%
1873 \onlypreamble\caption@ifpackageloaded
1874 \newcommand\caption@ifpackageloaded[1]{%
1875 \testopt{\caption@ifpackageloaded{#1}}{}%
1876 \onlypreamble\caption@ifpackageloaded
1877 \long\def\caption@ifpackageloaded#1[#2]{%
1878 \ifpackageloaded{#1}{%
1879 \caption@Info{#1 package is loaded}%
1880 \ifpackagelater{#1}[#2]\@firstoftwo{%
1881 \caption@Error{%
1882 For a successful cooperation we need at least version\MessageBreak
1883 `#2' of package #1,\MessageBreak
1884 but only version\MessageBreak

```

```

1885         \csname ver@#1.\@pkgextension\endcsname'\MessageBreak
1886         is available}%
1887         \@secondoftwo}%
1888     }\@secondoftwo}}
1889 \@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.

```

1890 \newcommand*\caption@clearmargin{%
1891     \setcaptionmargin\z@
1892     \let\caption@minmargin\@undefined}

1893 \caption@setbool{needfreeze}{0}
1894 \caption@AtBeginDocument*{%
1895     \caption@ifneedfreeze{%

```

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

```

1896 \newcommand*\caption@freeze{%
1897     \caption@teststar\caption@@freeze\@gobble\@firstofone}%

1898 \newcommand*\caption@@freeze[1]{%
1899     \global\let\caption@SCcontinued\relax
1900     \global\let\caption@SCsetup\@undefined
1901     \global\let\caption@SClentry\@undefined
1902     \global\let\caption@SCtext\@undefined
1903     \global\let\caption@SClabel\@undefined

1904     \let\caption@ORI@ContinuedFloat\ContinuedFloat
1905     \def\ContinuedFloat{%
1906         \let\caption@ORI@setcounter\setcounter
1907         \let\caption@ORI@addtocounter\addtocounter
1908         \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
1909         \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
1910         \caption@ORI@ContinuedFloat
1911         \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
1912         \let\setcounter\caption@ORI@setcounter
1913         \let\addtocounter\caption@ORI@addtocounter}%
1914     \def\captionsetup##1{%
1915         \caption@g@addto@list\caption@SCsetup{##1}%
1916         \caption@setup{##1}}%
1917     \def\caption{%
1918         \let\captionsetup\@gobble
1919         \caption@star
1920         {\let\caption@ORI@refstepcounter\caption@@refstepcounter
1921         \def\caption@@refstepcounter#####1{%
1922             \advance\csname c@#####1\endcsname\@ne\relax}%
1923         \caption@refstepcounter\@captype
1924         \let\caption@@refstepcounter\caption@ORI@refstepcounter}%
1925         {\caption@dblarg\caption@SC}}%
1926     \def\@caption@SC[##1]##2{%
1927         \gdef\caption@SClentry{##1}\gdef\caption@SCtext{##2}%
1928         \ignorespaces}%
1929     #1{% is \@gobble in star form, and \@firstofone otherwise
1930         \def\label##1{\gdef\caption@SClabel{##1}\ignorespaces}}%

```

```

1931 }%

\caption@defrost \caption@defrost
1932 \newcommand*\caption@defrost{%
1933   \ifx\caption@ORI@ContinuedFloat\@undefined
1934     \caption@SCcontinued
1935     \ifx\caption@SCsetup\@undefined \else
1936       \expandafter\captionsetup\expandafter{\caption@SCsetup}%
1937     \fi
1938     \ifx\caption@SCtext\@undefined \else
1939       \expandafter\expandafter\expandafter\caption
1940       \expandafter\expandafter\expandafter[%
1941       \expandafter\expandafter\expandafter[%
1942       \expandafter\caption@SClentry\expandafter]\expandafter]%
1943       \expandafter{\caption@SCtext}%
1944     \fi
1945     \ifx\caption@SClabel\@undefined \else
1946       \expandafter\label\expandafter{\caption@SClabel}%
1947     \fi
1948   \else
1949     \caption@Error{Internal error:\MessageBreak
1950       \noexpand\caption@defrost in same group as \string\caption@freeze}%
1951   \fi}%

1952 }{}%
1953 \caption@undefbool{needfreeze}}

```

### 2.19.1 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.)

```

1954 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
1955   \@ifpackageloaded{floatrow}{%
1956     \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
1957   }{}%

```

`\@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>]`.

```

1958 \let\caption@ORI@float@setevery\@float@setevery
1959 \def\@float@setevery#1{%
1960   \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.

