

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

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

2007/10/25

## 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 v3.1d, last revised 2007/10/25.

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

# Contents

<b>1</b>	<b>Kernel</b>	<b>4</b>
1.1	Identification . . . . .	4
1.2	Generic helpers . . . . .	4
1.3	Errors . . . . .	7
1.4	Using the keyval package . . . . .	8
1.5	Margin resp. width . . . . .	12
1.6	Indentions . . . . .	14
1.7	Styles . . . . .	14
1.8	Formats . . . . .	16
1.9	Label formats . . . . .	16
1.10	Label separators . . . . .	17
1.11	Text formats . . . . .	18
1.12	Fonts . . . . .	18
1.13	Justifications . . . . .	20
1.13.1	The ragged2e package . . . . .	21
1.14	Vertical spaces before and after captions . . . . .	22
1.15	Positioning . . . . .	22
1.16	Hooks . . . . .	23
1.17	Lists . . . . .	24
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 . . . . .	25
1.19.3	The beamer class . . . . .	26
1.19.4	The KOMA-Script classes . . . . .	27
1.19.5	The NTG Dutch classes . . . . .	28
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 . . . . .	35
<b>2</b>	<b>Main package</b>	<b>37</b>
2.1	Identification . . . . .	37
2.2	Loading the kernel . . . . .	37
2.3	Check against incompatible packages . . . . .	37
2.4	Check document class . . . . .	37
2.5	Adaption to the $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF document classes . . . . .	38

2.6	Emulation of the KOMA-Script commands	38
2.7	Option for configuration files	40
2.8	Options for figure and table	40
2.9	caption v1.x compatibility options	40
2.10	caption2 v2.x compatibility options	41
2.11	Obsolete caption v3.0 options	41
2.12	ftpage package support options	42
2.13	hyperref package support options	42
2.14	Processing of options	42
2.15	Option ‘type’, \captionof and \ContinuedFloat	42
2.16	Internal helpers	45
2.17	\caption, \@caption, and \@makecaption	47
2.18	Document class & Babel package support	54
2.18.1	The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes	54
2.18.2	The beamer class	54
2.18.3	The KOMA-Script classes	54
2.18.4	The frenchb Babel option	54
2.18.5	The frenchle/pro package	54
2.19	Package support	55
2.19.1	The float package	57
2.19.2	The floatflt package	60
2.19.3	The ftpage package	61
2.19.4	The hyperref package	63
2.19.5	The hypcap package	67
2.19.6	The listings package	67
2.19.7	The longtable package	68
2.19.8	The picinpar package	70
2.19.9	The picins package	71
2.19.10	The rotating package	73
2.19.11	The sidecap package	74
2.19.12	The subfigure package	75
2.19.13	The supertabular package	76
2.19.14	The threeparttable package	77
2.19.15	The wrapfig package	77

# 1 Kernel

## 1.1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2007/10/25 v3.1d 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 \@removeelement is using \edef
to build the new list!
43 \newcommand*\caption@l@removefrom@list[2]{%
44 \caption@ifempty@list#1{\@expandtwoargs\@removeelement{#2}#1#1}}

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

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

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

```

```

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

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

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

\caption@teststar \caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
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@Warning`  
`\caption@WarningNoLine`  
`\caption@Error`  
`\caption@KV@err`

```

\caption@Error{<message>}
127 \newcommand*\caption@Warning[1]{%
128   \caption@WarningNoLine{#1\on@line}}
129 \newcommand*\caption@WarningNoLine[1]{%
130   \PackageWarning{caption}{#1.^J\caption@wh\@gobbletwo}}
131 \newcommand*\caption@Error[1]{%
132   \PackageError{caption}{#1}\caption@eh}
133 \let\caption@KV@err\caption@Error

```

`\caption@wh`  
`\caption@eh`

At the moment we only offer these two simple warning resp. error helpers.

```

134 \newcommand*\caption@wh{%
135   See the caption package documentation for explanation.}
136 \newcommand*\caption@eh{%
137   If you do not understand this error, please take a closer look\MessageBreak
138   at the documentation of the 'caption' package, especially the\MessageBreak
139   section about errors.\MessageBreak\@ehc}

```

## 1.4 Using the keyval package

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

```
140 \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.

```
141 \providecommand*\undefine@key[2]{%
142   \@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>.

```
143 \providecommand*\@onlypreamble@key[2]{%
144   \expandafter\@onlypreamble\csname KV@#1@#2\endcsname}
145 \@onlypreamble\@onlypreamble@key
```

```
\DeclareCaptionOption \DeclareCaptionOption{<option>}[<default value>]{<code>}
\DeclareCaptionOption*{<option>}[<default value>]{<code>}
```

We declare our options using these commands (instead of using `\DeclareOption` offered by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>), so the keyval package is used. The starred form makes the option available during the lifetime of the current package only, so they can be used with `\usepackage`, but *not* with `\captionsetup` later on.

```
146 \newcommand*\DeclareCaptionOption{%
147   \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
148 \@onlypreamble\DeclareCaptionOption
149 \newcommand*\caption@declareoption[2]{%
150   #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
151 \@onlypreamble\caption@declareoption
```

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

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

```
152 \newcommand*\DeclareCaptionOptionNoValue{%
153   \caption@teststar\caption@declareoption@novalue\AtEndOfPackage\@gobble}
154 \@onlypreamble\DeclareCaptionOptionNoValue
155 \newcommand\caption@declareoption@novalue[3]{%
156   \caption@declareoption{#1}{#2}[KV@err]{%
157     \caption@option@novalue{#2}{##1}{#3}}}
158 \@onlypreamble\caption@declareoption@novalue
159 \newcommand*\caption@option@novalue[2]{%
160   \ifx\KV@err#2%
161     \expandafter\@firstofone
162   \else
163     \KV@err{No value allowed for #1}%
164     \expandafter\@gobble
165   \fi}
```

```
\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`.

```
166 \newcommand*\captionsetup{%
167   \caption@teststar\captionsetup\@gobble\@firstofone}
```



```

168 \newcommand*\captionsetup[1]{%
169   \@ifnextchar[{\caption@setup@options#1}\caption@setup}
170 \newcommand*\caption@setup{\caption@setkeys{caption}}
171 \def\caption@setup@options#1[#2]{%
172   % #1 is \gobble in star form, and \@firstofone otherwise
173   #1{\caption@addtooptlist{#2}}%
174   \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.

```

175 \newcommand*\clearcaptionsetup{%
176   \caption@teststar\@clearcaptionsetup\gobble\@firstofone}
177 \newcommand*\@clearcaptionsetup[1]{%
178   \let\caption@tempa#1%
179   \@testopt\@clearcaptionsetup{}}
180 \def\@clearcaptionsetup[#1]#2{%
181   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
182   {\caption@tempa{\caption@Warning{Option list `#2' undefined}}}%
183   {\ifx,#1,%
184     \caption@clearsetup{#2}%
185     \else
186     \caption@@removefromsetup{#1}{#2}%
187     \fi}}
188 \newcommand*\caption@clearsetup[1]{%
189   \caption@removefromoptlist{#1}%
190   \@nameundef{caption@opt@#1}}
191 \newcommand*\caption@removefromsetup{%
192   \let\caption@tempa\gobble
193   \caption@@removefromsetup}
194 \newcommand*\caption@@removefromsetup[2]{%
195   \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
196   \expandafter\let\csname caption@opt@#2\endcsname\@undefined
197   \def\@tempb##1=##2\@nil{##1}%
198   \edef\@tempc{#1}%
199   \@for\@tempa:=\@tempa\do{%
200     \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
201     \ifx\@tempd\@tempc
202       \let\caption@tempa\gobble
203     \else
204       \expandafter\expandafter\expandafter\caption@l@addto@list
205       \expandafter\csname caption@opt@#2\expandafter\endcsname
206       \expandafter{\@tempa}%
207     \fi}%
208   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
209   {\caption@removefromoptlist{#2}}}%
210   \caption@tempa{\caption@Warning{%
211     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>*.

```

212 \newcommand*\showcaptionsetup[2][\@firstofone]{%
213   \GenericWarning{ }{%
214     #1 Caption Info: KV list on `#2'\MessageBreak
215     #1 Caption Data: \@ifundefined{caption@opt@#2}{%
216       -none-%
217     }{%
218       {\expandafter\expandafter\expandafter\strip@prefix
219         \expandafter\meaning\csname caption@opt@#2\endcsname}%
220     }}
221 \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.

```

222 \newcommand*\caption@setoptions[1]{%
223   \caption@Debug{options=#1}%
224   \expandafter\let\expandafter\@tempa\csname caption@opt@#1\endcsname
225   \ifx\@tempa\relax \else
226     \let\caption@iftypewarning\@firstoftwo % = true
227     \expandafter\caption@setup\expandafter{\@tempa}%
228     \caption@clearsetup{#1}%
229   \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”).

```

230 \newcommand*\caption@addtooptlist[1]{%
231   \@ifundefined{caption@opt@#1@lineno}{%
232     \caption@dooptlist\caption@g@addto@list{#1}%
233     \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
234   }{}}
235 \newcommand*\caption@removefromoptlist[1]{%
236   \caption@dooptlist\caption@g@removefrom@list{#1}%
237   \global\expandafter\let\csname caption@opt@#1@lineno\endcsname\@undefined}
238 \newcommand*\caption@dooptlist[2]{%
239   \begingroup
240     \edef\@tempa{#2}\@onelevel@sanitize\@tempa
241     \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
242   \endgroup}
243 \AtEndDocument{%
244   \caption@for@list\caption@optlist{%
245     \PackageWarningNoLine{caption}{%
246       Unused \string\captionsetup{#1}
247       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.

```

248 \newcommand*\caption@setkeys{\@dblarg\caption@@setkeys}
249 \long\def\caption@@setkeys[#1]#2#3{%
250   \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
251   \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
252   \ifundefined{#1@KV@err}%
253     {\let\KV@err\ORI@KV@err\let\KV@errx\ORI@KV@errx}%
254     {\expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
255      \let\KV@errx\KV@err}%
256   \l@addto@macro\caption@keydepth i%
257   \caption@Debug{\protect\setkeys{#2}{#3}}%
258   \setkeys{#2}{#3}%
259   \edef\caption@keydepth{\expandafter\@cdr\caption@keydepth\@nil}%
260   \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
261   \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname
262 \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`.

```

263 \newcommand*\caption@ExecuteOptions[2]{%
264   \@expandtwoargs\caption@setkeys{#1}{#2}}%
265 \@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.)

```

266 \newcommand*\caption@ProcessOptions{%
267   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
268 \@onlypreamble\caption@ProcessOptions
269 \newcommand*\caption@@ProcessOptions[2]{%
270   \let\@tempc\relax
271   \let\caption@tempa\@empty
272   #1{% \@firstofone -or- \@gobble
273     \@for\CurrentOption:=\@classoptionslist\do{%
274       \ifundefined{KV@#2@CurrentOption}{}{%
275         \ifundefined{KV@#2@CurrentOption @default}{%
276           \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
277         }{%
278           \PackageInfo{#2}{Global option '\CurrentOption' processed}%
279           \edef\caption@tempa{\caption@tempa,\CurrentOption,%
280             \@expandtwoargs\removeelement\CurrentOption
281             \@unusedoptionlist\@unusedoptionlist
282           }%
283         }%
284       }%
285       \let\CurrentOption\@empty
286     }%

```

```

287 \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
288 \AtEndOfPackage{\let\@unprocessedoptions\relax}
289 \@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@`.

```

290 \newdimen\captionmargin
291 \newdimen\captionmargin@
292 \newdimen\captionwidth

293 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
294 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
295 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
296 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}

297 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
298 \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.

```

299 \newcommand*\setcaptionmargin{%
300   \caption@teststar\caption@setmargin\@gobble\@firstofone}

301 \newcommand*\caption@setmargin[2]{%
302   #1{\captionwidth\z@}%
303   \caption@@setmargin#2,#2,\@nil}

304 \def\caption@@setmargin#1,#2,#3\@nil{%
305   \setlength\captionmargin@{#2}%
306   \setlength\captionmargin{#1}%
307   \addtolength\captionmargin@{-\captionmargin}}

```

`\setcaptionwidth` `\setcaptionwidth{<amount>}`

Please never use this in user documents, it’s just there to provide compatibility to the `caption2` package.

```

308 \newcommand*\setcaptionwidth{%
309   \captionmargin\z@
310   \captionmargin@\z@
311   \setlength\captionwidth}

```

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

```

312 \newcommand*\caption@count{0}

```

`\caption@newlabel` This command is a modified version of `\newlabel` from L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>. 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`.)

```

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

```

`\caption@thepage` This command is a modified version of `\thepage` from L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>. It will be used inside `\caption@ifoddpage` only.

```

314 \newcommand*\caption@thepage{\the\c@page}

```

`\caption@label` This command is a modified version of `\label` from L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

315 \newcommand*\caption@label[1]{%
316   \protected@write\@auxout{\let\caption@thepage\relax}%
317     {\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>X2<sub>ε</sub>. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

318 \newcommand*\caption@pageref[1]{%
319   \expandafter\ifx\csname caption@r@#1\endcsname\relax
320     \G@refundefinedtrue % => 'There are undefined references.'
321     \caption@Warning{Reference on page \thepage \space undefined}%
322   \else
323     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
324   \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.

```

325 \newcommand*\caption@ifoddpage{%
326   \caption@iftwoside{%
327     \begingroup
328       \@tempcnta\caption@count
329       \advance\@tempcnta\@ne
330       \xdef\caption@count{\the\@tempcnta}%
331     \endgroup
332     \caption@label\caption@count
333     \caption@pageref\caption@count
334     \ifodd\caption@thepage
335       \let\caption@ifoddpage\@firstoftwo
336     \else
337       \let\caption@ifoddpage\@secondoftwo
338     \fi
339   }{\let\caption@ifoddpage\@firstoftwo}%
340   \caption@ifoddpage}

```

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

```

341 \newcommand*\caption@setoptcmd[2]{%
342   \caption@ifinlist{#2}{0,false,no,off}{\let#1\undefined}{\def#1{#2}}}

```

## 1.6 Indentions

```

\caption@indent
\caption@parindent
\caption@hangindent

These are the indentions we support.

343 \newdimen\caption@indent
344 \newdimen\caption@parindent
345 \newdimen\caption@hangindent

346 \DeclareCaptionOption{indent}[\leftmargini]{%
347     \setlength\caption@indent{#1}}% obsolete!
348 \DeclareCaptionOption{indentation}[\leftmargini]{%
349     \setlength\caption@indent{#1}}
350 \DeclareCaptionOption{parindent}{%
351     \setlength\caption@parindent{#1}}
352 \DeclareCaptionOption{hangindent}{%
353     \setlength\caption@hangindent{#1}}
354 \DeclareCaptionOption{parskip}{%
355     \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>

```

356 \@ifundefined{scr@caption}{}{%
357     \let\caption@KV@parindent\KV@caption@parindent
358     \DeclareCaptionOption{parindent}[]{%
359         \ifx,#1,%
360             \caption@Debug{Option 'parindent' ignored}%
361         \else
362             \caption@KV@parindent{#1}%
363         \fi}%
364     \let\caption@KV@parskip\KV@caption@parskip
365     \DeclareCaptionOption{parskip}[]{%
366         \ifx,#1,%
367             \caption@Debug{Option 'parskip' ignored}%
368         \else
369             \caption@KV@parskip{#1}%
370         \fi}%
371 }

```

## 1.7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
372 \newcommand*\DeclareCaptionStyle[1]{%
373     \@testopt{\caption@declarestyle{#1}}{}%
374 \@onlypreamble\DeclareCaptionStyle

```

<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!)

```

375 \def\caption@declarestyle#1[#2]#3{%
376   \global\@namedef{caption@sls@#1}{#2}%
377   \global\@namedef{caption@sty@#1}{#3}}
378 \@onlypreamble\caption@declarestyle

379 \DeclareCaptionOption{style}{\caption@setstyle{#1}}
380 \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.

