

The Implementation of the caption package^{*}

Axel Sommerfeldt
caption@sommerfee.de

2007/10/17

Abstract

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

The kernel provides all the user commands and internal macros which are necessary for typesetting captions and setting parameters regarding these. While the standard L^AT_EX document classes 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^AT_EX document – it just provides functionality which can be used by L^AT_EX 2_ε packages which typesets captions, like the caption package and the subfig package.

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

caption-eng.pdf	The English documentation
caption-rus.pdf	The Russian documentation ¹
caption-deu.pdf	The German documentation

^{*}This package has version number v3.1c, last revised 2007/10/17.

¹Thanks a lot to Olga Lapko for this translation

Contents

1	Kernel	4
1.1	Identification	4
1.2	Generic helpers	4
1.3	Errors	7
1.4	Using the keyval package	8
1.5	Margin resp. width	12
1.6	Indentions	14
1.7	Styles	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 ^A T _E 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
2	Main package	37
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	66
2.19.6	The listings package	67
2.19.7	The longtable package	67
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}[2007/10/17 v3.1c caption3 kernel (AR)]
```

1.2 Generic helpers

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

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

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

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

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

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

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

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

would expand to yes.

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

```

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

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

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

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

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

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

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

\caption@setbool For setting and testing boolean options we offer these three helper macros:
\caption@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@Error{<message>}
\caption@WarningNoLine 127 \newcommand*\caption@Warning[1]{\caption@WarningNoLine{#1\on@line}}
\caption@Error 128 \newcommand*\caption@WarningNoLine[1]{\PackageWarning{caption}{#1.^J\caption@wh\
\caption@KV@err 129 \newcommand*\caption@Error[1]{\PackageError{caption}{#1}\caption@eh}
130 \let\caption@KV@err\caption@Error

```

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

```

\caption@eh 131 \newcommand*\caption@wh{%
132   See the caption package documentation for explanation.}
133 \newcommand*\caption@eh{%
134   If you do not understand this error, please take a closer look\MessageBreak
135   at the documentation of the 'caption' package, especially the\MessageBreak
136   section about errors.\MessageBreak\@ehc}

```

1.4 Using the keyval package

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

```
137 \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.

```
138 \providecommand*\undefine@key[2]{%
139   \@nameundef{KV@#1@#2}\@nameundef{KV@#1@#2@default}}
```

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

Analogous to `\@onlypreamble` from L^AT_EX 2_ε.

```
140 \providecommand*\@onlypreamble@key[2]{%
141   \expandafter\@onlypreamble\csname KV@#1@#2\endcsname}
142 \@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^AT_EX 2_ε), so the keyval package is used. The starred form makes the option available during the lifetime of the current package only, so they can be used with `\usepackage`, but *not* with `\captionsetup` later on.

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

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

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

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

```
163 \newcommand*\captionsetup{%
164   \caption@teststar\captionsetup\@gobble\@firstofone}
```



```

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

```

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

```

209 \newcommand*\showcaptionsetup[2][\@firstofone]{%
210   \GenericWarning{ }{%
211     #1 Caption Info: KV list on `#2'\MessageBreak
212     #1 Caption Data: \@ifundefined{caption@opt@#2}{%
213       -none-%
214     }{%
215       {\expandafter\expandafter\expandafter\strip@prefix
216        \expandafter\meaning\csname caption@opt@#2\endcsname}%
217     }}
218 \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.

```

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

```

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

```

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

```

260 \newcommand*\caption@ExecuteOptions[2]{%
261   \@expandtwoargs\caption@setkeys{#1}{#2}}%
262 \@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.)

```

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

```

```

284 \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@currentt}}%
285 \AtEndOfPackage{\let\@unprocessedoptions\relax}
286 \@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@`.