```
1961 \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.

```
1962 \renewcommand\caption@setfloatcapt{%
1963 \let\@makecaption\caption@@make
1964 \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.

```
1965 \float@getstyle\float@style{#1}%
1966 \caption@setstyle*\float@style
1967 \caption@setoptions\float@style
1968 }{}%
1969 \caption@freezeHref % will be defrosted in \float@makebox
1970 \caption@ORI@float@setevery{#1}}%
```

`\caption@setfloatcapt` Here we do a consistency check: If the caption to handle is a one from the float package (`\newfloat` or `\restylefloat`), `\setfloatcapt` should be redefined, so actually this error should never occur.

```
1971 \renewcommand*\caption@setfloatcapt{%
1972 \float@ifcaption\@capttype{%
1973 \caption@Error{Something is going wrong here...\MessageBreak
1974 For a removal of this problem please consult the\MessageBreak
1975 caption package documentation, especially the sections\MessageBreak
1976 about the 'float' and the 'wrapfig' package}%
1977 \@gobble
1978 }{%
1979 \@firstofone
1980 }}%
```

`\caption@settypehook` 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.

```
1981 \g@addto@macro\caption@settypehook{%
```

```

1982     \expandafter\ifx\csname #1name\endcsname\relax
1983     \expandafter\let\csname #1name\expandafter\endcsname
1984         \csname fname@#1\endcsname
1985     \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.

1986 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
1987 \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).

1988 \providecommand*\float@ifstyle[1]{%
1989     \expandafter\ifx\csname fst@#1\endcsname\relax
1990     \expandafter\@secondoftwo
1991     \else
1992     \expandafter\@firstoftwo
1993     \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.

1994 \providecommand*\float@getstyle[2]{%
1995     \edef#1{%
1996         \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
1997         \expandafter\expandafter\expandafter\noexpand
1998         \csname fst@#2\endcsname}%
1999     \edef#1{#1}%
2000     \caption@Debug{floatstyle{#2} = '#1'}}%

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

2001 \providecommand*\float@setstyle[2]{%
2002     \expandafter\edef\csname fst@#1\endcsname{%
2003         \expandafter\noexpand\csname fs@#2\endcsname}}%

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

2004 \providecommand*\float@dostyle[1]{%
2005     \@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.)

2006 \providecommand*\float@ifcaption[1]{%
2007   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2008   \expandafter\@firstoftwo
2009   \else
2010   \expandafter\@secondoftwo
2011   \fi}%

2012 }{%
2013 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2014 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2015 % \clearcaptionsetup{boxed}% used by the floatrow package?
2016 }

```

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

```
2017 \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’.

```

2018 \caption@ifbool{ruled}{%
2019   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
2020 }{%
2021   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2022 }
2023 \caption@undefbool{ruled}

```

## 2.19.2 The floatflt package

```
2024 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

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

```

2025 \let\caption@ORI@floatingfigure\floatingfigure
2026 \def\floatingfigure{%
2027   \caption@floatflt{figure}%
2028   \caption@ORI@floatingfigure}%

```

\floatingtable Same with \floatingtable...

```

2029 \let\caption@ORI@floatingtable\floatingtable
2030 \def\floatingtable{%
2031   \caption@floatflt{table}%
2032 % \caption@setautoposition b%
2033   \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.

```

2034 \newcommand*\caption@floatflt[1]{%
2035   \caption@settype{#1}%
2036   \caption@setoptions{floating#1}%
2037   \let\caption@boxrestore\@parboxrestore}%

2038 }{}

```

### 2.19.3 The fltpage package

```

2039 \caption@ifpackageloaded{fltpage}[1998/10/29 v.0.3]{%
2040   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

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

```

```

2041 \renewcommand\FP@helpNote[2]{%
2042   \begingroup % save \caption@thepage
2043     \caption@pageref{#2}%
2044     \typeout{FP#1 is inserted on page \caption@thepage!}%
2045   \endgroup}%

```

`\FP@floatBegin` Original code:

```

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