```

381 \newcommand*\caption@setstyle{%
382   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

383 \newcommand*\caption@@setstyle[2]{%
384   \@ifundefined{caption@sty@#2}{%
385     {#1{\caption@Error{Undefined style `#2'}}}}%
386     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
387       \ifx\caption@setstyle@flag\@undefined
388         \let\caption@setstyle@flag\caption@setstyle
389         \let\caption@iftypewarning\@firstoftwo % = true
390         \caption@resetstyle
391         \expandafter\caption@setup\expandafter{\caption@sty}%
392         \let\caption@setstyle@flag\@undefined
393       }else
394         \expandafter\caption@setup\expandafter{\caption@sty}%
395       }fi
396     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
397     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
398     \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!*

```

399 \newcommand*\caption@resetstyle{%
400   \caption@setup{%
401     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
402     justification=justified,font=,labelfont=,textfont=,%
403     margin=0pt,minmargin=0,maxmargin=0,%
404     indent=0pt,parindent=0pt,hangindent=0pt,%
405     singlelinecheck=1,strut=1}%
406   \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.

```

407 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
408 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
409   format=default,labelsep=default,textformat=default,%
410   justification=default,font=default,labelfont=default,textfont=default}

```

## 1.8 Formats

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

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

```
411 \newcommand*\DeclareCaptionFormat{%
412   \caption@teststar\caption@declareformat\@gobble\@firstofone}
413 \@onlypreamble\DeclareCaptionFormat

414 \newcommand*\caption@declareformat[2]{%
415   \@dblarg{\caption@@declareformat#1{#2}}}%
416 \@onlypreamble\caption@declareformat

417 \long\def\caption@@declareformat#1#2[#3]#4{%
418   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
419   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
420   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
421 \@onlypreamble\caption@@declareformat

422 \DeclareCaptionOption{format}{\caption@setformat{#1}}
```

```
\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`).

```
423 \newcommand*\caption@setformat[1]{%
424   \ifundefined{caption@fmt@#1}%
425     {\caption@Error{Undefined format `#1'}}%
426     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
427      \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
428      \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}
```

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

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

430 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
431   \caption@ifin@list\caption@lsepclist\caption@lsepname
432   {\caption@Error{%
433     The option ‘labelsep=\caption@lsepname’ does not work\MessageBreak
434     with ‘format=hang’}}%
435   {\@hangfrom{#1#2}%
436     \advance\caption@parindent\hangindent
437     \advance\caption@hangindent\hangindent
438     \caption@@par#3\par}}
```

‘default’ usually maps to ‘plain’.

```
439 \def\caption@fmt@default{\caption@fmt@plain}
440 \def\caption@ifh@default{\caption@ifh@plain}
```

## 1.9 Label formats

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

441 \newcommand*\DeclareCaptionLabelFormat[2]{%
442   \global\@namedef{caption@lfmt@#1}##1##2##3{#2}}
443 \@onlypreamble\DeclareCaptionLabelFormat
```



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

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

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

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

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

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

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

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

```
449 \DeclareCaptionLabelFormat{empty}{}%
```

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

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

‘default’ usually maps to ‘simple’.

```
452 \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`.

```
453 \newcommand\DeclareCaptionLabelSeparator{%
```

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

```
455 \@onlypreamble\DeclareCaptionLabelSeparator
```

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

```
457 \global\expandafter\let\csname caption@iflf@#2\endcsname#1%
```

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

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

```
460 \@onlypreamble\caption@declarelabelseparator
```

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

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

```
463 \caption@g@addto@list\caption@lsep@crlist{#1}%
```

```
464 \fi}
```

```
465 \@onlypreamble\caption@@declarelabelseparator
```

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

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

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

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

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

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

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

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

```
472 \expandafter\let\expandafter\caption@iflf\csname caption@iflf@#1\endcsname
```

```
473 \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’.

```
474 \DeclareCaptionLabelSeparator{none}{}%
```

```
475 \DeclareCaptionLabelSeparator{colon}{: }
```

```

476 \DeclareCaptionLabelSeparator{period}{. }
477 \DeclareCaptionLabelSeparator{space}{ }
478 \DeclareCaptionLabelSeparator*{quad}{\quad}
479 \DeclareCaptionLabelSeparator*{newline}{\\}
480 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}
‘default’ usually maps to ‘colon’.
481 \def\caption@lsep@default{\caption@lsep@colon}
482 \def\caption@iflf@default{\caption@iflf@colon}

```

## 1.11 Text formats

```

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

486 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
487 \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`).

```

488 \newcommand*\caption@settextformat[1]{%
489   \ifundefined{caption@tfmt@#1}%
490     {\caption@Error{Undefined text format `#1'}}%
491     {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

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

```

492 \DeclareCaptionTextFormat{simple}{#1}
493 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

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

## 1.12 Fonts

```

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

DeclareDefaultCaptionFont \DeclareDefaultCaptionFont{<name>}{<code>}
498 \newcommand*\DeclareDefaultCaptionFont[2]{%
499   \global\@namedef{caption#1@default}{#2}}
500 \@onlypreamble\DeclareDefaultCaptionFont

501 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
502 \DeclareDefaultCaptionFont{font}{}

503 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
504 \DeclareDefaultCaptionFont{labelfont}{}

505 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
506 \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.

```
507 \newcommand*\caption@setfont[2]{%
508   \begingroup
509     \let\caption@fnt\@empty
510     \define@key{caption@fnt}{default}[]{%
511       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
512     \caption@setkeys[caption]{caption@fnt}{#2}%
513     \global\let\caption@tempa\caption@fnt
514   \endgroup
515   \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`.

```
516 \newcommand*\caption@font{%
517   \caption@teststar\caption@@font\@firstofone
518     {\caption@setkeys[caption]{caption@fnt}}%
519 \newcommand*\caption@@font[2]{%
520   \begingroup
521   \def\caption@fnt{\endgroup}%
522   #1{#2}%
523   \caption@fnt}
```

These are the pre-defined font code snippets.

```
524 \DeclareCaptionFont{normalcolor}{\normalcolor}
525 \DeclareCaptionFont{color}{\color{#1}}

526 \DeclareCaptionFont{normalfont}{\normalfont}
527 \DeclareCaptionFont{up}{\upshape}
528 \DeclareCaptionFont{it}{\itshape}
529 \DeclareCaptionFont{sl}{\slshape}
530 \DeclareCaptionFont{sc}{\scshape}
531 \DeclareCaptionFont{md}{\mdseries}
532 \DeclareCaptionFont{bf}{\bfseries}
533 \DeclareCaptionFont{rm}{\rmfamily}
534 \DeclareCaptionFont{sf}{\sffamily}
535 \DeclareCaptionFont{tt}{\ttfamily}

536 \DeclareCaptionFont{scriptsize}{\scriptsize}
537 \DeclareCaptionFont{footnotesize}{\footnotesize}
538 \DeclareCaptionFont{small}{\small}
539 \DeclareCaptionFont{normalsize}{\normalsize}
540 \DeclareCaptionFont{large}{\large}
541 \DeclareCaptionFont{Large}{\Large}

542 \DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespace}% normally 1
543 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
544 \DeclareCaptionFont{doublespacing}{\doublespacing}
545 \DeclareCaptionFont{stretch}{\setstretch{#1}}
546 \caption@AtBeginDocument{\providecommand*\setstretch[1]{} }
547 %\DeclareCaptionFont{normal}{%
```

```

548 % \caption@font{normalcolor,normalfont,normalsize,singlespacing}
549 \DeclareCaptionFont{normal}{%
550 \caption@font*{%
551 \KV@caption@fnt@normalcolor\@unused
552 \KV@caption@fnt@normalfont\@unused
553 \KV@caption@fnt@normalsize\@unused
554 \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.

```

555 \providecommand\captionsize{}

556 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
557 \DeclareDefaultCaptionFont{size}{}

```

### 1.13 Justifications

```

\DeclareCaptionJustification \DeclareCaptionJustification{<name>}{<code>}
558 \newcommand*\DeclareCaptionJustification[2]{%
559 \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
560 \DeclareCaptionFont{#1}{#2}}
561 \@onlypreamble\DeclareCaptionJustification

```

```

\DeclareDefaultCaptionJustification{<code>}
562 \newcommand*\DeclareDefaultCaptionJustification[1]{%
563 \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
564 \DeclareDefaultCaptionFont{@hj}{#1}}
565 \@onlypreamble\DeclareDefaultCaptionJustification

566 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
567 \DeclareDefaultCaptionJustification{}

```

```

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

```

These are the pre-defined justification code snippets.

```

569 \DeclareCaptionJustification{justified}{}
570 \DeclareCaptionJustification{centering}{\centering}
571 \DeclareCaptionJustification{centerfirst}{\centerfirst}
572 \DeclareCaptionJustification{centerlast}{\centerlast}
573 \DeclareCaptionJustification{raggedleft}{\raggedleft}
574 \DeclareCaptionJustification{raggedright}{\raggedright}

```

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

```

575 \providecommand\centerfirst{%
576 \let\\\@centercr
577 \edef\caption@normaladjust{%
578 \leftskip\the\leftskip
579 \rightskip\the\rightskip
580 \parfillskip\the\parfillskip\relax}%
581 \leftskip\z@\@plus -1fil%
582 \rightskip\z@\@plus 1fil%

```

```

583 \parfillskip\z@skip
584 \noindent\hskip\z@\@plus 2fil%
585 \@setpar{\@@par\@restorepar\caption@normaladjust}}

```

\centerlast This is based on code from Anne Brüggemann-Klein[22]

```

586 \providecommand\centerlast{%
587 \let\\\@centercr
588 \leftskip\z@\@plus 1fil%
589 \rightskip\z@\@plus -1fil%
590 \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.

```

591 \DeclareCaptionJustification{Centering}{%
592 \caption@ragged\Centering\centering}
593 \DeclareCaptionJustification{RaggedLeft}{%
594 \caption@ragged\RaggedLeft\raggedleft}
595 \DeclareCaptionJustification{RaggedRight}{%
596 \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.

```

597 \newcommand*\caption@ragged{%
598 \caption@Debug{We need ragged2e}%
599 \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
600 \global\let\caption@ragged\caption@@ragged
601 \caption@ragged}

602 \caption@AtBeginDocument{%
603 \@ifundefined{caption@r@ragged2e}{%
604 \newcommand*\caption@@ragged{%
605 \caption@Warning{%
606 'ragged2e' support has been changed.\MessageBreak
607 Rerun to get captions right}%
608 \global\let\caption@ragged\@secondoftwo % suppress further warnings
609 \caption@ragged}%
610 }{%
611 \caption@Debug{We load ragged2e}%
612 \IfFileExists{ragged2e.sty}{%
613 \RequirePackage{ragged2e}%
614 \let\caption@@ragged\@firstoftwo
615 }{%
616 \newcommand*\caption@@ragged[2]{%
617 \@ifundefined{caption\string#1}{%
618 \caption@Warning{%

```

```

619         'ragged2e' package not loaded, therefore\MessageBreak
620         substituting \string#2 for \string#1\MessageBreak}%
621         \global\@namedef{caption\string#1}}{}%
622     #2}%
623 }%
624 }}

```

## 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.

```

625 \@ifundefined{abovecaptionskip}{%
626     \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
627 \@ifundefined{belowcaptionskip}{%
628     \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

629 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
630 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
631 \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.

```

632 \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.*

```

633 \newcommand*\caption@setposition[1]{%
634     \caption@ifinlist{#1}{d,default}{%
635         \let\caption@position\caption@defaultpos
636     }{\caption@ifinlist{#1}{t,top,above}{%
637         \let\caption@position\@firstoftwo
638     }{\caption@ifinlist{#1}{b,bottom,below}{%
639         \let\caption@position\@secondoftwo
640     }{\caption@ifinlist{#1}{a,auto}{%
641         \let\caption@position\@undefined
642     }{%
643         \caption@Error{Undefined position '#1'}%
644     }}}}}

```

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

The setting 'bottom' correspondents to the `\@makecaption` implementation in the standard  $\text{\LaTeX}$  document classes, but 'auto' should give better results in most cases.

```

645 %\caption@setdefaultpos{a}% default = auto
646 \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.)
647 \newcommand*\caption@iftop{%
648   \ifx\caption@position\@undefined
649     \let\caption@position\@secondoftwo
650 %   = \caption@setposition b%
651   \fi
652   \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.
653 \newcommand*\caption@fixposition{%
654   \ifx\caption@position\@undefined
655     \caption@autoposition
656   \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.
657 \newcommand*\caption@autoposition{%
658   \ifvmode
659     \edef\caption@tempa{\the\prevdepth}%
660     \caption@Debug{\protect\prevdepth=\caption@tempa}%
661     \ifdim\prevdepth>-\p@
662       \let\caption@position\@secondoftwo
663     \else
664       \let\caption@position\@firstoftwo
665     \fi
666 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
667   \else
668     \caption@Debug{no \protect\prevdepth}%
669     \let\caption@position\@secondoftwo
670 %   = \caption@setposition b%
671   \fi}

\caption@setautoposition \caption@setautoposition{<position>}
replaces the above algorithm by a different one (or a fixed position setting).
672 \newcommand*\caption@setautoposition[1]{%
673   \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`.