```

287 \newdimen\captionmargin
288 \newdimen\captionmargin@
289 \newdimen\captionwidth

290 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
291 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
292 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
293 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}

294 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
295 \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.

```

296 \newcommand*\setcaptionmargin{%
297   \caption@teststar\caption@setmargin\@gobble\@firstofone}

298 \newcommand*\caption@setmargin[2]{%
299   #1{\captionwidth\z@}%
300   \caption@@setmargin#2,#2,\@nil}

301 \def\caption@@setmargin#1,#2,#3\@nil{%
302   \setlength\captionmargin@{#2}%
303   \setlength\captionmargin{#1}%
304   \addtolength\captionmargin@{-\captionmargin}}

```

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

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

```

305 \newcommand*\setcaptionwidth{%
306   \captionmargin\z@
307   \captionmargin@\z@
308   \setlength\captionwidth}

```

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

```

309 \newcommand*\caption@count{0}

```

`\caption@newlabel` This command is a modified version of `\newlabel` from L^AT_EX2e. 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`.)

```

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

```

`\caption@thepage` This command is a modified version of `\thepage` from L^AT_EX2e. It will be used inside `\caption@ifoddpage` only.

```

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

```

`\caption@label` This command is a modified version of `\label` from L^AT_EX2e. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

312 \newcommand*\caption@label[1]{%
313   \protected@write\@auxout{\let\caption@thepage\relax}%
314     {\string\caption@newlabel{#1}{\caption@thepage}}}

```

`\caption@pageref` This command is a modified version of `\pageref` from L^AT_EX2e. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

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

```

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

```

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

```

338 \newcommand*\caption@setoptcmd[2]{%
339   \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.

340 \newdimen\caption@indent
341 \newdimen\caption@parindent
342 \newdimen\caption@hangindent

343 \DeclareCaptionOption{indent}[\leftmargini]{%
344     \setlength\caption@indent{#1}}% obsolete!
345 \DeclareCaptionOption{indentation}[\leftmargini]{%
346     \setlength\caption@indent{#1}}
347 \DeclareCaptionOption{parindent}{%
348     \setlength\caption@parindent{#1}}
349 \DeclareCaptionOption{hangindent}{%
350     \setlength\caption@hangindent{#1}}
351 \DeclareCaptionOption{parskip}{%
352     \l@addto@macro\caption@par{\setlength\parskip{#1}}}
```

There is an option clash between the KOMA-Script document classes and the caption kernel, both define the options `parindent` and `parskip` but with different meaning. Furthermore the ones defined by the caption kernel take a value as parameter but the KOMA-Script ones do not. So we need special versions of the options `parindent` and `parskip` here which determine if a value is given (and therefore should be treated as our option) or not (and therefore should be ignored by us).²

```

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

1.7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
369 \newcommand*\DeclareCaptionStyle[1]{%
370     \@testopt{\caption@declarestyle{#1}}{}
371 \onlypreamble\DeclareCaptionStyle
```

²This problem was completely solved due a change of `\caption@ProcessOptions` in the caption package v3.0j, but we still need this workaround since these options otherwise would still collide with the current version 1.3 of the subfig package (Sigh!)

```

372 \def\caption@declarestyle#1[#2]#3{%
373   \global\@namedef{caption@sls@#1}{#2}%
374   \global\@namedef{caption@sty@#1}{#3}}
375 \@onlypreamble\caption@declarestyle

376 \DeclareCaptionOption{style}{\caption@setstyle{#1}}
377 \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.

```

378 \newcommand*\caption@setstyle{%
379   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

380 \newcommand*\caption@@setstyle[2]{%
381   \@ifundefined{caption@sty@#2}{%
382     {#1{\caption@Error{Undefined style '#2'}}}%
383     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
384       \ifx\caption@setstyle@flag\@undefined
385         \let\caption@setstyle@flag\caption@setstyle
386         \let\caption@iftypewarning\@firstoftwo % = true
387         \caption@resetstyle
388         \expandafter\caption@setup\expandafter{\caption@sty}%
389         \let\caption@setstyle@flag\@undefined
390       }else
391         \expandafter\caption@setup\expandafter{\caption@sty}%
392       }fi
393     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
394     \expandafter\caption@l@addto@list\expandafter\caption@opt@SingleLine
395     \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!*

```

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

```

Currently there are two pre-defined styles, called ‘base’ & ‘default’. The first one is a perfect match to the behavior of `\@makecaption` offered by the standard L^AT_EX document classes (and was called ‘default’ in the caption package v3.0), the second one matches the document class actually used.