```

```

2046 \renewcommand*\FP@floatBegin[1]{%
2047   \def\@captive{#1}%
2048   \let\FP@LabelText\@empty
2049   \begin{lrbox}{\FP@floatCorpusBOX}%
2050     \caption@ifFPrefcap\caption@freeze{%
2051       \def\label##1{\gdef\FP@LabelText{##1}\ignorespaces}%
2052       \caption@freeze*}%

```

`\FP@floatEnd` Original code:

```

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

2053 \renewcommand*\FP@floatEnd{%
2054   \end{lrbox}%
2055   \stepcounter{FP@{\@cptype} C}%
2056   \caption@label\FP@positionLabel
2057   \FP@helpNote{\@cptype}\FP@positionLabel
2058   \let\FP@ContinuedFloat\relax
2059   \FP@float
2060     {\FP@positionLabel}% location label test
2061     {\begin{\@cptype}[p!]}%
2062       \FP@ContinuedFloat
2063       \usebox\FP@floatCorpusBOX
2064       \caption@ifFlistcap
2065         {\caption@refstepcounter{\@cptype}
2066           \expandafter\caption@makecurrent\expandafter{\@cptype}
2067             \expandafter{\caption@SClentry}}}%
2068         {\expandafter\caption@listentry\expandafter{\caption@SClentry}}}%
2069       \caption@anchor\relax
2070       \ifx\FP@LabelText\@empty \else
2071         \expandafter\label\expandafter{\FP@LabelText}%
2072       \fi
2073     \end{\@cptype}%
2074   {\let\FP@ContinuedFloat\ContinuedFloat
2075     \caption@ifcompatibility

```

```

2076         {\let\caption@ContinuedFloat\@gobble}%
2077         {\caption@clearsetup{ContinuedFloat}}%
2078         \caption@clearsetup{continued\@capttype}}}%
2079     {\begin\@capttype[b!]}%
2080         \FP@ContinuedFloat
2081         \let\FP@savedSetfnumCommand\caption@setfnum
2082         \def\caption@setfnum##1{%
2083             \FP@savedSetfnumCommand{##1}%
2084             \ifx\FP@guide\@empty \else
2085                 \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}%
2086             \fi}%
2087         \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2088         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2089         \caption@setoptions{FP\@capttype}%
2090         \FP@separatorCaption
2091         \caption@ifFPlistcap{}\{\caption@set@bool\caption@iflof0}%
2092         \caption@defrost
2093     \end\@capttype}%
2094 }%

2095 \ForFloatingEnvironments{%
2096     \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2097 }{%
2098     \let\caption@ifFPlistcap\@undefined
2099     \let\caption@ifFPrefcap\@undefined
2100 }

```

#### 2.19.4 The hyperref package

```

2101 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2102     \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2103         \PackageWarningNoLine{caption}{%
2104             Hyperref support is turned off\MessageBreak
2105             because hyperref has stopped early}%
2106     }{%

```

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

```

2107     \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

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

```

2108     \renewcommand*\caption@makecurrent[2]{%
2109         \caption@makecurrentHref{#1}%
2110         \caption@Debug{hyper current=\@currentHref}%
2111         \def\@currentlabelname{#2}}%
2112     \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

`\caption@makestart` We redefine `\caption@makestart` so a `hyperref` label will be defined inside `\caption@start`. Since we offer `\ContinuedFloat` the float counter can be 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.xxx.<counter>` (instead of `#1.<counter>`) as naming scheme for `\@currentHref` to avoid conflicts with other hyper links which are generated with `hypertextnames=true`.

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

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

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

```
2113 \renewcommand*\caption@makestart[1]{%
2114 \begingroup
2115 \Hy@hypertextnamesfalse
2116 \gdef\@currentHlabel{}%
2117 \hyper@makecurrent{#1.xxx}%
2118 \endgroup
2119 \caption@Debug{hyper start=\@currentHref}}%
```

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

```
2120 \renewcommand*\caption@anchor[1]{%
2121 \caption@Debug{hyper anchor \@currentHref}%
2122 % If we cannot have nesting, the anchor is empty.
2123 \ifHy@nesting
2124 \hyper@@anchor{\@currentHref}{#1}%
2125 \else
2126 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2127 \fi}%