```
674 \newcommand*\caption@beginhook{}
675 \newcommand*\caption@endhook{}
676 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
677 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}
```

## 1.17 Lists

```
678 \DeclareCaptionOption{listof}{\caption@set@bool\caption@iflof{#1}}
\DeclareCaptionListFormat{<name>}{<code with #1 and #2>}
679 \newcommand*\DeclareCaptionListFormat[2]{%
680   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
681 \onlypreamble\DeclareCaptionListFormat

682 \DeclareCaptionOption{listofformat}{\caption@setlistofformat{#1}}
\caption@setlistofformat \caption@setlistofformat{<name>}
```

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

```
683 \newcommand*\caption@setlistofformat[1]{%
684   \@ifundefined{caption@lstfmt@#1}%
685     {\caption@Error{Undefined list-of format `#1'}}%
686     {\expandafter\let\expandafter\caption@lstfmt
687       \csname caption@lstfmt@#1\endcsname}}
```

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

```
688 \DeclareCaptionListFormat{empty}{}
689 \DeclareCaptionListFormat{simple}{#1#2}
690 \DeclareCaptionListFormat{parens}{#1(#2)}
691 \DeclareCaptionListFormat{subsimple}{#2}
692 \DeclareCaptionListFormat{subparens}{(#2)}
693 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}
```

## 1.18 Debug option

```
694 \DeclareCaptionOption{debug}[1]{%
695   \caption@set@bool\caption@ifdebug{#1}%
696   \let\caption@iftypewarning\caption@ifdebug
697   \caption@ifdebug
698     {\def\caption@Debug{\PackageInfo{caption}}}%
699     {\let\caption@Debug@gobble}}
700 \DeclareOption{debug}{\setkeys{caption}{debug}}
701 \setkeys{caption}{debug=0}
```

## 1.19 Document classes & Babel support

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

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

```
703 \caption@CheckCommand\makecaption{%
704   % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
705   \long\def\makecaption#1#2{%
706     \vskip\abovecaptionskip
```



```

707 \sbox\@tempboxa{#1: #2}%
708 \ifdim \wd\@tempboxa >\hsize
709 #1: #2\par
710 \else
711 \global \@minipagefalse
712 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
713 \fi
714 \vskip\belowcaptionskip}}

```

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

```

715 \@ifundefined{@captionheadfont}{}{%
716 \caption@CheckCommand\@makecaption{%
717 % amsart|amsproc|amsbook [2004/08/06 v2.20]
718 \long\def\@makecaption#1#2{%
719 \setbox\@tempboxa\vbox{\color@setgroup
720 \advance\hsize-2\captionindent\noindent
721 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
722 {\@cdr#2\@nil}\@captionfont\upshape\enspace#2}%
723 \unskip\kern-2\captionindent\par
724 \global\setbox\@ne\lastbox\color@endgroup}%
725 \ifhbox\@ne % the normal case
726 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
727 \fi
728 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
729 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
730 \else % tempboxa contained more than one line
731 \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
732 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
733 \fi
734 \ifnum\@tempcnta<64 % if the float IS a figure...
735 \addvspace\abovecaptionskip
736 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
737 \else % if the float IS NOT a figure...
738 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
739 \nobreak
740 \vskip\belowcaptionskip
741 \fi
742 \relax
743 }}

744 \caption@CheckCommand\@makecaption{%
745 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
746 \long\def\@makecaption#1#2{%
747 \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
748 \setbox\@tempboxa\vbox{\color@setgroup
749 \advance\hsize-2\captionindent\noindent
750 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
751 {\@cdr#2\@nil}\@addpunct{.}\@captionfont\upshape\enspace#2}%
752 \unskip\kern-2\captionindent\par
753 \global\setbox\@ne\lastbox\color@endgroup}%
754 \ifhbox\@ne % the normal case
755 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
756 \fi
757 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line

```

```

758     \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
759     \@tempdima\wd\@ne\advance\@tempdima-\captionindent
760     \wd\@ne\@tempdima
761   \else % tempboxa contained more than one line
762     \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
763       \unvbox\@tempboxa\parskip\z@skip
764       \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
765   \fi
766   \ifnum\@tempcnta<64 % if the float IS a figure...
767     \addvspace\abovecaptionskip
768     \noindent\kern\captionindent\box\@ne
769   \else % if the float IS NOT a figure...
770     \noindent\kern\captionindent\box\@ne
771     \nobreak
772     \vskip\belowcaptionskip
773   \fi
774   \relax
775 }

776 \let\captionmargin\captionindent % set to 3pc by AMS class
777 \begingroup\edef\@tempa{\endgroup
778   \noexpand\caption@g@addto@list\noexpand\caption@sty@default
779   {margin=\the\captionmargin
780     \@ifundefined{smf@makecaption}{},{,maxmargin=.1\linewidth}}}
781 \@tempa
782 \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
783 \DeclareCaptionLabelSeparator{default}{.\enspace}
784 \DeclareDefaultCaptionFont{font}{\@captionfont}
785 \DeclareDefaultCaptionFont{labelfont}{\@captionheadfont}
786 \DeclareDefaultCaptionFont{textfont}{\@captionfont\upshape}
787 \captionsetup*[figure]{position=b}
788 \captionsetup*[table]{position=t}
789 }

```

### 1.19.3 The beamer class

```

790 \@ifclassloaded{beamer}{%
791   \caption@CheckCommand\beamer@makecaption{%
792     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
793     \long\def\beamer@makecaption#1#2{%
794       \def\insertcaptionname{\csname#1name\endcsname}%
795       \def\insertcaptionnumber{\csname the#1\endcsname}%
796       \def\insertcaption{#2}%
797       \nobreak\vskip\abovecaptionskip\nobreak
798       \sbox\@tempboxa{\usebeamertemplate**{caption}}%
799       \ifdim \wd\@tempboxa >\hsize
800         \usebeamertemplate**{caption}\par
801       \else
802         \global \@minipagefalse
803         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
804       \fi
805       \nobreak\vskip\belowcaptionskip\nobreak}}
806   \DeclareCaptionLabelFormat{default}{#1}
807   \DeclareDefaultCaptionJustification{\raggedright}

```

```

808 \DeclareDefaultCaptionFont{font}{%
809   \usebeamerfont*{caption}%
810   \usebeamercolor[fg]{caption}}
811 \DeclareDefaultCaptionFont{labelfont}{%
812   \usebeamercolor[fg]{caption name}%
813   \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.)

```

814 \defbeamertemplate{caption}{caption3}{%
815   \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
816 \DeclareOption{beamer}{%
817   % \usebeamertemplate*{caption} will set font
818   \DeclareDefaultCaptionFont{font}{}%
819   \setbeamertemplate{caption}[caption3]}
820 }{}

```

#### 1.19.4 The KOMA-Script classes

```

821 \@ifundefined{scr@caption}{}{%
822   \caption@CheckCommand\@makecaption{%
823     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
824     \long\def\@makecaption#1#2{%
825       \if@captionabove
826         \vskip\belowcaptionskip
827       \else
828         \vskip\abovecaptionskip
829       \fi
830       \@@makecaption\@firstofone{#1}{#2}%
831       \if@captionabove
832         \vskip\abovecaptionskip
833       \else
834         \vskip\belowcaptionskip
835       \fi}}
836 \DeclareCaptionFormat{default}[#1#2#3\par]{%
837   \ifdofullc@p
838     \caption@ifin@list\caption@lsep@crlist\caption@lsep@name
839     {\caption@Error{%
840       The option ‘labelsep=\caption@lsep@name’ does not work\MessageBreak
841       with \noexpand\setcaphanging (which is set by default)}}%
842     {\caption@fmt@hang{#1}{#2}{#3}}%
843   \else
844     #1#2%
845     \ifdim\cap@indent<\z@
846       \par
847       \noindent\hspace*{-\cap@indent}%
848     \else\if@capbreak
849       \par
850     \fi\fi
851     #3\par
852   \fi}
853 \DeclareCaptionLabelSeparator{default}{\captionformat}
854 \DeclareDefaultCaptionFont{font}{\scr@fnt@caption}

```

```

855 \DeclareDefaultCaptionFont{labelfont}{\scr@fnt@captionlabel}
856 }

```

### 1.19.5 The NTG Dutch classes

```

857 \@ifundefined{CaptionFonts}{}{%
858 \caption@CheckCommand\@makecaption{%
859 % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
860 \long\def\@makecaption#1#2{%
861 \vskip\abovecaptionskip
862 \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
863 \ifdim \wd\@tempboxa >\hsize
864 {\CaptionLabelFont#1:} \CaptionTextFont#2\par
865 \else
866 \global \@minipagefalse
867 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
868 \fi
869 \vskip\belowcaptionskip}}
870 \DeclareDefaultCaptionFont{labelfont}{\CaptionLabelFont}
871 \DeclareDefaultCaptionFont{textfont}{\CaptionTextFont}
872 }

```

### 1.19.6 The frenchb Babel option

```

873 \@ifundefined{FB@makecaption}{}{%
874 \caption@CheckCommand\@makecaption{%
875 % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
876 \long\def\@makecaption#1#2{%
877 \vskip\abovecaptionskip
878 \sbox\@tempboxa{#1\CaptionSeparator #2}%
879 \ifdim \wd\@tempboxa >\hsize
880 #1\CaptionSeparator #2\par
881 \else
882 \global \@minipagefalse
883 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
884 \fi
885 \vskip\belowcaptionskip}}
886 \ifx\@makecaption\STD@makecaption
887 \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
888 \def\caption@frenchb{% supress frenchb warning
889 \let\STD@makecaption\@makecaption
890 \let\FB@makecaption\@makecaption}
891 \else
892 \ifx\@makecaption\@undefined\else
893 \PackageInfo{caption}{%
894 The definition of \protect\@makecaption\space
895 has been changed,\MessageBreak
896 frenchb will NOT customize it}%
897 \fi
898 \fi
899 }

```

### 1.19.7 The frenchle/pro package

```

900 \ifundefined{frenchTeXmods}{}{%
901   \caption@CheckCommand\@makecaption{%
902     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
903     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
904     \def\@makecaption#1#2{%
905       \ifFTY%
906         \def\@secondofmany##1##2\void{##2}%
907         \def\@tempa{\@secondofmany#2\void}%
908         \ifx\@tempa\empty%
909           \let\captionseparator\empty%
910         \fi%
911         \@mcORI{#1}{\relax\captionfont{#2}}%
912       \else
913         \@mcORI{#1}{#2}%
914       \fi}}
915   \caption@CheckCommand\@makecaption{%
916     % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
917     \def\@makecaption#1#2{%
918       \ifFTY%
919         \def\@secondofmany##1##2\void{##2}%
920         \protected@edef\@tempa{\@secondofmany#2\void}%
921         \ifx\@tempa\empty%
922           \let\captionseparator\empty%
923         \fi%
924         \@mcORI{#1}{\relax\captionfont{#2}}%
925       \else
926         \@mcORI{#1}{#2}%
927       \fi}}
928   \DeclareDefaultCaptionFont{textfont}{\itshape}%
929   \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
930 }

```

## 1.20 Execution of options

```

931 \captionsetup{style=default,position=default,listof=1,listofformat=default,%
932               twoside=\if@twoside 1\else 0\fi}
933 \ProcessOptions*
934 \caption@IfCheckCommand{%
935   \caption@setbool{documentclass}{1}%
936 }{%
937   \caption@setbool{documentclass}{0}%
938   \PackageInfo{caption}{%
939     Unknown document class (or package), \MessageBreak
940     standard defaults will be used}%
941   \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
942 }

```

## 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`.

```

943 \newcommand*\caption@addcontentsline[2]{%
944   \caption@iflof%
945     {\def\@tempa{#2}}%
946     {\let\@tempa\@empty}%
947   \ifx\@tempa\@empty \else
948     {\let\\space
949       \addcontentsline{\csname ext@#1\endcsname}{#1}%
950                       {\protect\numberline
951                       {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
952                       {\ignorespaces #2}}}%
953   \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.)
954 \newif\ifcaption@star

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

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

\caption@@make \caption@@make{<caption label>}{<caption text>}
959 \newcommand\caption@@make[2]{%
960   \begingroup
961   \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
962   \ifx\caption@maxmargin\undefined \else
963     \ifdim\captionmargin>\caption@maxmargin\relax
964       \captionmargin\caption@maxmargin\relax
965     \fi
966   \fi
967   \ifx\caption@minmargin\undefined \else
968     \ifdim\captionmargin<\caption@minmargin\relax
969       \captionmargin\caption@minmargin\relax
970     \fi
971   \fi

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

Typeset the left margin (option margin=)
973   \caption@calcmargin
974   \@tempdima\captionmargin
975   \ifdim\captionmargin@=\z@ \else
976     \caption@ifoddpagel{\advance\@tempdima\captionmargin@}%
977   \fi
978   \caption@ifh{\advance\@tempdima\caption@indent}%
979   \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*`.

```
980 \@tempdima\captionwidth
981 \caption@ifh{\advance\@tempdima-\caption@indent}%
982 \captionbox\@tempdima{%
```

Typeset the indentation (option `indention=`)

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

```
983 \caption@ifh{%
984 \ifdim\caption@indent=\z@
985 \leavevmode
986 \else
987 \hskip-\caption@indent
988 \fi}%
```

Typeset the caption itself and close the `\captionbox`

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

Typeset the right margin (option `margin=`)

```
990 \@tempdima\captionmargin
991 \ifdim\captionmargin@=\z@ \else
992 \caption@ifoddpages{\advance\@tempdima\captionmargin@}{}%
993 \fi
994 \hspace\@tempdima
995 \caption@endhook
996 \endgroup
997 \global\caption@starfalse}
```

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

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

```
998 \newcommand*\caption@calcmargin{%
999 \ifdim\captionwidth=\z@
1000 \captionwidth\linewidth
1001 \advance\captionwidth by -2\captionmargin
1002 \advance\captionwidth by -\captionmargin@
1003 \else
1004 \captionmargin\linewidth
1005 \advance\captionmargin by -\captionwidth
1006 \divide\captionmargin by 2
1007 \captionmargin@\z@
1008 \fi
1009 \caption@Debug{%
1010 \string\hsize=\the\hsize,
1011 \string\linewidth=\the\linewidth, \MessageBreak
1012 \string\leftmargin=\the\leftmargin,
1013 \string\rightmargin=\the\rightmargin, \MessageBreak
1014 \string\margin=\the\captionmargin,
1015 \string\margin@=\the\captionmargin@,
1016 \string\width=\the\captionwidth}%
1017 }
```

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

**This one does the single-line-check.**

```

1018 \newcommand\caption@slc[4]{%
1019   \caption@Debug{Begin SLC}%
1020   \begingroup
1021   \caption@singleline
1022   \let\caption@hj\@empty
1023   \caption@calcmargin % calculate #3 if necessary
1024   \caption@prepareslc
1025   \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1026   \ifdim\wd\@tempboxa>#3%
1027     \endgroup
1028   \else
1029     \endgroup
1030     \caption@singleline
1031     #4%
1032   \fi
1033   \caption@Debug{End SLC}}
1034 \newcommand*\caption@singleline{%
1035   \expandafter\caption@setup\expandafter{\caption@opt@singleline}%
1036   \let\caption@fmt\caption@slfmt}

```

`\caption@prepareslc` `\caption@prepareslc`

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

```

1037 \newcommand*\caption@prepareslc{%
1038   \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1039   \def\label{\caption@withoptargs\@gobbletwo}%
1040   \let\stepcounter\caption@l@stepcounter
1041   \let\refstepcounter\stepcounter\let\H\refstepcounter\stepcounter}
1042 \newcommand*\caption@l@stepcounter[1]{\advance\c@#1\endcsname\@ne\relax}

```

`\captionbox` `\captionbox{<width>}{<contents>}`

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

```

1043 \newcommand*\captionbox{\parbox[b]}

```

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

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

```

1044 \newcommand\caption@@@make[2]{%

```

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

```

1045   \sbox\@tempboxa{#1}%
1046   \ifdim\wd\@tempboxa=\z@
1047     \let\caption@lsep\relax
1048 %   \@capbreakfalse
1049   \fi

```

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

```

1050   \caption@ifempty{#2}{%
1051     \let\caption@lsep\relax
1052 %   \@capbreakfalse
1053 %   \let\caption@ifstrut\@secondoftwo
1054 }%

```



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

```
1055 \setpar{\@@par\caption@@par}\caption@@par
```

Finally typeset the caption.