```

404 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
405 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
406   format=default,labelsep=default,textformat=default,%
407   justification=default,font=default,labelfont=default,textfont=default}

```

1.8 Formats

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

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

```
408 \newcommand*\DeclareCaptionFormat{%
409   \caption@teststar\caption@declareformat\@gobble\@firstofone}
410 \@onlypreamble\DeclareCaptionFormat

411 \newcommand*\caption@declareformat[2]{%
412   \@dblarg{\caption@@declareformat#1{#2}}}%
413 \@onlypreamble\caption@declareformat

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

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

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

Selecting a caption format simply means saving the code (in `\caption@fmt`) and if the code should be used in horizontal or vertical mode (`\caption@ifh`).

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

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

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

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

‘default’ usually maps to ‘plain’.

```
436 \def\caption@fmt@default{\caption@fmt@plain}
437 \def\caption@ifh@default{\caption@ifh@plain}
```

1.9 Label formats

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

438 \newcommand*\DeclareCaptionLabelFormat[2]{%
439   \global\@namedef{caption@lfmt@#1}##1##2##3{#2}}
440 \@onlypreamble\DeclareCaptionLabelFormat
```



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

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

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

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

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

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

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

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

```
446 \DeclareCaptionLabelFormat{empty}{}%
```

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

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

‘default’ usually maps to ‘simple’.

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

1.10 Label separators

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

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

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

```
450 \newcommand\DeclareCaptionLabelSeparator{%
```

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

```
452 \@onlypreamble\DeclareCaptionLabelSeparator
```

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

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

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

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

```
457 \@onlypreamble\caption@declarelabelseparator
```

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

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

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

```
461 \fi}
```

```
462 \@onlypreamble\caption@@declarelabelseparator
```

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

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

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

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

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

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

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

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

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

```
470 \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’.

```
471 \DeclareCaptionLabelSeparator{none}{}%
```

```
472 \DeclareCaptionLabelSeparator{colon}{: }
```

```

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

```

1.11 Text formats

```

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

483 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
484 \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`).

```

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

```

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

```

489 \DeclareCaptionTextFormat{simple}{#1}
490 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

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

1.12 Fonts

```

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

DeclareDefaultCaptionFont \DeclareDefaultCaptionFont{<name>}{<code>}
495 \newcommand*\DeclareDefaultCaptionFont[2]{%
496   \global\@namedef{caption#1@default}{#2}}
497 \@onlypreamble\DeclareDefaultCaptionFont

498 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
499 \DeclareDefaultCaptionFont{font}{}

500 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
501 \DeclareDefaultCaptionFont{labelfont}{}

502 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
503 \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.

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

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

These are the pre-defined font code snippets.

```
521 \DeclareCaptionFont{normalcolor}{\normalcolor}
522 \DeclareCaptionFont{color}{\color{#1}}

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

533 \DeclareCaptionFont{scriptsize}{\scriptsize}
534 \DeclareCaptionFont{footnotesize}{\footnotesize}
535 \DeclareCaptionFont{small}{\small}
536 \DeclareCaptionFont{normalsize}{\normalsize}
537 \DeclareCaptionFont{large}{\large}
538 \DeclareCaptionFont{Large}{\Large}

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

```

545 % \caption@font{normalcolor,normalfont,normalsize,singlespacing}
546 \DeclareCaptionFont{normal}{%
547   \caption@font*{%
548     \KV@caption@fnt@normalcolor\@unused
549     \KV@caption@fnt@normalfont\@unused
550     \KV@caption@fnt@normalsize\@unused
551     \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.

```

552 \providecommand\captionsize{}

553 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
554 \DeclareDefaultCaptionFont{size}{}

```

1.13 Justifications

```

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

```

```

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

563 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
564 \DeclareDefaultCaptionJustification{}

```

```

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

```

These are the pre-defined justification code snippets.