```

`\caption@@anchor` We define `\caption@@anchor` so a `hyperref` anchor will be set inside `\caption@start`.

```
2128 \newcommand*\caption@@anchor[1]{%
2129 \caption@Debug{hyper anchor #1}%
2130 \begingroup
2131 \@tempdima\prevdepth
2132 \nointerlineskip
2133 \vspace*{-\caption@hypcapspace}%
2134 \let\leavevmode\relax
2135 \hyper@@anchor{#1}\relax
2136 \vspace*{\caption@hypcapspace}%
2137 \prevdepth\@tempdima
2138 \endgroup}%

```

`\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 \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.

```
2139 \renewcommand*\caption@@ContinuedFloat[1]{%
2140 \ifundefined{theH#1}}{}%
2141 \expandafter\l@addto@macro\csname theH#1\endcsname{%
2142 \@alph\c@ContinuedFloat}}}%
2143 \newif\if@capstart
```

`\caption@start` This one emulates the `hypcap` package (if `hypcap=true`).

```
2144 \def\caption@start{%
2145 \caption@ifhypcap{%
2146 \ifvmode
```

Generate the `hyperref` label and set the `hyperref` anchor, usually both is done inside `\@caption`.

```
2147 \caption@makestart\@captype
2148 \caption@@anchor\@currentHref
```

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

```
2149 \global\@capstarttrue
2150 \let\hc@currentHref\@currentHref
2151 \def\caption@makecurrentHref##1{%
2152 \global\@capstartfalse
2153 \global\let\@currentHref\hc@currentHref}%
```

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

```
2154 \let\caption@anchor\@firstofone
2155 \let\caption@@anchor\@gobble
2156 \else
2157 \PackageWarning{caption}{Wrong TeX mode, no hyperlink anchor was set}%
2158 \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.

```
2159 \def\caption@@start{%
2160 \ifundefined{hc@currentHref}}{%
2161 \PackageWarning{caption}{%
2162 The option 'hypcap=true' will be ignored for this\MessageBreak
2163 particular \string\caption}}}%
```

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

```
2164 \def\caption@freezeHref{%
2165 \let\caption@ORI@start\caption@start
2166 \def\caption@start{\let\caption@start\caption@ORI@start}%
2167 \let\caption@ORI@@start\caption@@start
2168 % \l@addto@macro\subcaption@settypehook{%
2169 % \let\caption@@start\caption@ORI@@start}%
2170 \global\let\caption@currentHref\@undefined
2171 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
```

```

2172      \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2173      \renewcommand*\caption@setfloatcapt{%
2174        \ifx\caption@currentHref\@undefined \else
2175          \let\caption@anchor\@firstofone
2176        \fi
2177        \caption@ORI@setfloatcapt}}%

\caption@defrostHref If there is a freezed \@currentHref, we set the hyperref anchor here.
2178      \def\caption@defrostHref{%
2179        \ifx\caption@currentHref\@undefined \else
2180          \caption@@anchor\caption@currentHref
2181          \global\let\caption@currentHref\@undefined
2182        \fi}%

\float@makebox Do our own redefinition of \float@makebox, if it was redefined by the hyperref package.
2183      \@ifundefined{HyOrg@float@makebox}{}{%
2184        \caption@Debug{%
2185          Redefining \noexpand\float@makebox (again)\@gobble}%
2186        \let\Hy@float@makebox\float@makebox
2187        \renewcommand\float@makebox[1]{%
2188          \HyOrg@float@makebox{#1\relax \caption@defrostHref}}%
2189      }%

2190 }{}

2.19.5 The hypcap package

2191 \caption@IfPackageLoaded{hypcap}{% v1.0
2192   \ifx\caption@start\relax \else % hyperref hasn't stopped early

If the hypcap package was loaded, we give up our own hypcap emulation and give the control to the hypcap package instead.