```
1056 \caption@hj\captionsize\captionfont\caption@fmt
1057 {\ifcaption@star\else{\captionlabelfont#1}\fi}%
1058 {\ifcaption@star\else{\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1059 {\captiontextfont
1060 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1061 \nobreak\hskip\z@skip % enable hyphenation
1062 \caption@tfmt{#2}%
1063 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1064 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1065 \par}}}
```

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

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

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

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

```
1066 \newcommand\caption@ifempty[1]{%
1067 \def\caption@tempa{#1}%
1068 \def\caption@tempb{\ignorespaces}%
1069 \ifx\caption@tempa\caption@tempb
1070 \let\caption@tempa\@empty
1071 \fi
1072 \ifx\caption@tempa\@empty
1073 \let\caption@ifempty\@secondoftwo
1074 \else
1075 \let\caption@ifempty\@gobbletwo
1076 \fi
1077 \caption@ifempty\@unused}
```

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

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

```
1078 \newcommand*\caption@@par{%
1079 \parindent\caption@parindent\hangindent\caption@hangindent}%

```

## 1.22 Bonus features

```
eclareFloatingEnvironment \DeclareFloatingEnvironment[<file ext>]{<environment>}[<name>][<list name>]
1080 \providecommand*\DeclareFloatingEnvironment{%
1081 \ifnextchar[%]
1082 \caption@@DeclareFloatingEnvironment\caption@DeclareFloatingEnvironment}
1083 \@onlypreamble\DeclareFloatingEnvironment

1084 \newcommand*\caption@DeclareFloatingEnvironment[1]{%
1085 \caption@@DeclareFloatingEnvironment[lo#1]{#1}}
1086 \@onlypreamble\caption@DeclareFloatingEnvironment

1087 \def\caption@@DeclareFloatingEnvironment[#1]#2{%
1088 \ifundefined{c@chapter}%
1089 {\newcounter{#2}%

```

```

1090 \@namedef{the#2}{\arabic{#2}}%
1091 \@namedef{theH#2}{\arabic{#2}}}%
1092 {\newcounter{#2}[chapter]%
1093 \@namedef{the#2}{\ifnum\c@chapter>\z@ \thechapter.\fi \arabic{#2}}%
1094 \@namedef{theH#2}{\theHchapter.\arabic{#2}}}%
1095 \@namedef{fps@#2}{tbp}%
1096 \@ifundefined{c@float@type}%
1097 {\newcounter{float@type}%
1098 \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1099 {}}%
1100 \caption@Debug{float type `#2'=\the\value{float@type}}%
1101 \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1102 \addtocounter{float@type}{\value{float@type}}%
1103 \@namedef{ext@#2}{#1}%
1104 \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1105 \newenvironment{#2}{\@float{#2}}{\end@float}%
1106 \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1107 \expandafter\newcommand\csname listof#2s\endcsname{\begin@group
1108 \expandafter\let\expandafter\listfigurename\csname list#2name\endcsname
1109 \expandafter\let\expandafter\ext@figure\csname ext@#2\endcsname
1110 \let\caption@ORI@starttoc\@starttoc
1111 \renewcommand*\@starttoc[1]{%
1112 \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1113 \listoffigures
1114 \endgroup}%
1115 \@ifundefined{l@figure}%
1116 {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1117 {\expandafter\let\csname l@#2\endcsname\l@figure}%
1118 \expandafter\newcommand\csname #2name\endcsname{}%
1119 \edef\@tempa{%
1120 \def\noexpand\@tempa{\expandafter\noexpand\@car#2\@nil}}%
1121 \uppercase\expandafter{\@tempa}%
1122 \edef\@tempb{%
1123 \noexpand\g@addto@macro\noexpand\@tempa{\expandafter\@cdr#2\@nil}}%
1124 \@tempb
1125 \expandafter\let\csname #2name\endcsname\@tempa
1126 \expandafter\newcommand\csname list#2name\endcsname{}%
1127 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1128 \@cons\caption@floatlist{{#2}}%
1129 \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#2}}{\relax}
1130 \@onlypreamble\caption@@DeclareFloatingEnvironment
1131 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1132 \@namedef{#1name}{#2}%
1133 \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#1}}{\relax}
1134 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1135 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1136 \@namedef{list#1name}{#2}}
1137 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1138 \newcommand*\caption@floatlist{}
\ForFloatingEnvironments \ForFloatingEnvironments{<code with #1>}
1139 \providecommand\ForFloatingEnvironments[1]{%
1140 \caption@AtBeginDocument{%

```

```

1141 \def\@elt##1{#1}%
1142 \caption@floatlist
1143 \let\@elt\relax}%
1144 \@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.

```

1145 \caption@AtBeginDocument{%
1146 \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1147 \ifx\subref\@tempa
1148 \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1149 \let\caption@setfloattype\@gobble
1150 %
1151 \let\sf@ifinlist\caption@ifinlist
1152 \let\sf@ifpositiontop\caption@iftop
1153 \def\@dottedxxxline#1#2#3#4#5#6#7{%
1154 \begingroup
1155 \caption@setfloattype{#1}%
1156 \caption@setoptions{subfloat}%
1157 \caption@setoptions{sub#1}%
1158 \ifnum #3>\@nameuse{c@#2depth}\else
1159 \dottedtocline{#4}{#5}{#6}{#7}%
1160 \fi
1161 \endgroup}%
1162 \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.

```

1162 \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1163 \caption@removefromoptlist{subfloat}%
1164 \caption@removefromoptlist{subfigure}%
1165 \caption@removefromoptlist{subtable}%
1166 \fi
1167 \let\sf@NEW@subfloat\@undefined}
1168 \def\sf@NEW@subfloat{%
1169 \begingroup
1170 \caption@setfloattype\@capttype
1171 \sf@ifpositiontop{%
1172 \maincaptiontoptrue
1173 }{%
1174 \maincaptiontopfalse
1175 }%
1176 \caption@setoptions{subfloat}%
1177 \caption@setoptions{sub\@capttype}%
1178 \let\sf@oldlabel=\label
1179 \let\label=\subfloat@label
1180 \ifmaincaptiontop\else
1181 \advance\@nameuse{c@\@capttype}\@ne
1182 \fi
1183 \refstepcounter{sub\@capttype}%

```

```
1184 \setcounter{sub\@cuptype @save}{\value{sub\@cuptype}}%
1185 \@ifnextchar [% %] match left bracket
1186 {\sf@@subfloat}%
1187 {\sf@@subfloat[\@empty]}}%
```

## 2 Main package

### 2.1 Identification

```
1188 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1189 \ProvidesPackage{caption}[2007/10/25 v3.1d Customising captions (AR)]
1190 %\@ifundefined{PackageRedefines}{}{\PackageRedefines{caption}{caption}}

\caption@Info Note: The \@gobble at the end of the 2nd argument of \PackageInfo suppresses the line
number info. See TLC2[1], A.4.7, p885 for details.

1191 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1192 \@onlypreamble\caption@Info
```

### 2.2 Loading the kernel

```
1193 \RequirePackage{caption3}[2007/10/25] % needs v3.1d or newer
```

### 2.3 Check against incompatible packages

```
1194 \@ifpackageloaded{caption2}{%
1195   \caption@Error{%
1196     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1197     the (current) caption package}%
1198   \endinput
1199 }{}

1200 \caption@AtBeginDocument{%
1201   \@ifpackageloaded{ftcap}{%
1202     \caption@IncompatiblePackage{ftcap}{}%
1203   }{}%
1204   \@ifpackageloaded{nonfloat}{%
1205     \caption@IncompatiblePackage{nonfloat}{. %
1206       Furthermore you could use \string\captionof\MessageBreak
1207       instead of \noexpand\figcaption or \string\tabcaption}%
1208   }{}%
1209   \@ifpackageloaded{topcapt}{%
1210     \caption@IncompatiblePackage{topcapt}{}%
1211   }{}%
1212 }

\caption@Incompatible Issue an ‘incompatible’ warning.

1213 \newcommand*\caption@IncompatiblePackage[2]{%
1214   \PackageWarningNoLine{caption}{%
1215     Usage of the #1 package together with the\MessageBreak
1216     caption package is strongly *not* recommended.\MessageBreak
1217     Instead of loading the #1 package you should\MessageBreak
1218     use the caption package option\MessageBreak
1219     \space\space'tableposition=top'\MessageBreak
1220     #2}}

1221 \@onlypreamble\caption@IncompatiblePackage
```

### 2.4 Check document class

```
1222 \caption@ifbool{documentclass}{}{%
1223   \PackageWarningNoLine{caption}{%
```

```

1224     Unsupported document class (or package) detected,\MessageBreak
1225     usage of the caption package is *not* recommended}%
1226 \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1227 }

```

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

```

1228 \@ifundefined{@captionheadfont}{}{%
1229 \caption@Info{AMS or SMF document class}%
1230 \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1231 }

```

## 2.6 Emulation of the KOMA-Script commands

```

1232 \@ifundefined{scr@caption}{}{%
1233 \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 1234 \def\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1235 \def\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1236 \if@tablecaptionabove
1237 \caption@AtBeginDocument{%
1238 \def\caption@tempa#1{%
1239 \ifpackageloaded{#1}{\caption@Error{%
1240 You can't use the global option\MessageBreak
1241 \space\space`tablecaptionabove'\MessageBreak
1242 and use the #1 package, too}}}%
1243 \caption@tempa{ftcap}%
1244 \caption@tempa{nonfloat}%
1245 \caption@tempa{topcapt}}%
1246 \@tablecaptionabovetrue
1247 \else
1248 \@tablecaptionabovefalse
1249 \fi

\onelinecaptionstrue
\onelinecaptionsfalse 1250 \def\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1251 \def\onelinecaptionsfalse{\let\caption@ifslc\@secondoftwo}

1252 \ifonelinecaptions
1253 \onelinecaptionstrue
1254 \else
1255 \onelinecaptionsfalse
1256 \fi

\@captionabovetrue
\@captionabovefalse 1257 \def\@captionabovetrue{\let\caption@position\@firstoftwo}
1258 \def\@captionabovefalse{\let\caption@position\@secondoftwo}

\setcapindent
1259 \let\caption@KOMA@setcapindent\@setcapindent
1260 \renewcommand*\@setcapindent[1]{%
1261 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

```

```

1262 \let\caption@KOMA@@setcapindent\@@setcapindent
1263 \renewcommand*\@@setcapindent[1]{%
1264   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1265 \newcommand*\caption@setcapindent{%
1266   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1267 \@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.)
1268 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1269   \csname\string\setcapwidth\endcsname
1270 \@namedef{\string\setcapwidth}[1]#2{%
1271   \caption@KOMA@setcapwidth{#1}{#2}\caption@setcapwidth{#1}}
1272 \newcommand*\caption@setcapwidth[1]{%
1273   \ifx\#1\\\else
1274     \@ifundefined{cap@margin}{}{%
1275       \def\@tempa{captionbeside}%
1276       \ifx\@tempa\@currenvir\else\caption@Warning{%
1277         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1278       \fi}%
1279   \fi
1280   \captionsetup{width=\cap@width}}

1281 \def\caption@tempa{\hsize}%
1282 \ifx\caption@tempa\cap@width \else
1283   \caption@setcapwidth{?}
1284 \fi

\setcapmargin
1285 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1286   \csname\string\@setcapmargin\endcsname
1287 \@namedef{\string\@setcapmargin}[1]#2{%
1288   \caption@KOMA@setcapmargin{#1}{#2}\caption@setcapmargin}
1289 \expandafter\let\expandafter\caption@KOMA@@setcapmargin
1290   \csname\string\@@setcapmargin\endcsname
1291 \@namedef{\string\@@setcapmargin}[1]#2{%
1292   \caption@KOMA@@setcapmargin{#1}{#2}\caption@setcapmargin}
1293 \newcommand*\caption@setcapmargin{%
1294   \begingroup
1295     \let\onelinecaptionsfalse\relax
1296     \def\@twoside{0}%
1297     \def\if@twoside{\def\@twoside{1}\iffalse}%
1298     \cap@margin
1299     \def\@tempa{\endgroup}%
1300     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1301       \def\hspace##1##{\@firstofone}%
1302       \edef\@tempa{\endgroup
1303         \noexpand\captionsetup{%
1304           twoside=\@twoside,singlelinecheck=0,%
1305           margin={\cap@left,\cap@right}}}%
1306     \fi\fi
1307   \@tempa}

```

```

1308 \ifx\cap@margin\relax \else
1309 \caption@setcapmargin
1310 \fi
1311 }

```

## 2.7 Option for configuration files

```

1312 \DeclareCaptionOption*{config}[caption]{%
1313 \InputIfFileExists{#1.cfg}%
1314 {\typeout{*** Local configuration file #1.cfg used ***}}%
1315 {\caption@Warning{Configuration file #1.cfg not found}}

```

## 2.8 Options for figure and table

```

1316 \DeclareCaptionOption{figureposition}{%
1317 \captionsetup*[figure]{position=#1}}
1318 \@onlypreamble@key{caption}{figureposition}

1319 \DeclareCaptionOption{tableposition}{%
1320 \caption@AtBeginDocument{%
1321 \def\caption@tempa##1{%
1322 \ifpackageloaded{##1}{\caption@Error{%
1323 You can't use the caption package option\MessageBreak
1324 \space\space`tableposition=#1'\MessageBreak
1325 and use the ##1 package, too}}{}}}%
1326 \caption@tempa{ftcap}%
1327 \caption@tempa{nonfloat}%
1328 \caption@tempa{topcapt}}}%
1329 \captionsetup*[table]{position=#1}}
1330 \@onlypreamble@key{caption}{tableposition}