```

566 \DeclareCaptionJustification{justified}{}
567 \DeclareCaptionJustification{centering}{\centering}
568 \DeclareCaptionJustification{centerfirst}{\centerfirst}
569 \DeclareCaptionJustification{centerlast}{\centerlast}
570 \DeclareCaptionJustification{raggedleft}{\raggedleft}
571 \DeclareCaptionJustification{raggedright}{\raggedright}

```

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

```

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

```

```

580 \parfillskip\z@skip
581 \noindent\hskip\z@\@plus 2fil%
582 \@setpar{\@@par\@restorepar\caption@normaladjust}}

```

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

```

583 \providecommand\centerlast{%
584 \let\\\@centercr
585 \leftskip\z@\@plus 1fil%
586 \rightskip\z@\@plus -1fil%
587 \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.

```

588 \DeclareCaptionJustification{Centering}{%
589 \caption@ragged\Centering\centering}
590 \DeclareCaptionJustification{RaggedLeft}{%
591 \caption@ragged\RaggedLeft\raggedleft}
592 \DeclareCaptionJustification{RaggedRight}{%
593 \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.

```

594 \newcommand*\caption@ragged{%
595 \caption@Debug{We need ragged2e}%
596 \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
597 \global\let\caption@ragged\caption@@ragged
598 \caption@ragged}

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

```

```

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

```

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.

```

622 \@ifundefined{abovecaptionskip}{%
623     \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
624 \@ifundefined{belowcaptionskip}{%
625     \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

626 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
627 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
628 \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.

```

629 \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.*

```

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

```

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

The setting 'bottom' corresponds to the `\@makecaption` implementation in the standard L^AT_EX document classes, but 'auto' should give better results in most cases.

```

642 %\caption@setdefaultpos{a}% default = auto
643 \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.)
644 \newcommand*\caption@iftop{%
645   \ifx\caption@position\@undefined
646     \let\caption@position\@secondoftwo
647 %   = \caption@setposition b%
648   \fi
649   \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.
650 \newcommand*\caption@fixposition{%
651   \ifx\caption@position\@undefined
652     \caption@autoposition
653   \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.
654 \newcommand*\caption@autoposition{%
655   \ifvmode
656     \edef\caption@tempa{\the\prevdepth}%
657     \caption@Debug{\protect\prevdepth=\caption@tempa}%
658     \ifdim\prevdepth>-\p@
659       \let\caption@position\@secondoftwo
660     \else
661       \let\caption@position\@firstoftwo
662     \fi
663 %   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
664   \else
665     \caption@Debug{no \protect\prevdepth}%
666     \let\caption@position\@secondoftwo
667 %   = \caption@setposition b%
668   \fi}

\caption@setautoposition \caption@setautoposition{<position>}
replaces the above algorithm by a different one (or a fixed position setting).
669 \newcommand*\caption@setautoposition[1]{%
670   \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`.

```
671 \newcommand*\caption@beginhook{}
672 \newcommand*\caption@endhook{}
673 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
674 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}
```

1.17 Lists

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

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

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

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

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

```
685 \DeclareCaptionListFormat{empty}{}
686 \DeclareCaptionListFormat{simple}{#1#2}
687 \DeclareCaptionListFormat{parens}{#1(#2)}
688 \DeclareCaptionListFormat{subsimple}{#2}
689 \DeclareCaptionListFormat{subparens}{(#2)}
690 \def\caption@lstfmt@default{\caption@lstfmt@subsimple}
```

1.18 Debug option

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

1.19 Document classes & Babel support

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

1.19.1 The standard L^AT_EX classes

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



```

704 \sbox\@tempboxa{#1: #2}%
705 \ifdim \wd\@tempboxa >\hsize
706 #1: #2\par
707 \else
708 \global \@minipagefalse
709 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
710 \fi
711 \vskip\belowcaptionskip}}