\capstart We do this simply by mapping \capstart to \caption@start, although our code does not behave exactly like the original one: The original \capstart has an effect on the next \caption only but our version affects all \captions in the same environment, at least unless a new \capstart will be placed.
2193   \let\caption@ORI@capstart\capstart
2194   \let\capstart\caption@start
2195   \let\caption@start\relax
2196   \let\caption@@start\relax

\caption@hypcapspace Furthermore we map our \caption@hypcapspace to \hypcapspace offered by the hypcap package.
2197   \renewcommand*\caption@hypcapspace{\hypcapspace}%

\float@makebox Redo the redefinition of \float@makebox, if necessary.
2198   \@ifundefined{Hy@float@makebox}{}{%
2199     \caption@Debug{%
2200       Restore hyperref redefinition of \string\float@makebox\@gobble}%
2201     \let\float@makebox\Hy@float@makebox
2202     \let\Hy@float@makebox\@undefined}%
2203   \fi}{}

```

## 2.19.6 The listings package

```

2204 \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.
2205 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2206 \def\lst@MakeCaption#1{%
2207 \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.
2208 \caption@setautoposition{#1}%
2209 \caption@begin{lstlisting}%
2210 \caption@ORI@lst@MakeCaption{#1}%
2211 \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.
2212 \providecommand*\ext@lstlisting{lol}%
2213 {}

```

## 2.19.7 The longtable package

```

\LTcapttype \LTcapttype is preset to table.
2214 \providecommand*\LTcapttype{table}
2215 \caption@ifpackageloaded{longtable}[1995/05/24 v3.14]{%
2216 \RequirePackage{ltcaption}[2007/09/01]%
2217 \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.
2218 \caption@AtBeginDocument{%
2219 \let\caption@ORI@LT@array\LT@array
2220 \renewcommand*\LT@array{%
\captionsetup for longtable:
2221 \global\let\caption@opt@@longtable\@undefined
2222 \def\captionsetup##1{\noalign{\LT@captionsetup{##1}}}%
2223 \def\LT@captionsetup##1{%
2224 \captionsetup\@gobble[@longtable]{##1}%
2225 \global\let\caption@opt@@longtable\caption@opt@@longtable}%
\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2226 \def\captionabovetrue{\LT@captionsetup{position=t}}%
2227 \def\captionabovefalse{\LT@captionsetup{position=b}}%

```

```

\captionlistentry for longtable:
2228      \def\captionlistentry##1{\noalign{%
2229      \caption@listentry[\LTcapttype]{##1}}}%

\ContinuedFloat for longtable:
(Commented out, since it's not deeply tested and quite useless anyway)
Note: hyperref versions < v6.76j uses 2x \hyper@makecurrent
2230 %      \caption@ifhyhcap{%
2231 %      \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2232 %      \def\hyper@makecurrent##1{%
2233 %      \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2234 %      \caption@makestart{##1}%
2235 %      \let\Hy@LT@currentHlabel\@currentHlabel
2236 %      \let\Hy@LT@currentHref\@currentHref
2237 %      \def\hyper@makecurrent###1{%
2238 %      \let\@currentHlabel\Hy@LT@currentHlabel
2239 %      \let\@currentHref\Hy@LT@currentHref}}%
2240 %      \let\caption@ORI@ContinuedFloat\ContinuedFloat
2241 %      \def\ContinuedFloat{\noalign{%
2242 %      \gdef\caption@setContinuedFloat{%
2243 %      \let\caption@resetContinuedFloat\@gobble}%
2244 %      \def\caption@setoptions####1{%
2245 %      \g@addto@macro\caption@setContinuedFloat{%
2246 %      \caption@setoptions{####1}}}%
2247 %      \let\@capttype\LTcapttype
2248 %      \caption@ORI@ContinuedFloat}}%
2249 %      }{%
2250 %      \def\ContinuedFloat{\noalign{%
2251 %      \caption@Error{%
2252 %      \noexpand\ContinuedFloat inside longtables\MessageBreak
2253 %      is only available with 'hyhcap=true'}}}%
2254 %      }%
2255 %      \global\let\caption@setContinuedFloat\@empty
2256 %      \def\ContinuedFloat{\noalign{%
2257 %      \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2258 %      \caption@ORI@LT@array}}%

```

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

```

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

```

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

```

2259 \def\LT@c@ption#1[#2]#3{%
2260   \LT@makecaption#1{\csname fnun@\LTcapttype\endcsname}{#3}%
2261   \def\@currentlabelname{#2}%
2262   \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:

```
2263 \renewcommand\LT@makecaption[3]{%
2264   \caption@LT@make{%
```

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

```
2265   \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`.)