1331 \DeclareCaptionOption{figurename}{\caption@SetFloatName{figure}{#1}}
1332 \@onlypreamble@key{caption}{figurename}
1333 \DeclareCaptionOption{tablename}{\caption@SetFloatName{table}{#1}}
1334 \@onlypreamble@key{caption}{tablename}

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

1335 \newcommand*\caption@SetFloatName[2]{%
1336 \begingroup
1337 \def\@tempa{\endgroup\@firstofone}%
1338 \ifundefined{language\@tempa}{\@tempa}%
1339 \ifundefined{captions\language\@tempa}{\@tempa}%
1340 \def\@tempa{\endgroup
1341 \expandafter\addto\csname captions\language\@tempa\endcsname}}}%
1342 \@tempa{\@namedef{#1name}{#2}}}%
1343 \@onlypreamble\caption@SetFloatName

```

## 2.9 caption v1.x compatibility options

```

1344 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1345 \@onlypreamble@key{caption}{compatibility}

1346 \DeclareCaptionOptionNoValue*{normal}{%
1347 \caption@setformat{plain}%
1348 \caption@setjustification{justified}}
1349 \DeclareCaptionOptionNoValue*{isu}{%

```



```

1350 \caption@setformat{hang}%
1351 \caption@setjustification{justified}}
1352 \DeclareCaptionOptionNoValue*{hang}{%
1353 \caption@setformat{hang}%
1354 \caption@setjustification{justified}}
1355 \DeclareCaptionOptionNoValue*{center}{%
1356 \caption@setformat{plain}%
1357 \caption@setjustification{centering}}
1358 \DeclareCaptionOptionNoValue*{anne}{%
1359 \caption@setformat{plain}%
1360 \caption@setjustification{centerlast}}
1361 \DeclareCaptionOptionNoValue*{centerlast}{%
1362 \caption@setformat{plain}%
1363 \caption@setjustification{centerlast}}

1364 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1365 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1366 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1367 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1368 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1369 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1370 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1371 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1372 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1373 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1374 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1375 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1376 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1377 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1378 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1379 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}

1380 \caption@setbool{ruled}{0}
1381 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}

```

## 2.10 caption2 v2.x compatibility options

```

1382 \DeclareCaptionOptionNoValue*{flushleft}{%
1383 \caption@setformat{plain}%
1384 \caption@setjustification{raggedright}}
1385 \DeclareCaptionOptionNoValue*{flushright}{%
1386 \caption@setformat{plain}%
1387 \caption@setjustification{raggedleft}}

1388 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}

1389 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
1390 \caption@Warning{Obsolete option 'ignoreLTcapwidth' ignored}}

```

## 2.11 Obsolete caption v3.0 options

```

1391 \DeclareCaptionOption*{caption}{%
1392 \caption@setbool{temp}{#1}%
1393 \caption@ifbool{temp}{}{%
1394 \caption@Error{%
1395 The package option 'caption=#1' is obsolete.\MessageBreak
1396 Please pass this option to the subfig package instead\MessageBreak

```

```
1397         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.

```
1398 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1399 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1400 \@onlypreamble@key{caption}{FPlist}
1401 \@onlypreamble@key{caption}{FPref}

1402 \newcommand*\caption@setFPoption[2]{%
1403   \edef\@tempa{\@car#2\@nil}%
1404   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1405 \@onlypreamble\caption@setFPoption

1406 \captionsetup{FPlist=caption,FPref=figure}
```

## 2.13 hyperref package support options

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

```
1407 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1408 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}
1409 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

## 2.14 Processing of options

```
1410 \caption@ProcessOptions*{caption}
```

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

```
1411 \caption@AtBeginDocument{%
1412   \DeclareCaptionOption{type}{\caption@settype{#1}}
1413   \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 `\@capytype` and executes the options associated with it (using `\caption@setoptions`). Furthermore we check `\currentgrouplevel` (if avail), redefine `\@currentlabel` so a `\label` before `\caption` will result in a hint instead of a wrong reference, and use the macro `\caption@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).

```
1414 \newcommand*\caption@settype{%
1415   \caption@@settype{}}

1416 \newcommand*\caption@@settype[1]{%
1417   \caption@teststar{\caption@@@settype{#1}}{\@firstoftwo\@secondoftwo}
```

```

1418 \newcommand*\caption@@@settype[3]{%
1419 % #1 = "" or "sub"
1420 % #2 = \@firstoftwo in star form, \@secondoftwo otherwise
1421 % #3 = <type>, e.g. "figure" or "table"
1422 \caption@Debug{type=#3}%
1423 \caption@checkgrouplevel{#1}{%
1424   \captionsetup{#1type#2*\@empty=...}#2{ or \@backslashchar#1captionof}{}}%
1425 \@ifundefined{@#1capttype}{}{%
1426   \edef\@tempa{#3}%
1427   \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1428     \caption@iftypewarning{\caption@Warning{%
1429       Caption type was already set to `#3'\MessageBreak}}{}%
1430   \fi}%
1431 \expandafter\edef\csname @#1capttype\endcsname{#3}%
1432 \let\caption@iftypewarning\caption@ifdebug
1433 \@nameuse{#1caption@settypehook}%
1434 \expandafter\caption@setoptions\csname @#1capttype\endcsname
1435 \let\caption@ifrefstepcounter\@secondoftwo
1436 #2{}{%
1437   \let\@currentlabel\caption@undefinedlabel
1438 % \let\@currentHlabel\@undefined
1439   \ifx\caption@ORI@label\@undefined
1440     \let\caption@ORI@label\label
1441     \def\label{\caption@withoptargs\caption@xlabel}%
1442   \fi
1443   \caption@start}}

```

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

```
1444 \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.

```

1445 \newcommand*\caption@iftype{%
1446   \@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)

```

1447 \begingroup\expandafter\expandafter\expandafter\endgroup
1448 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1449   \caption@Debug{TeX engine: TeX}
1450   \let\caption@checkgrouplevel\@gobbletwo
1451 \else
1452   \caption@Debug{TeX engine: e-TeX}
1453   \newcommand*\caption@checkgrouplevel[2]{%
1454     \@ifundefined{#1caption@grouplevel}{%
1455       \@ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}}{%
1456       \ifnum\currentgrouplevel>\caption@grouplevel\relax
1457         \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1458           \the\currentgrouplevel}%
1459       \else
1460         \caption@Error{\string#2\MessageBreak outside box or environment}%
1461       \fi

```

```

1462     }{} }
1463 \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.
1464 \newcommand*\caption@undefinedlabel{??}

\caption@xlabel
1465 \newcommand*\caption@xlabel[2]{%
1466   \ifx\@currentlabel\caption@undefinedlabel
1467     \caption@Warning{\noexpand\label before \noexpand\caption ignored}%
1468   \else
1469     \caption@ORI@label#1{#2}%
1470   \fi}

\captionof \captionof{<type>}[<lst_entry>]{<heading>}
\captionof* [<lst_entry>]{<heading>}
1471 \def\captionof{\caption@teststar\caption@of{\caption*}\caption}
1472 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

\captionlistentry \captionlistentry[<float type>]{<list entry>}
1473 \newcommand*\captionlistentry{%
1474   \@testopt\caption@listentry\@capttype}
1475 \def\caption@listentry[#1]#2{%
1476   \caption@refstepcounter{#1}%
1477   \caption@makecurrent{#1}{#2}%
1478   \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.
Please forget about the optional argument, it was never working well, is incompatible to
the subfig package, but is still there for compatibility reasons.
Note: The definition of \ContinuedFloat itself is compatible to the one inside the subfig pack-
age, except for the optional argument.
1479 \def\ContinuedFloat{%
1480   \@ifnextchar[\@ContinuedFloat{%
1481     \caption@iftype
1482       {\addtocounter\@capttype\m@ne
1483       \caption@ContinuedFloat\@capttype}%
1484     {\caption@Error{\noexpand\ContinuedFloat outside float}}}}
1485 \def\caption@ContinuedFloat#1{%
1486   \edef\@tempa{#1}%
1487   \ifx\@tempa\caption@CFtype \else
1488     \caption@Error{Continued `#1' after `\caption@CFtype'}%
1489   \fi
1490   \stepcounter{ContinuedFloat}%
1491   \caption@@ContinuedFloat{#1}%
1492   \caption@setoptions{ContinuedFloat}%
1493   \caption@setoptions{continued#1}%
1494   \let\caption@resetContinuedFloat\@gobble}

```

```

1495 \def\@ContinuedFloat[#1]{\addtocounter{#1}\m@ne}
1496 \newcounter{ContinuedFloat}
1497 \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.)
1498 \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.)
1499 \newcommand*\caption@resetContinuedFloat[1]{%
1500 \stpelt{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.
1501 \newcommand*\caption@refstepcounter[1]{%
1502 \caption@resetContinuedFloat{#1}%
1503 \caption@@refstepcounter{#1}%
1504 \let\caption@ifrefstepcounter\@firstoftwo}

1505 \newcommand*\caption@@refstepcounter{\refstepcounter}
1506 \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[]{}).
1507 \newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1508 \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)

1509 \newcommand*\caption@begin[1]{%
1510 \begingroup
1511 \caption@setfnum{#1}%
1512 \caption@fixposition
1513 \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 $\langle heading \rangle$ .

```
1514 \newcommand\caption@beginex[3]{%
1515   \caption@begin{#1}%
1516   \caption@addcontentsline{#1}{#2}%
1517   \caption@ifempty{#3}{} }
```

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

```
1518 \newcommand*\caption@end{%
1519   \endgroup
1520   \let\caption@position\caption@fixedposition}
```

`\caption@setfnum` `\caption@setfnum{ $\langle type \rangle$ }`

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

```
1521 \newcommand*\caption@setfnum[1]{%
1522   \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1523   \namedef{fnum@#1}{\caption@fnum{#1}}%
1524   \fi}
```

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

```
1525 \caption@AtBeginDocument*{%
1526   \@ifundefined{sf@ifpositiontop}{}{%
1527     \let\caption@kernel@addcontentsline\caption@addcontentsline
1528     \renewcommand*\caption@addcontentsline[2]{%
1529       \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
1530       \caption@kernel@addcontentsline{#1}{#2}%
1531       \sf@ifpositiontop{}{\@listsubcaptions{#1}}}%
1532   }}
```

`\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`.

```
1533 \newcommand*\caption@boxrestore{%
1534   \let\if@nobreak\iffalse
1535   \let\if@noskipsec\iffalse
```

```

1536 \let\par\@@par
1537 % \let\-\@dischyph
1538 % \let'\@acci\let'\@accii\let\=\@acciii
1539 \parindent\z@ \parskip\z@skip
1540 \everypar{}%
1541 % \linewidth\hsize
1542 % \@totalleftmargin\z@
1543 \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1544 \parfillskip\@flushglue \lineskip\normallineskip
1545 \baselineskip\normalbaselineskip
1546 \sloppy
1547 \let\\\@normalcr
1548 }

```

`\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).

```

1549 \newcommand*\caption@normalsize{%
1550 \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.

```
1551 \let\caption@setfloatcapt\@firstofone
```

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

```

\caption@@anchor 1552 \newcommand*\caption@makecurrent[2]{}
\caption@start    1553 \let\caption@@anchor\@firstofone
\caption@@start   1554 \let\caption@start\relax
\caption@freezeHref 1555 \let\caption@@start\relax
\caption@defrostHref 1556 \let\caption@freezeHref\relax
                  1557 \let\caption@defrostHref\relax

```

## 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.

```

1558 \caption@AtBeginDocument{%
1559 \caption@setbool{incompatible}{0}%
1560 \caption@CheckCommand\caption{%
1561 % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1562 \def\caption{%
1563 \ifx\@capttype\@undefined

```

```

1564         \@latex@error{\noexpand\caption outside float}\@ehd
1565         \expandafter\@gobble
1566     \else
1567         \refstepcounter\@capttype
1568         \expandafter\@firstofone
1569     \fi
1570     {\@dblarg{\@caption\@capttype}}%
1571 }}%

1572 \caption@CheckCommand\caption{%
1573 % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1574 \def\caption{
1575     \ifx\@capttype\@undefined
1576         \@latex@error{\noexpand\caption outside figure or table}\@ehd
1577         \expandafter\@gobble
1578     \else
1579         \refstepcounter\@capttype
1580         \expandafter\@firstofone
1581     \fi
1582     {\@dblarg{\@caption\@capttype}}%
1583 }}%

1584 \caption@CheckCommand\caption{%
1585 % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1586 \renewcommand\caption{%
1587     \ifx\@capttype\@undefined
1588         \@latex@error{\noexpand\caption outside float}\@ehd
1589         \expandafter\@gobble
1590     \else
1591         \refstepcounter\@capttype
1592         \let\@tempf\@caption
1593         \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1594             \expandafter\expandafter\let
1595             \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1596         \fi
1597     \fi
1598     \@dblarg{\@tempf\@capttype}}}%

1599 \caption@CheckCommand\caption{%
1600 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1601 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1602 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1603 \def\caption{%
1604     \ifx\@capttype\@undefined
1605         \@latex@error{\noexpand\caption outside float}\@ehd
1606         \expandafter\@gobble
1607     \else
1608         \H@refstepcounter\@capttype
1609         \@ifundefined{fst@\@capttype}{%
1610             \let\Hy@tempa\@caption
1611         }{%
1612             \let\Hy@tempa\Hy@float@caption
1613         }%
1614         \expandafter\@firstofone
1615     \fi
1616     {\@dblarg{\Hy@tempa\@capttype}}%

```



```

1617     }}%
1618 \caption@CheckCommand\caption{%
1619   % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1620   \def\caption{%
1621     \ifx\@capttype\@undefined
1622       \@latex@error{\noexpand\caption outside float}\@ehd
1623       \expandafter\@gobble
1624     \else
1625       \H@refstepcounter\@capttype
1626       \let\Hy@tempa\@caption
1627       \@ifundefined{float@caption}{%
1628         }{%
1629           \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
1630             \let\Hy@tempa\Hy@float@caption
1631           \fi
1632         }%
1633       \expandafter\@firstofone
1634       \fi
1635       {\@dblarg{\Hy@tempa\@capttype}}}%
1636   }}%

1637 \caption@IfCheckCommand{}{%
1638   \caption@Info{%
1639     Incompatible package detected (regarding \string\caption).\MessageBreak
1640     \string\caption\space=\space\meaning\caption}%
1641   \caption@setbool{incompatible}{1}}%

1642 \caption@CheckCommand\@caption{%
1643   % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1644   \long\def\@caption#1[#2]#3{%
1645     \par
1646     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1647     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1648     \begingroup
1649     \@parboxrestore
1650     \if@minipage
1651       \setminipage
1652     \fi
1653     \normalsize
1654     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1655     \endgroup}}%

1656 \caption@CheckCommand\@caption{%
1657   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1658   \long\def\@caption#1[#2]#3{% second argument ignored
1659     \par\nobreak
1660     \begingroup
1661     \@parboxrestore
1662     \if@minipage
1663       \setminipage
1664     \fi
1665     \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
1666     \endgroup}}%

1667 % \caption@CheckCommand\float@caption{%
1668 %   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]