```

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

```

712 \@ifundefined{@captionheadfont}{}{%
713 \caption@CheckCommand\@makecaption{%
714 % amsart|amsproc|amsbook [2004/08/06 v2.20]
715 \long\def\@makecaption#1#2{%
716 \setbox\@tempboxa\vbox{\color@setgroup
717 \advance\hsize-2\captionindent\noindent
718 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
719 {\@cdr#2\@nil}\@captionfont\upshape\enspace#2}%
720 \unskip\kern-2\captionindent\par
721 \global\setbox\@ne\lastbox\color@endgroup}%
722 \ifhbox\@ne % the normal case
723 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
724 \fi
725 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
726 \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
727 \else % tempboxa contained more than one line
728 \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
729 \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
730 \fi
731 \ifnum\@tempcnta<64 % if the float IS a figure...
732 \addvspace\abovecaptionskip
733 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
734 \else % if the float IS NOT a figure...
735 \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
736 \nobreak
737 \vskip\belowcaptionskip
738 \fi
739 \relax
740 }}

741 \caption@CheckCommand\@makecaption{%
742 % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
743 \long\def\@makecaption#1#2{%
744 \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
745 \setbox\@tempboxa\vbox{\color@setgroup
746 \advance\hsize-2\captionindent\noindent
747 \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
748 {\@cdr#2\@nil}\@addpunct{.}\@captionfont\upshape\enspace#2}%
749 \unskip\kern-2\captionindent\par
750 \global\setbox\@ne\lastbox\color@endgroup}%
751 \ifhbox\@ne % the normal case
752 \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
753 \fi
754 \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line

```

```

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

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

```

1.19.3 The beamer class

```

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

```

```

805 \DeclareDefaultCaptionFont{font}{%
806   \usebeamerfont*{caption}%
807   \usebeamercolor[fg]{caption}}
808 \DeclareDefaultCaptionFont{labelfont}{%
809   \usebeamercolor[fg]{caption name}%
810   \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.)

```

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

```

1.19.4 The KOMA-Script classes

```

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

```

```

852 \DeclareDefaultCaptionFont{labelfont}{\scr@fnt@captionlabel}
853 }

```

1.19.5 The NTG Dutch classes

```

854 \@ifundefined{CaptionFonts}{}{%
855 \caption@CheckCommand\@makecaption{%
856 % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
857 \long\def\@makecaption#1#2{%
858 \vskip\abovecaptionskip
859 \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
860 \ifdim \wd\@tempboxa >\hsize
861 {\CaptionLabelFont#1:} \CaptionTextFont#2\par
862 \else
863 \global \@minipagefalse
864 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
865 \fi
866 \vskip\belowcaptionskip}}
867 \DeclareDefaultCaptionFont{labelfont}{\CaptionLabelFont}
868 \DeclareDefaultCaptionFont{textfont}{\CaptionTextFont}
869 }

```

1.19.6 The frenchb Babel option

```

870 \@ifundefined{FB@makecaption}{}{%
871 \caption@CheckCommand\@makecaption{%
872 % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
873 \long\def\@makecaption#1#2{%
874 \vskip\abovecaptionskip
875 \sbox\@tempboxa{#1\CaptionSeparator #2}%
876 \ifdim \wd\@tempboxa >\hsize
877 #1\CaptionSeparator #2\par
878 \else
879 \global \@minipagefalse
880 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
881 \fi
882 \vskip\belowcaptionskip}}
883 \ifx\@makecaption\STD@makecaption
884 \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
885 \def\caption@frenchb{% supress frenchb warning
886 \let\STD@makecaption\@makecaption
887 \let\FB@makecaption\@makecaption}
888 \else
889 \ifx\@makecaption\@undefined\else
890 \PackageInfo{caption}{%
891 The definition of \protect\@makecaption\space
892 has been changed,\MessageBreak
893 frenchb will NOT customize it}%
894 \fi
895 \fi
896 }

```

1.19.7 The frenchle/pro package

```

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

```

1.20 Execution of options

```

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

```

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`.