```
2266   \caption@settype*\LTcapytype
2267   \ifdim\LTcapwidth=4in \else
2268     \setcaptionwidth\LTcapwidth
2269   \fi
2270   \caption@setoptions{longtable}%
2271 %   \caption@setContinuedFloat
2272   \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!

```
2273   \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
2274   \caption@resetContinuedFloat\LTcapytype
2275   \caption@begin\LTcapytype
2276   \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...)

```
2277   \vskip-\ht\strutbox
```

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

```

2278      \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2279      \caption@@make{#2}{#3}\endgraf
2280      \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2281      \caption@end}}%

2282 }{}

```

## 2.19.8 The picinpar package

```

2283 \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.

```

```

2284 \long\def\figwindow[#1,#2,#3,#4] {%
2285   \caption@window{figure}%
2286   \caption@setoptions{figwindow}%
2287   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%

2288 \long\def\tabwindow[#1,#2,#3,#4] {%
2289   \caption@window{table}%
2290   \caption@setoptions{tabwindow}%
2291   \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.

```

```

2292 \newcommand*\caption@window{%
2293   \let\caption@boxrestore\@parboxrestore
2294   \let\@makecaption\caption@@make
2295   \caption@setautoposition b%
2296   \caption@settype}%

```

```

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

```

```

2297 \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.

```

```

2298   \ifdim\picwd=\z@
2299     \let\caption@makecurrent\@gobbletwo
2300     \let\caption@@start\relax
2301     \caption@prepareslc
2302   \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.

```

2303   \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2304   \if\@tempa*
2305     \let\@tempa\@firstofone
2306   \else\if\@tempa[%

```

```

2307     \let\@tempa\@firstofone
2308     \else
2309         \let\@tempa\@empty
2310     \fi\fi
2311     \expandafter\caption\@tempa{#1}}%
2312 {}{}

```

## 2.19.9 The picins package

`\piccaptiontype` `\piccaptiontype{<type>}`

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

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

```
2313 \newcommand*\piccaptiontype[1]{\def\@piccaptiontype{#1}}
```

```
2314 \caption@ifPackageLoaded{picins}{%
```

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

```

2315     \ifundefined{\@piccaptiontype}{%
2316         \caption@iftype{%
2317             \let\@piccaptiontype\@capttype
2318         }{%
2319             \def\@piccaptiontype{figure}%
2320         }%
2321     }{}%
2322     \let\@capttype\undefined

```

`\piccaption` The original code:

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

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

```
2323 \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.

```

2324 \let\caption@ORI@make@piccaption\make@piccaption
2325 \def\make@piccaption{%
2326   \let\caption@ORI\caption
2327   \long\def\caption[##1]##2{%
2328     \caption@freezeHref % will be defrosted in \ivparpic
2329     \caption@settype\@piccaptiontype
2330     \ifnum\c@piccaptionpos>2\relax
2331       \caption@clearmargin
2332     \else
2333       \captionwidth\z@ % do not use "width=" setting
2334     \fi
2335     \caption@setoptions{parpic}%
2336     \let\caption@boxrestore\@parboxrestore
2337     \caption@setautoposition b%
2338     \expandafter\expandafter\expandafter\caption@ORI
2339       \expandafter\expandafter\expandafter[%
2340       \expandafter\expandafter\expandafter{%
2341       \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
2342 \caption@ORI@make@piccaption
2343 \let\caption\caption@ORI}%

```

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

```

2344 \let\caption@ORI@ivparpic\ivparpic
2345 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
2346   \let\caption@ORI@noindent\noindent
2347   \def\noindent{%
2348     \caption@defrostHref
2349     \let\noindent\caption@ORI@noindent
2350     \noindent}%
2351   \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2352   \let\noindent\caption@ORI@noindent}%
2353 }{%
2354 \let\piccaptiontype\@undefined
2355 }

```

## 2.19.10 The rotating package

```

2356 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%

```

```

\rotcaption Make \rotcaption* work.
2357 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
2358 % \let\@rotcaption\@undefined

\rotcaptionof Make \rotcaptionof(*) work.
2359 \def\rotcaptionof{%
2360 \caption@teststar\caption@of{\rotcaption*}\rotcaption}%

\@makerotcaption Original (bugfixed) code:

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

```

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