```

```

1669 % \long\def\float@caption#1[#2]#3{%
1670 % \addcontentsline{\@nameuse{ext@#1}}{#1}%
1671 % {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
1672 % \global\setbox\@floatcapt\ vbox\bgroup\@parboxrestore
1673 % \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
1674 % \@ifnextchar[{\float@ccon}{\egroup}}%
1675 % \long\def\float@ccon[#1]{#1\par\egroup}}%

1676 \caption@CheckCommand\@caption{%
1677 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1678 \long\def\@caption#1[#2]#3{%
1679 \hyper@makecurrent{\@capttype}%
1680 \def\@currentlabelname{#2}%
1681 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1682 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1683 }%
1684 \begingroup
1685 \@parboxrestore
1686 \if@minipage
1687 \setminipage
1688 \fi
1689 \normalsize
1690 \makecaption{\csname fnum@#1\endcsname}{%
1691 \ignorespaces
1692 \ifHy@nesting
1693 \hyper@@anchor{\@currentHref}{#3}%
1694 \else
1695 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1696 \fi
1697 }%
1698 \par
1699 \endgroup
1700 }%

1701 \caption@CheckCommand\@caption{%
1702 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1703 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1704 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1705 \long\def\@caption#1[#2]#3{%
1706 \expandafter\ifx\csname if@capstart\expandafter\endcsname
1707 \csname iftrue\endcsname
1708 \global\let\@currentHref\hc@currentHref
1709 \else
1710 \hyper@makecurrent{\@capttype}%
1711 \fi
1712 \def\@currentlabelname{#2}%
1713 \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1714 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1715 }%
1716 \begingroup
1717 \@parboxrestore
1718 \if@minipage
1719 \setminipage
1720 \fi
1721 \normalsize

```

```

1722     \expandafter\ifx\csname if@capstart\expandafter\endcsname
1723         \csname iftrue\endcsname
1724     \global\@capstartfalse
1725     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1726 \else
1727     \@makecaption{\csname fnum@#1\endcsname}{%
1728         \ignorespaces
1729         \ifHy@nesting
1730             \hyper@@anchor{\@currentHref}{#3}%
1731         \else
1732             \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1733         \fi
1734     }%
1735 \fi
1736 \par
1737 \endgroup
1738 }%
1739 \caption@CheckCommand\@caption{%
1740 % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1741 \long\def\@caption#1[#2]{%
1742     \def\@currentlabelname{#2}%
1743     \NR@@caption{#1}[{#2}]%
1744 }%
1745 \caption@CheckCommand\@caption{%
1746 % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1747 \long\def\@caption#1[#2]#3{%
1748     \ifundefined{if#1topcap}%
1749     {\subfig@oldcaption{#1}[{#2}]{#3}}%
1750     {\@nameuse{if#1topcap}%
1751         \@listsubcaptions{#1}%
1752         \subfig@oldcaption{#1}[{#2}]{#3}%
1753     \else
1754         \subfig@oldcaption{#1}[{#2}]{#3}%
1755         \@listsubcaptions{#1}%
1756     \fi}}%
1757 \caption@CheckCommand\@caption{%
1758 % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1759 \def\@caption{\caption@}%
1760 % \long\def\caption@#1[#2]#3{%
1761 %     \ifundefined{caption@setfloattype}%
1762 %     \caption@settype
1763 %     \caption@setfloattype
1764 %     \capttype
1765 %     \sf@ifpositiontop{%
1766 %     \@listsubcaptions{#1}%
1767 %     \sf@old@caption{#1}[{#2}]{#3}%
1768 %     }{%
1769 %     \sf@old@caption{#1}[{#2}]{#3}%
1770 %     \@listsubcaptions{#1}%
1771 %     }%
1772 % }%
1773 \caption@IfCheckCommand{}{%
1774 \caption@Info{%

```

```

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

```

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

```

1778     \@ifundefined{caption@ifcompatibility}%
1779     {\let\caption@ifcompatibility\caption@ifincompatible
1780     \let\@tempa\@firstofone}%
1781     {\let\@tempa\@gobble}% suppress warning
1782     \caption@ifcompatibility{%
1783     \@tempa{%
1784     \caption@WarningNoLine{%
1785     \noexpand\caption will not be redefined since it's already\MessageBreak
1786     redefined by a document class or package which is\MessageBreak
1787     unknown to the caption package.\MessageBreak
1788     As a result some new features of the caption\MessageBreak
1789     package will not work as expected}%
1790     }%

```

**\ContinuedFloat is not supported in compatibility mode.**

```

1791     \renewcommand*\caption@ContinuedFloat[1]{%
1792     \caption@Error{Not available in compatibility mode}}%

```

**\caption@start is not supported in compatibility mode.**

```

1793     \caption@AtBeginDocument*{%
1794     \let\caption@start\relax
1795     \@ifundefined{caption@ORI@capstart}{}{%
1796     \caption@Debug{%
1797     Undo redefinition of \string\capstart\@gobble}%
1798     \let\capstart\caption@ORI@capstart
1799     \let\caption@ORI@capstart\undefined}%
1800     \@ifundefined{Hy@float@makebox}{}{%
1801     \caption@Debug{%
1802     Restore hyperref redefinition of \string\float@makebox\@gobble}%
1803     \let\float@makebox\Hy@float@makebox
1804     \let\Hy@float@makebox\undefined}%
1805     }%

```

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

```

1806     \newcommand*\caption@star[2]{#1#2}%
1807     }{%
1808     \caption@ifincompatible{%
1809     \caption@WarningNoLine{%
1810     Forced redefinition of \noexpand\caption since the\MessageBreak
1811     unsupported(!) package option 'compatibility=false'\MessageBreak
1812     was given. Please fasten your seat belts, as we may\MessageBreak
1813     experience turbulence}%
1814     }{}%

```

**\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.**

```

1815 \def\caption{%
1816 \caption@iftype
1817 {\caption@checkgrouplevel\@empty\caption
1818 \caption@star
1819 {\caption@refstepcounter\@capttype}%
1820 {\caption@dblarg{\@caption\@capttype}}}%
1821 {\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.

1822 \newcommand*\caption@star[2]{%
1823 \ifstar{\global\caption@startrue#2[]}{#1#2}}%

\@caption As above, our version has been adapted to the packages we support. Addition-
ally our code is nested by \caption@beginex & \caption@end instead of
\begingroup & \endgroup. Furthermore we use \caption@boxrestore in-
stead of \@parboxrestore so this code also works correctly inside list-based environ-
ments like wide & addmargin. (This, and the fact that we use \linewidth instead
of \hsize inside \@makecaption, solves LATEX PR latex/2472.)

1824 \long\def\@caption#1[#2]#3{%
1825 \ifcaption@star \else
1826 \caption@makecurrent{#1}{#2}%
1827 \caption@ifhycap{\caption@@start}{}%
1828 \fi

1829 \par
1830 \caption@beginex{#1}{#2}{#3}%
1831 \caption@setfloatcapt{%
1832 \caption@boxrestore
1833 \caption@normalsize
1834 \ifcaption@star
1835 \let\caption@@anchor\@firstofone
1836 \fi
1837 \@makecaption{\csname fnum@#1\endcsname}%
1838 {\ignorespaces\caption@@anchor{#3}}\par}%
1839 \caption@end}%

1840 }%

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

1841 \let\caption@ORI@xfloat\@xfloat
1842 \def\@xfloat#1[#2]{%
1843 \caption@ORI@xfloat{#1}[#2]%
1844 \caption@settype{#1}}%

1845 }

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

1846 \long\def\@makecaption#1#2{%

```

```

1847 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
1848 \caption@@make{#1}{#2}%
1849 \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

```

1850 \@ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}

```

### 2.18.2 The beamer class

```

1851 \@ifclassloaded{beamer}{%
1852 \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.

```

1853 \captionsetup{listof=false}
1854 \undefine@key{caption}{listof}
1855 \undefine@key{caption}{listofformat}
1856 }{}

```

### 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.

```

1857 \@ifundefined{scr@caption}{}{%
1858 \caption@AtBeginDocument{\let\scr@caption\caption}}

```

### 2.18.4 The frenchb Babel option

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

```

1859 \@ifundefined{caption@frenchb}{}{\caption@frenchb}

```

### 2.18.5 The frenchle/pro package

```

1860 \caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
1861 \caption@Info{frenchle/pro package is loaded}%
1862 \let\captionfont@ORI\captionfont
1863 \let\captionlabelfont@ORI\captionlabelfont
1864 \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`.

```

1865 \@ifundefined{GOfrench}%
1866 { \let\@tempa\@firstofone}%
1867 { \def\@tempa{\g@addto@macro\GOfrench}}%
1868 \@tempa{%
1869 \let\captionfont\captionfont@ORI
1870 \let\captionfont@ORI\undefined
1871 \let\captionlabelfont\captionlabelfont@ORI

```

```

1872 \let\captionlabelfont@ORI\@undefined
1873 \let\@makecaption\@makecaption@ORI
1874 \let\@makecaption@ORI\@undefined

\@cnORI We update the definition of \@cnORI so it actually reflects our definition of \caption.
1875 \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.
1876 \let\caption@tcORI\@tablescaption
1877 \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.
1878 \let\@eatDP\@undefined
1879 \let\@tempa\@empty
1880 \ifx\@ffrench\@fnum@figure
1881 \l@addto@macro\@tempa{\let\fnum@figure\@ffrench}%
1882 \fi
1883 \ifx\@tfrench\@fnum@table
1884 \l@addto@macro\@tempa{\let\fnum@table\@tfrench}%
1885 \fi
1886 \def\@ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
1887 \def\@tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
1888 \@tempa

1889 }%
1890 } }

```

## 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.
1891 \newcommand\caption@ifpackageloaded[1]{%
1892 \ttestopt{\caption@ifpackageloaded{#1}}{}}
1893 \onlypreamble\caption@ifpackageloaded

1894 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
1895 \@ifpackageloaded{#1}\@firstofone{%
1896 \caption@Debug{#1 package is not loaded (yet)\@gobble}%
1897 \caption@AtBeginDocument}{%
1898 \caption@ifpackageloaded{#1}[#2]{#3}{#4}}
1899 \onlypreamble\caption@ifpackageloaded

1900 \newcommand\caption@ifpackageloaded[1]{%
1901 \ttestopt{\caption@ifpackageloaded{#1}}{}}
1902 \onlypreamble\caption@ifpackageloaded

```

```

1903 \long\def\caption@@ifpackageloaded#1[#2]{%
1904   \@ifpackageloaded{#1}{%
1905     \caption@Info{#1 package is loaded}%
1906     \@ifpackagelater{#1}{#2}\@firstoftwo{%
1907       \caption@Error{%
1908         For a successful cooperation we need at least version\MessageBreak
1909         `#2' of package #1,\MessageBreak
1910         but only version\MessageBreak
1911         '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
1912         is available}%
1913       \@secondoftwo}%
1914     }{\@secondoftwo}}
1915 \@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.

```

1916 \newcommand*\caption@clearmargin{%
1917   \setcaptionmargin\z@
1918   \let\caption@minmargin\@undefined}

1919 \caption@setbool{needfreeze}{0}
1920 \caption@AtBeginDocument*{%
1921   \caption@ifneedfreeze{%

```

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

```

1922 \newcommand*\caption@freeze{%
1923   \caption@teststar\caption@@freeze\@gobble\@firstofone}%

1924 \newcommand*\caption@@freeze[1]{%
1925   \global\let\caption@SCcontinued\relax
1926   \global\let\caption@SCsetup\@undefined
1927   \global\let\caption@SClentry\@undefined
1928   \global\let\caption@SCtext\@undefined
1929   \global\let\caption@SClabel\@undefined

1930   \let\caption@ORI@ContinuedFloat\ContinuedFloat
1931   \def\ContinuedFloat{%
1932     \let\caption@ORI@setcounter\setcounter
1933     \let\caption@ORI@addtocounter\addtocounter
1934     \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
1935     \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
1936     \caption@ORI@ContinuedFloat
1937     \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
1938     \let\setcounter\caption@ORI@setcounter
1939     \let\addtocounter\caption@ORI@addtocounter}%
1940   \def\captionsetup##1{%
1941     \caption@g@addto@list\caption@SCsetup{##1}%
1942     \caption@setup{##1}}%
1943   \let\caption@ORI\caption
1944   \def\caption{%
1945     \def\caption{\caption@Error{%
1946       Only one \noexpand\caption can be placed in this environment}}%
1947     \let\captionsetup\caption@setup
1948     \let\caption@@refstepcounter\caption@l@stepcounter

```



```

1949     \caption@ORI}%
1950     \long\def\@caption##1[##2]##3{%
1951         \gdef\caption@SClentry{##2}\gdef\caption@SCtext{##3}%
1952         \ignorespaces}%
1953     #1{% is \@gobble in star form, and \@firstofone otherwise
1954         \def\label##1{\gdef\caption@SClabel{##1}\ignorespaces}}%
1955 }%

\caption@defrost \caption@defrost
1956 \newcommand*\caption@defrost{%
1957     \ifx\caption@ORI@ContinuedFloat\@undefined
1958         \caption@SCcontinued
1959         \ifx\caption@SCsetup\@undefined \else
1960             \expandafter\captionsetup\expandafter{\caption@SCsetup}%
1961         \fi
1962         \ifx\caption@SCtext\@undefined \else
1963             \expandafter\expandafter\expandafter\caption
1964             \expandafter\expandafter\expandafter[%
1965             \expandafter\expandafter\expandafter{%
1966             \expandafter\caption@SClentry\expandafter}\expandafter]%
1967             \expandafter{\caption@SCtext}%
1968         \fi
1969         \ifx\caption@SClabel\@undefined \else
1970             \expandafter\label\expandafter{\caption@SClabel}%
1971         \fi
1972     \else
1973         \caption@Error{Internal Error:\MessageBreak
1974             \noexpand\caption@defrost in same group as \string\caption@freeze}%
1975     \fi}%

1976 }{}%
1977 \caption@undefbool{needfreeze}}

```

### 2.19.1 The float package

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

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

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

```

1978 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
1979     \@ifpackageloaded{floatrow}{%
1980         \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
1981     }{}%

```

`\@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>]`.

```
1982 \let\caption@ORI@float@setevery\@float@setevery
1983 \def\@float@setevery#1{%
1984   \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.

```
1985   \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.