```

2361 \long\def\@makerotcaption#1#2{%
2362 \ifdim\captionwidth=\z@
2363 \setcaptionwidth{.8\textheight}%
2364 \caption@slc{#1}{#2}{.8\vsiz}{%
2365 \let\caption@makerot\caption@@make
2366 \caption@clearmargin
2367 % \long\def\captionbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2368 % (not needed because \rotatebox uses an \hbox anyway)
2369 \let\captionbox\@secondoftwo}%
2370 \caption@set@bool\caption@ifslc0% been there, done that
2371 \fi
2372 \rotatebox{90}{\caption@makerot{#1}{#2}}%
2373 \nobreak\hspace{12pt}}%
2374 \newcommand\caption@makerot[2]{%
2375 \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
2376 \ForFloatingEnvironments{%
2377 \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
2378 \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
2379 {}

```

### 2.19.11 The sidecap package

```

2380 \caption@ifPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
2381 \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.)
2382 \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`.

```
2383 \let\caption@ORI@SC@zfloat\SC@zfloat
2384 \def\SC@zfloat#1#2#3[#4]{%
2385   \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
```

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.

```
2386   \caption@settype*{#2}%
2387   \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.

```
2388 \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
2389 \def\endSC@FLOAT{%
```

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

```
2390   \let\caption@ORI@settype\caption@settype
2391   \def\caption@settype##1{% will be done in \@xfloat
2392     \caption@ORI@settype*{##1}% do not change \@currentlabel
2393     \caption@setSC@justify
2394     \caption@setoptions{SCfloat}%
2395     \caption@setoptions{SC\@captype}%
2396     \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

```
2397   \def\caption@setSC@justify{%
2398     \caption@clearmargin
2399     \@ifundefined{SC@justify}{}{%
2400       \ifx\SC@justify\@empty \else
2401         \let\caption@hj\SC@justify
2402         \let\SC@justify\@empty
2403       \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!

```
2404   \let\caption\SC@orig@caption
2405   \def\SC@orig@caption[##1]##2{\caption@defrost}%
```

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

```
2406   \caption@setSC@justify % for compatibility mode
2407   \caption@ORI@endSC@FLOAT}%
```

```

2408 \ForFloatingEnvironments{%
2409   \newenvironment{SC#1}%
2410     {\SC@float[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@float}%
2411   \newenvironment{SC#1*}%
2412     {\SC@dblfloat[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@dblfloat}}%
2413 {}

```

### 2.19.12 The subfigure package

```

2414 \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.

```

2415 \def\sf@ifpositiontop{%
2416   \ifx\@capttype\undefined
2417     \expandafter\@gobbletwo
2418   \else\ifx\@capttype\relax
2419     \expandafter\expandafter\expandafter\@gobbletwo
2420   \else
2421     \expandafter\expandafter\expandafter\sf@if@position@top
2422   \fi\fi}
2423 \def\sf@if@position@top{%
2424   \ifundefined{if\@capttype topcap}%
2425     {\@gobbletwo}%
2426     {\@nameuse{if\@capttype topcap}%
2427       \expandafter\@firstoftwo
2428     \else
2429       \expandafter\@secondoftwo
2430     \fi}}
2431 {}

```

### 2.19.13 The supertabular package

```

2432 \caption@ifPackageLoaded{supertabular}[2002/07/19 v4.1e]{%

```

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

```

2433 \let\caption@ORI@tablecaption\tablecaption
2434 \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}

```

```

2435 \long\def\ST@caption#1[#2]#3{\par%
2436   \caption@settype*{#1}%
2437   \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.

```

2438   \def\caption@fixposition{%
2439     \caption@setposition{\if@topcaption t\else b\fi}}%
2440   \caption@beginex{#1}{#2}{#3}%
2441   \caption@boxrestore
2442   \caption@normalsize
2443   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2444   \caption@end}%
2445 }{%
2446   \let\caption@setSTposition\@undefined
2447 }

```

## 2.19.14 The threeparttable package

```

2448 \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.

```

2449   \let\caption@ORI@threeparttable\threeparttable
2450   \renewcommand*\threeparttable{%
2451     \caption@settype{table}%
2452     \caption@setposition a% ?
2453     \caption@clearmargin
2454     \caption@setoptions{threeparttable}%
2455     \caption@ORI@threeparttable}%