```
1986   \renewcommand\caption@setfloatcapt{%
1987     \let\@makecaption\caption@make
1988     \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.

```
1989   \float@getstyle\float@style{#1}%
1990   \caption@setstyle*\float@style
1991   \caption@setoptions\float@style
1992   }{}%
1993   \caption@freezeHref % will be defrosted in \float@makebox
1994   \caption@ORI@float@setevery{#1}}%
```

`\caption@settypehook` L<sup>A</sup>T<sub>E</sub>X and almost every other packages use `\<type>name` to provide a macro for the type resp. environment name – for example the command `\figurename` will usually contain the name of the floating environment figure:

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

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

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

```
1995 \g@addto@macro\caption@settypehook{%
1996   \expandafter\ifx\csname #1name\endcsname\relax
1997     \expandafter\let\csname #1name\expandafter\endcsname
1998                       \csname fname@#1\endcsname
1999   \fi}%
```

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

```
2000 \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
2001 \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`).

```
2002 \providecommand*\float@ifstyle[1]{%
2003   \expandafter\ifx\csname fst@#1\endcsname\relax
2004     \expandafter\@secondoftwo
2005   \else
2006     \expandafter\@firstoftwo
2007   \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.

```
2008 \providecommand*\float@getstyle[2]{%
2009   \edef#1{%
2010     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2011     \expandafter\expandafter\expandafter\noexpand
2012     \csname fst@#2\endcsname}%
2013   \edef#1{#1}%
2014   \caption@Debug{floatstyle{#2} = '#1'}%}
```

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

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

```
2015 \providecommand*\float@setstyle[2]{%
2016   \expandafter\edef\csname fst@#1\endcsname{%
2017     \expandafter\noexpand\csname fs@#2\endcsname}%}
```

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

```
2018 \providecommand*\float@dostyle[1]{%
2019   \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.)

```

2020 \providecommand*\float@ifcaption[1]{%
2021   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2022   \expandafter\@firstoftwo
2023   \else
2024     \expandafter\@secondoftwo
2025   \fi}%

2026 }}{%
2027   \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2028   \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2029 % \clearcaptionsetup{boxed}% used by the floatrow package?
2030 }

```

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

```
2031 \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’.

```

2032 \caption@ifbool{ruled}{%
2033   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
2034 }{%
2035   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2036 }
2037 \caption@undefbool{ruled}

```

## 2.19.2 The floatflt package

```
2038 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%
```

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

```

2039 \let\caption@ORI@floatingfigure\floatingfigure
2040 \def\floatingfigure{%
2041   \caption@floatflt{figure}%
2042   \caption@ORI@floatingfigure}%

```

\floatingtable Same with \floatingtable...

```

2043 \let\caption@ORI@floatingtable\floatingtable
2044 \def\floatingtable{%
2045   \caption@floatflt{table}%
2046 %   \caption@setautoposition b%
2047   \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.

```

2048 \newcommand*\caption@floatflt[1]{%
2049   \caption@settype{#1}%
2050   \caption@clearmargin
2051   \caption@setoptions{floating#1}%
2052   \let\caption@boxrestore\@parboxrestore}%

2053 }{}

```

### 2.19.3 The fltpage package

```

2054 \caption@ifPackageLoaded{fltpage}[1998/10/29 v.0.3]{%
2055   \caption@setbool{needfreeze}{1}%

```

`\FP@helpNote` Original code:

```

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

2056 \renewcommand\FP@helpNote[2]{%
2057   \begingroup % save \caption@thepage
2058     \caption@pageref{#2}%
2059     \typeout{FP#1 is inserted on page \caption@thepage!}%
2060   \endgroup}%

```

`\FP@floatBegin` Original code:

```

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

2061 \renewcommand*\FP@floatBegin[1]{%
2062   \def\@captype{#1}%
2063   \let\FP@LabelText\@empty
2064   \begin{lrbox}{\FP@floatCorpusBOX}%
2065     \caption@ifFPrefcap
2066       {\caption@freeze\relax}%
2067       {\def\label##1{\gdef\FP@LabelText{##1}\ignorespaces}%
2068       \caption@freeze*}}%

```

\FP@floatEnd **Original code:**

```

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

2069 \renewcommand*\FP@floatEnd{%
2070   \end{lrbox}%
2071   \stepcounter{FP@\@captive C}%
2072   \caption@label\FP@positionLabel
2073   \FP@helpnote{\@captive}\FP@positionLabel
2074   \let\FP@ContinuedFloat\relax
2075   \FP@float
2076     {\FP@positionLabel}% location label test
2077     {\begin{\@captive}[p!]}%
2078       \FP@ContinuedFloat
2079       \usebox\FP@floatCorpusBOX
2080       \caption@ifFPlistcap
2081         {\caption@refstepcounter{\@captive}
2082           \expandafter\caption@makecurrent\expandafter{\@captive}
2083             \expandafter{\caption@Sclentry}}}%
2084       {\expandafter\captionlistentry\expandafter{\caption@Sclentry}}}%
2085       \caption@@anchor\relax
2086       \ifx\FP@LabelText\@empty \else
2087         \expandafter\label\expandafter{\FP@LabelText}%
2088       \fi
2089       \end{\@captive}%

```

```

2090     {\let\FP@ContinuedFloat\ContinuedFloat
2091     \caption@ifcompatibility
2092     {\let\caption@ContinuedFloat\@gobble}%
2093     {\caption@clearsetup{ContinuedFloat}%
2094     \caption@clearsetup{continued\@capttype}}}%
2095     {\begin\@capttype[b!]%
2096     \FP@ContinuedFloat
2097     \let\FP@savedSetfnumCommand\caption@setfnum
2098     \def\caption@setfnum##1{%
2099     \FP@savedSetfnumCommand{##1}%
2100     \ifx\FP@guide\@empty \else
2101     \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ \FP@guide}}%
2102     \fi}%
2103     \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2104     \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2105     \caption@setoptions{FP\@capttype}%
2106     \FP@separatorCaption
2107     \caption@ifFPlistcap{}{\caption@set@bool\caption@iflof0}%
2108     \caption@defrost
2109     \end\@capttype}%
2110 }%

2111 \ForFloatingEnvironments{%
2112   \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
2113 }{%
2114   \let\caption@ifFPlistcap\@undefined
2115   \let\caption@ifFPrefcap\@undefined
2116 }

```

#### 2.19.4 The hyperref package

```

2117 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2118   \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2119     \caption@WarningNoLine{%
2120       Hyperref support is turned off\MessageBreak
2121       because hyperref has stopped early}%
2122   }{%
2123     \g@addto@macro\caption@prepareslc{\measuring@true}%

```

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

```

2124   \renewcommand*\caption@@refstepcounter{\H@refstepcounter}%

```

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

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

```

2125   \renewcommand*\caption@makecurrent[2]{%
2126     \caption@makecurrentHref{#1}%
2127     \caption@Debug{hyperref current=\@currentHref}%
2128     \def\@currentlabelname{#2}}%
2129   \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%

```

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

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

```

2130 \renewcommand\caption@@anchor[1]{%
2131 \caption@Debug{hyperref anchor: \@currentHref}%
2132 % If we cannot have nesting, the anchor is empty.
2133 \ifHy@nesting
2134 \hyper@@anchor{\@currentHref}{#1}%
2135 \else
2136 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2137 \fi}%
2138 \g@addto@macro\caption@prepareslc{\let\caption@@anchor\@firstofone}%

```

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

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

will be used instead of

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

(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 \alpha\c@ContinuedFloat}}}%

```

### The `hypcap` option

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

```

2143 \newif\if@capstart

```

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

```

2144 \def\caption@start{\caption@ifhypcap{%

```

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

```

2145 \caption@makestart\@captype
2146 \caption@startanchor\@currentHref

```

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

```

2147 \global\@capstarttrue
2148 \let\hc@currentHref\@currentHref
2149 \def\caption@makecurrentHref##1{%
2150 \global\@capstartfalse
2151 \global\let\@currentHref\hc@currentHref}%

```

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

```

2152 \let\caption@@anchor\@firstofone

```



2153        }{} }%

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

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

-or-

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

```
2154        \newcommand*\caption@makestart[1]{%
2155            \beginngroup
2156            \Hy@hypertextnamesfalse
2157 %        \gdef\@currentHlabel{}%
2158        \hyper@makecurrent{#1.caption}%
2159        \endngroup
2160        \caption@Debug{hycap start=\@currentHref}}%
```

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

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

```
2161        \newcommand*\caption@startanchor[1]{%
2162            \caption@Debug{hycap anchor: #1}%
2163            \ifvmode\beginngroup
2164            \@tempdima\prevdepth
2165            \nointerlineskip
2166            \vspace*{-\caption@hycapsspace}%
2167            \caption@anchor{#1}%
2168            \vspace*{\caption@hycapsspace}%
2169            \prevdepth\@tempdima
2170            \endngroup\else
2171            \caption@Warning{Internal Warning: Vertical TeX mode expected}%
2172            \caption@anchor{#1}%
2173            \fi}%%
```

`\caption@anchor`        `\caption@anchor{⟨Href⟩}` sets a `hyperref` anchor.

```
2174        \newcommand*\caption@anchor[1]{%
2175            \ifmeasuring@ \else
2176            \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2177            \fi}%%
```

*Note:* Since `\Hy@raisedlink` change `\@tempdima` we surrounded it by `\ifvmode`, suppressing “LaTeX Warning: Float too large for page by 1.0pt” in sideways floats. (This will not be necessary with future versions of `hyperref`.)

```
2178 \ifx\HyperRaiseLinkLength\@tempdima
2179 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2180 \else
2181 \let\caption@raisedlink\Hy@raisedlink
2182 \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.

```
2183 \def\caption@@start{%
2184 \ifundefined{hc@currentHref}{%
2185 \caption@Warning{%
2186 The option ‘hycap=true’ will be ignored for this\MessageBreak
2187 particular \string\caption}}}%
```

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

```
2188 \def\caption@freezeHref{%
2189 \let\caption@ORI@start\caption@start
2190 \def\caption@start{\let\caption@start\caption@ORI@start}%
2191 % \let\caption@ORI@@start\caption@@start
2192 % \l@addto@macro\subcaption@settypehook{%
2193 % \let\caption@@start\caption@ORI@@start}%
2194 \global\let\caption@currentHref\@undefined
2195 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2196 \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2197 \renewcommand*\caption@setfloatcapt{%
2198 \ifx\caption@currentHref\@undefined \else
2199 \let\caption@@anchor\@firstofone
2200 \fi
2201 \caption@ORI@setfloatcapt}}%
```

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

```
2202 \def\caption@defrostHref{%
2203 \ifx\caption@currentHref\@undefined \else
2204 \caption@startanchor\caption@currentHref
2205 \global\let\caption@currentHref\@undefined
2206 \fi}%
```

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

```
2207 \ifundefined{HyOrg@float@makebox}{}{%
2208 \caption@Debug{%
2209 Redefining \noexpand\float@makebox (again)\@gobble}%
2210 \let\Hy@float@makebox\float@makebox
2211 \renewcommand\float@makebox[1]{%
2212 \HyOrg@float@makebox{#1}\relax \caption@defrostHref}}%
2213 }%
```

```
2214 } {} }
```

### 2.19.5 The hypcap package

```
2215 \caption@ifpackageloaded{hypcap}{% v1.0
2216 \ifx\caption@start\relax \else % hyperref hasn't stopped early
```

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

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

```
2217 \let\caption@ORI@caption@start\caption@start
2218 \let\caption@start\caption@start
2219 \let\caption@start\relax
2220 \let\caption@@start\relax
```

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

```
2221 \caption@@setbool\caption@ifhypcap1%
2222 \renewcommand*\caption@hypcapSPACE{\hypcapSPACE}%
```

`\float@makebox` Redo the redefinition of `\float@makebox`, if necessary.

```
2223 \ifundefined{Hy@float@makebox}{}{%
2224 \caption@Debug{%
2225 Restore hyperref redefinition of \string\float@makebox\@gobble}%
2226 \let\float@makebox\Hy@float@makebox
2227 \let\Hy@float@makebox\@undefined}%
2228 \fi}{} }
```

### 2.19.6 The listings package

```
2229 \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`.

```
2230 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2231 \def\lst@MakeCaption#1{%
2232 \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.

```
2233 \caption@setautoposition{#1}%
2234 \caption@begin{lstlisting}%
2235 \caption@ORI@lst@MakeCaption{#1}%
2236 \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.

```
2237 \providecommand*\ext@lstlisting{lol}%
2238 } {} }
```

## 2.19.7 The longtable package

```

\LTcapttype \LTcapttype is preset to table.
2239 \providecommand*\LTcapttype{table}

2240 \caption@ifPackageLoaded{longtable}[1995/05/24 v3.14]{%
2241   \RequirePackage{ltcaption}[2007/09/01]%
2242   \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.
2243 \caption@AtBeginDocument{%
2244   \let\caption@ORI@LT@array\LT@array
2245   \renewcommand*\LT@array{%
\captionsetup for longtable:
2246     \global\let\caption@opt@@longtable\@undefined
2247     \def\captionsetup##1{\noalign{\LT@captionsetup{##1}}}%
2248     \def\LT@captionsetup##1{%
2249       \@captionsetup\@gobble[@longtable]{##1}%
2250       \global\let\caption@opt@@longtable\caption@opt@@longtable}%
\captionabove & \captionbelow for longtable: (KOMA-Script document class)
2251     \def\@captionabovetrue{\LT@captionsetup{position=t}}%
2252     \def\@captionabovefalse{\LT@captionsetup{position=b}}%
\captionlistentry for longtable:
2253     \def\captionlistentry##1{\noalign{%
2254       \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
2255 % \caption@ifhyppcap{%
2256 %   \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2257 %   \def\hyper@makecurrent##1{%
2258 %     \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2259 %     \caption@makestart{##1}%
2260 %     \let\Hy@LT@currentHlabel\@currentHlabel
2261 %     \let\Hy@LT@currentHref\@currentHref
2262 %     \def\hyper@makecurrent####1{%
2263 %       \let\@currentHlabel\Hy@LT@currentHlabel
2264 %       \let\@currentHref\Hy@LT@currentHref}}%
2265 %   \let\caption@ORI@ContinuedFloat\ContinuedFloat
2266 %   \def\ContinuedFloat{\noalign{%
2267 %     \gdef\caption@setContinuedFloat{%
2268 %       \let\caption@resetContinuedFloat\@gobble}%
2269 %     \def\caption@setoptions####1{%
2270 %       \g@addto@macro\caption@setContinuedFloat{%
2271 %         \caption@setoptions{####1}}}%
2272 %     \let\@capttype\LTcapttype
2273 %     \caption@ORI@ContinuedFloat}}%

```

```

2274 %      }{%
2275 %          \def\ContinuedFloat{\noalign{%
2276 %              \caption@Error{%
2277 %                  \noexpand\ContinuedFloat inside longtables\MessageBreak
2278 %                  is only available with 'hypcap=true'}}}%
2279 %      }%
2280 %      \global\let\caption@setContinuedFloat\@empty
2281 %      \def\ContinuedFloat{\noalign{%
2282 %          \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2283 %      \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}`:

```

2284 \def\LT@c@ption#1[#2]#3{%
2285   \LT@makecaption#1{\csname fnun@\LTcapttype\endcsname}{#3}%
2286   \def\@currentlabelname{#2}%
2287   \caption@addcontentsline\LTcapttype{#2}}%

```

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

The original definition:

```

\def\LT@makecaption#1#2#3{%
  \LT@mc@l\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:

```

2288 \renewcommand\LT@makecaption[3]{%
2289   \caption@LT@make{%

```

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

```

2290   \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`.)