```

\measuredfigure Same here...

```

2456   \let\caption@ORI@measuredfigure\measuredfigure
2457   \renewcommand*\measuredfigure{%
2458     \caption@settype{figure}%
2459     \caption@setposition a% ?
2460     \caption@clearmargin
2461     \caption@setoptions{measuredfigure}%
2462     \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}

```

```

2463 \def\TPT@caption#1[#2]#3{%
2464   \gdef\TPT@docapt{%
2465     \global\let\TPT@docapt\@undefined
2466     \caption@setautoposition\caption@TPT@position
2467     \TPT@LA@caption{#1}[{#2}]{#3}}%

```

```

2468 \ifx\TPT@hsize\@empty
2469 \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
2470 \gdef\caption@TPT@position{t}%
2471 \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2472 \else
2473 \def\caption@TPT@position{b}%
2474 \TPT@docapt
2475 \fi
2476 \ignorespaces}%

2477 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2478 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%

2479 {}{}

```

## 2.19.15 The wrapfig package

```

2480 \caption@ifPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)

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

2481 \providecommand*\float@ifstyle[1]{%
2482 \expandafter\ifx\csname fst@#1\endcsname\relax
2483 \expandafter\@secondoftwo
2484 \else
2485 \expandafter\@firstoftwo
2486 \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.

```

2487 \newcommand*\caption@restylewrapfloat[1]{%
2488 \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2489 \csname wrap#1\endcsname
2490 \caption@Error{%
2491 For a successful cooperation of the 'wrapfig' package\MessageBreak
2492 with the 'float' package you should load the 'wrapfig'\MessageBreak
2493 package *after* the 'float' package}%
2494 \else
2495 \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2496 \csname wrap#1\endcsname
2497 \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
2498 \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2499 \csname wrap#1\endcsname
2500 \fi}%

```

```

\caption@wrapfloat

2501 \newcommand*\caption@wrapfloat[1]{%
2502 \caption@settype*{#1}%
2503 \float@ifstyle{#1}{%
2504 \ifx\WF@floatstyhook\@undefined

```

```

2505         \caption@Error{%
2506             For a successful cooperation of the 'wrapfig' package\MessageBreak
2507             with the 'float' package you should use at least\MessageBreak
2508             'wrapfig' version 3.6}%
2509     \else
2510         \float@dostyle{#1}%
2511     \fi}}}%
2512     \caption@clearmargin
2513     \caption@setoptions{wrapfloat}%
2514     \caption@setoptions{wrap#1}%
2515     \@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`.)

```

2516     \caption@restylewrapfloat{figure}%
2517     \caption@restylewrapfloat{table}%
2518     \ForFloatingEnvironments{%
2519         \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
2520         \caption@restylewrapfloat{#1}}%
2521     \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.

```

2522     \ifundefined{float@restyle}{}{%
2523         \toks@=\expandafter{\float@restyle{#1}% (env may or may not be defined)
2524         \caption@restylewrapfloat{#1}}%
2525         \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2526         \@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`.

```

2527     \let\caption@ORI@wrapfloat\wrapfloat
2528     \def\wrapfloat#1{%
2529         \float@ifstyle{#1}{%
2530             \caption@Error{%
2531                 For a successful cooperation of the 'wrapfig' package\MessageBreak
2532                 with the 'float' package you should load the 'wrapfig'\MessageBreak
2533                 package *right after* the 'float' package}}}%
2534         \caption@ORI@wrapfloat{#1}}%
2535     \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\bgrouper \setlength\hsize{#2}%

```

```

\ifdim\hsize>\z@ \@parboxrestore \else
\setbox\z@\hbox\bgroup \let\wf@@caption\caption \let\caption\wf@caption
\ignorespaces \fi}

```

**Our code:**

```

2536 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2537 \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2538 \global\setbox\WF@box\top\bgroup \setlength\hsize{#2}%
2539 \caption@start
2540 \ifdim\hsize>\z@ \@parboxrestore \else
2541 \setbox\z@\hbox\bgroup \let\wf@@caption\caption \let\caption\wf@caption
2542 \ignorespaces \fi}%

2543 }{}

```

## 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*,  
2006/07/24
- [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*,  
June 14, 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