```

2291 \caption@settype*\LTcaptiontype
2292 \ifdim\LTcapwidth=4in \else
2293 \setcaptionwidth\LTcapwidth
2294 \fi
2295 \caption@setoptions{longtable}%
2296 % \caption@setContinuedFloat
2297 \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!

```

2298 \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
2299 \caption@resetContinuedFloat\LTcaptiontype
2300 \caption@begin\LTcaptiontype
2301 \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...)

```

2302 \vskip-\ht\strutbox

```

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

```

2303 \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2304 \caption@@make{#2}{#3}\endgraf
2305 \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2306 \caption@end}}%
2307 {}{}

```

## 2.19.8 The picinpar package

```

2308 \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.

```

2309 \long\def\figwindow[#1,#2,#3,#4] {%
2310 \caption@window{figure}%
2311 \caption@setoptions{figwindow}%
2312 \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
2313 \long\def\tabwindow[#1,#2,#3,#4] {%
2314 \caption@window{table}%
2315 \caption@setoptions{tabwindow}%
2316 \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.

```

2317 \newcommand*\caption@window[1]{%
2318   \let\caption@boxrestore\@parboxrestore
2319   \let\@makecaption\caption@\@make
2320   \caption@setautoposition b%
2321   \caption@settype{#1}%
2322   \caption@clearmargin}%

```

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

```

2323 \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.
2324   \ifdim\picwd=\z@
2325     \let\caption@makecurrent\@gobbletwo
2326     \let\caption@\@start\relax
2327     \caption@prepareslc
2328   \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.

```

2329   \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2330   \if\@tempa*%
2331     \let\@tempa\@firstofone
2332   \else\if\@tempa[%
2333     \let\@tempa\@firstofone
2334   \else
2335     \let\@tempa\@empty
2336   \fi\fi
2337   \expandafter\caption\@tempa{#1}}%
2338 {}{}

```

## 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 `\@capytype`, 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.

```

2339 \newcommand*\piccaptiontype[1]{\def\@piccapytype{#1}}

```

```

2340 \caption@ifPackageLoaded{picins}{%

```

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

```

2341   \@ifundefined{@piccapytype}{%
2342     \caption@iftype{%
2343       \let\@piccapytype\@capytype
2344     }{%
2345       \def\@piccapytype{figure}%
2346     }%
2347   }{}%
2348   \let\@capytype\@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.

```
2349 \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.

```
2350 \let\caption@ORI@make@piccaption\make@piccaption  
2351 \def\make@piccaption{%  
2352   \let\caption@ORI\caption  
  
2353   \long\def\caption[##1]##2{%  
2354     \caption@freezeHref % will be defrosted in \ivparpic  
2355     \caption@settype\@piccapttype  
2356     \ifnum\c@piccaptionpos>2\relax  
2357       \caption@clearmargin  
2358     \else  
2359       \captionwidth\z@ % do not use "width=" setting  
2360     \fi  
2361     \caption@setoptions{parpic}%  
2362     \let\caption@boxrestore\@parboxrestore  
2363     \caption@setautoposition b%  
  
2364     \expandafter\expandafter\expandafter\caption@ORI  
2365       \expandafter\expandafter\expandafter[%  
2366       \expandafter\expandafter\expandafter{%  
2367       \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
2368   \caption@ORI@make@piccaption
2369   \let\caption\caption@ORI}%

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

2370   \let\caption@ORI@ivparpic\ivparpic
2371   \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
2372     \let\caption@ORI@noindent\noindent
2373     \def\noindent{%
2374       \caption@defrostHref
2375       \let\noindent\caption@ORI@noindent
2376       \noindent}%
2377     \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2378     \let\noindent\caption@ORI@noindent}%

2379 }{%
2380   \let\piccaptiontype\@undefined
2381 }

```

### 2.19.10 The rotating package

```

2382 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%

\rotcaption Make \rotcaption* work.

2383   \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
2384 % \let\@rotcaption\@undefined

\rotcaptionof Make \rotcaptionof(*) work.

2385   \def\rotcaptionof{%
2386     \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.)

```

2387   \long\def\@makerotcaption#1#2{%

```

```

2388 \ifdim\captionwidth=\z@
2389 \setcaptionwidth{.8\textheight}%
2390 \caption@slc{#1}{#2}{.8\vsizel}{%
2391 \let\caption@makerot\caption@@make
2392 \caption@clearmargin
2393 % \long\def\captionbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2394 % (not needed because \rotatebox uses an \hbox anyway)
2395 \let\captionbox\@secondoftwo}%
2396 \caption@set@bool\caption@ifslc0% been there, done that
2397 \fi

2398 \rotatebox{90}{\caption@makerot{#1}{#2}}%
2399 \nobreak\hspace{12pt}}%

2400 \newcommand\caption@makerot[2]{%
2401 \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%

2402 \ForFloatingEnvironments{%
2403 \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
2404 \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
2405 {}{}

```

## 2.19.11 The sidecap package

```

2406 \caption@ifPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
2407 \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.)

```

2408 \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`.

```

2409 \let\caption@ORI\SC@zfloat\SC@zfloat
2410 \def\SC@zfloat#1#2#3[#4]{%

```

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

```

2411 \let\caption@ORI\caption
2412 \caption@ORI\SC@zfloat{#1}{#2}{#3}[#4]%
2413 \let\caption\caption@ORI

```

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

```

2414 \caption@settype*{#2}%
2415 \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.

```

2416 \let\caption@ORI\endSC@FLOAT\endSC@FLOAT
2417 \def\endSC@FLOAT{%

```

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

```

2418 \let\caption@ORI@settype\caption@settype
2419 \def\caption@settype##1{% will be done in \xfloat
2420 \caption@ORI@settype*{##1}% do not change \@currentlabel
2421 \caption@setSC@justify
2422 %%% \caption@setoptions{SCfloat}%
2423 \caption@setoptions{SC\@capttype}%
2424 \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 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

```

2425 \def\caption@setSC@justify{%
2426 \caption@clearmargin
2427 \@ifundefined{SC@justify}{}{%
2428 \ifx\SC@justify\@empty \else
2429 \let\caption@hj\SC@justify
2430 \let\SC@justify\@empty
2431 \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!

```

2432 \let\caption\SC@orig@caption
2433 \def\SC@orig@caption[##1]##2{\caption@defrost}%

```

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

```

2434 \caption@setSC@justify % for compatibility mode
2435 \caption@ORI@endSC@FLOAT}%

2436 \ForFloatingEnvironments{%
2437 \newenvironment{SC#1}%
2438 {\SC@float[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@float}%
2439 \newenvironment{SC#1*}%
2440 {\SC@dblfloat[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@dblfloat}}%
2441 {}

```

## 2.19.12 The subfigure package

```

2442 \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.

```

2443 \def\sf@ifpositiontop{%
2444 \ifx\@capttype\@undefined
2445 \expandafter\@gobbletwo
2446 \else\ifx\@capttype\relax

```

```

2447     \expandafter\expandafter\expandafter\@gobbletwo
2448   \else
2449     \expandafter\expandafter\expandafter\sfi@if@position@top
2450   \fi\fi}

2451 \def\sfi@if@position@top{%
2452   \ifundefined{if\@captype topcap}%
2453     {\@gobbletwo}%
2454     {\@nameuse{if\@captype topcap}%
2455       \expandafter\@firstoftwo
2456     \else
2457       \expandafter\@secondoftwo
2458     \fi}}

2459 }{}

```

### 2.19.13 The supertabular package

```

2460 \caption@ifPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.
2461   \let\caption@ORI@tablecaption\tablecaption
2462   \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}

2463 \long\def\ST@caption#1[#2]#3{\par%
2464   \caption@settype*{#1}%
2465   \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.**

```

2466   \def\caption@fixposition{%
2467     \caption@setposition{\if@topcaption t\else b\fi}}%

2468   \caption@beginex{#1}{#2}{#3}%
2469   \caption@boxrestore
2470   \caption@normalsize
2471   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2472   \caption@end}%

2473 }{%
2474   \let\caption@setSTposition\@undefined
2475 }

```

## 2.19.14 The threeparttable package

2476 \caption@ifPackageLoaded{threeparttable}[2003/06/13 v3.0]{%  
 \threeparttable Unfortunately \@capttype is not set when \TPT@common will be used, so we have to  
 redefine \threeparttable and \measuredfigure instead.

```
2477 \let\caption@ORI@threeparttable\threeparttable
2478 \renewcommand*\threeparttable{%
2479   \caption@settype{table}%
2480   \caption@setposition a% ?
2481   \caption@clearmargin
2482   \caption@setoptions{threeparttable}%
2483   \caption@ORI@threeparttable}%
```

\measuredfigure Same here...

```
2484 \let\caption@ORI@measuredfigure\measuredfigure
2485 \renewcommand*\measuredfigure{%
2486   \caption@settype{figure}%
2487   \caption@setposition a% ?
2488   \caption@clearmargin
2489   \caption@setoptions{measuredfigure}%
2490   \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}

2491 \def\TPT@caption#1[#2]#3{%
2492   \gdef\TPT@docapt{%
2493     \global\let\TPT@docapt\@undefined
2494     \caption@setautoposition\caption@TPT@position
2495     \TPT@LA@caption{#1}[#2][#3]}%
2496   \ifx\TPT@hsize\@empty
2497     \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
2498     \gdef\caption@TPT@position{t}%
2499     \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2500   \else
2501     \def\caption@TPT@position{b}%
2502     \TPT@docapt
2503   \fi
2504   \ignorespaces}%
2505   %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2506   \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima}%

2507 }{}
```

## 2.19.15 The wrapfig package

2508 \caption@ifPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)  
 \float@ifstyle \float@ifstyle{<type>}{<if-clause>}{<else-clause>}}

(see float package support for details)

```
2509 \providecommand*\float@ifstyle[1]{%
2510   \expandafter\ifx\csname fst@#1\endcsname\relax
2511   \expandafter\@secondoftwo
2512   \else
2513   \expandafter\@firstoftwo
2514   \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.

```
2515 \newcommand*\caption@restylewrapfloat[1]{%
2516   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2517   \csname wrap#1\endcsname
2518   \caption@Error{%
2519     For a successful cooperation of the 'wrapfig' package\MessageBreak
2520     with the 'float' package you should load the 'wrapfig'\MessageBreak
2521     package *after* the 'float' package}%
2522   \else
2523     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2524     \csname wrap#1\endcsname
2525     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
2526     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2527     \csname wrap#1\endcsname
2528   \fi}%
```

`\caption@wrapfloat`

```
2529 \newcommand*\caption@wrapfloat[1]{%
2530   \caption@settype*{#1}%
2531   \float@ifstyle{#1}{%
2532     \ifx\WF@floatstyhook\@undefined
2533     \caption@Error{%
2534       For a successful cooperation of the 'wrapfig' package\MessageBreak
2535       with the 'float' package you should use at least\MessageBreak
2536       'wrapfig' version 3.6}%
2537     \else
2538     \float@dostyle{#1}%
2539     \fi}{}%
2540   \caption@clearmargin
2541   %%% \caption@setoptions{wrapfloat}%
2542   \caption@setoptions{wrap#1}%
2543   \@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`.)

```

2544 \caption@restylewrapfloat{figure}%
2545 \caption@restylewrapfloat{table}%

2546 \ForFloatingEnvironments{%
2547   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
2548   \caption@restylewrapfloat{#1}}%
2549 \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.
2550 \ifundefined{float@restyle}{}{%
2551   \toks@=\expandafter{\float@restyle{#1}% (env may or may not be defined)
2552   \caption@restylewrapfloat{#1}}%
2553   \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2554   \@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.
2555 \let\caption@ORI@wrapfloat\wrapfloat
2556 \def\wrapfloat#1{%
2557   \float@ifstyle{#1}{%
2558     \caption@Error{%
2559       For a successful cooperation of the 'wrapfig' package\MessageBreak
2560       with the 'float' package you should load the 'wrapfig'\MessageBreak
2561       package *right after* the 'float' package}}{%
2562     \caption@ORI@wrapfloat{#1}}%

2563 \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\bgrou \setlength\hsize{#2}%
\ifdim\hsize>\z@ \@parboxrestore \else
\setbox\z@\hbox\bgrou \let\wf@caption\caption \let\caption\wf@caption
\ignorespaces \fi}

Our code:
2564 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2565   \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2566   \global\setbox\WF@box\top\bgrou \setlength\hsize{#2}%
2567   \caption@start
2568   \ifdim\hsize>\z@ \@parboxrestore \else
2569   \setbox\z@\hbox\bgrou \let\wf@caption\caption \let\caption\wf@caption
2570   \ignorespaces \fi}%
2571 {}

```

## References

- [1] Frank Mittelbach and Michel Goossens:  
*The L<sup>A</sup>T<sub>E</sub>X Companion (2nd. Ed.)*,  
Addison-Wesley, 2004.
- [2] Till Tantau:  
*User Guide to the Beamer Class, Version 3.07*,  
March 11, 2007
- [3] Markus Kohm & Jens-Uwe-Morawski:  
*KOMA-Script – a versatile L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> bundle*,  
2007-01-09
- [4] Victor Eijkhout:  
*An introduction to the Dutch L<sup>A</sup>T<sub>E</sub>X document classes*,  
3 September 1989
- [5] Anselm Lingnau:  
*An Improved Environment for Floats*,  
2001/11/08
- [6] Mats Dahlgren:  
*Welcome to the floatflt package*,  
1998/06/05
- [7] Olga Lapko:  
*The floatrow package documentation*,  
2007/08/24
- [8] Sebastian Gross:  
*Welcome to the beta test of fltpage package!*,  
1998/11/13
- [9] Sebastian Rahtz & Heiko Oberdiek:  
*Hypertext marks in L<sup>A</sup>T<sub>E</sub>X*,  
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 L<sup>A</sup>T<sub>E</sub>X*,  
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