```

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

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

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

\caption@@make \caption@@make{<caption label>}{<caption text>}
956 \newcommand\caption@@make[2]{%
957   \begingroup
958   \caption@beginhook

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

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

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

```
977 \@tempdima\captionwidth
978 \caption@ifh{\advance\@tempdima-\caption@indent}%
979 \captionbox\@tempdima{%
```

Typeset the indentation (option `indention=`)

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

```
980 \caption@ifh{%
981 \ifdim\caption@indent=\z@
982 \leavevmode
983 \else
984 \hskip-\caption@indent
985 \fi}%
```

Typeset the caption itself and close the `\captionbox`

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

Typeset the right margin (option `margin=`)

```
987 \@tempdima\captionmargin
988 \ifdim\captionmargin@=\z@ \else
989 \caption@ifoddpagel{\advance\@tempdima\captionmargin@}{}%
990 \fi
991 \hspace\@tempdima
992 \caption@endhook
993 \endgroup
994 \global\caption@starfalse}
```

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

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

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

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

This one does the single-line-check.

```

1015 \newcommand\caption@slc[4]{%
1016   \caption@Debug{Begin SLC}%
1017   \begingroup
1018   \caption@SingleLine
1019   \let\caption@hj\@empty
1020   \caption@calcmargin % calculate #3 if necessary
1021   \caption@prepareslc
1022   \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
1023   \ifdim\wd\@tempboxa>#3%
1024     \endgroup
1025   \else
1026     \endgroup
1027     \caption@SingleLine
1028     #4%
1029   \fi
1030   \caption@Debug{End SLC}}
1031 \newcommand*\caption@SingleLine{%
1032   \expandafter\caption@setup\expandafter{\caption@opt@SingleLine}%
1033   \let\caption@fmt\caption@slfmt}

```

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

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

```

1034 \newcommand*\caption@prepareslc{%
1035   \let\@footnotetext\@gobble\let\@endnotetext\@gobble
1036   \def\label{\caption@withoptargs\@gobbletwo}%
1037   \let\stepcounter\caption@l@stepcounter
1038   \let\refstepcounter\stepcounter\let\H\refstepcounter\stepcounter}
1039 \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.

```

1040 \newcommand*\captionbox{\parbox[b]}

```

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

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

```

1041 \newcommand\caption@@@make[2]{%

```

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

```

1042   \sbox\@tempboxa{#1}%
1043   \ifdim\wd\@tempboxa=\z@
1044     \let\caption@lsep\relax
1045   %   \@capbreakfalse
1046   \fi

```

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

```

1047   \caption@ifempty{#2}{%
1048     \let\caption@lsep\relax
1049   %   \@capbreakfalse
1050   \let\caption@ifstrut\@secondoftwo
1051   }%

```


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

```
1052 \setpar{\@@par\caption@@par}\caption@@par
```

Finally typeset the caption.

```
1053 \caption@hj\captionsize\captionfont\caption@fmt
1054 {\ifcaption@star\else{\captionlabelfont#1}\fi}%
1055 {\ifcaption@star\else{\caption@iflf\captionlabelfont\caption@lsep}\fi}%
1056 {\caption@textfont
1057 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1058 \nobreak\hskip\z@skip % enable hyphenation
1059 \caption@tfmt{#2}%
1060 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1061 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1062 \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.

```
1063 \newcommand\caption@ifempty[1]{%
1064 \def\caption@tempa{#1}%
1065 \def\caption@tempb{\ignorespaces}%
1066 \ifx\caption@tempa\caption@tempb
1067 \let\caption@tempa\@empty
1068 \fi
1069 \ifx\caption@tempa\@empty
1070 \let\caption@ifempty\@secondoftwo
1071 \else
1072 \let\caption@ifempty\@gobbletwo
1073 \fi
1074 \caption@ifempty\@unused}
```

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

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

```
1075 \newcommand*\caption@@par{%
1076 \parindent\caption@parindent\hangindent\caption@hangindent}%

```

1.22 Bonus features

```
eclareFloatingEnvironment \DeclareFloatingEnvironment[<file ext>]{<environment>}[<name>][<list name>]
1077 \providecommand*\DeclareFloatingEnvironment{%
1078 \ifnextchar[%]
1079 \caption@@DeclareFloatingEnvironment\caption@DeclareFloatingEnvironment}
1080 \@onlypreamble\DeclareFloatingEnvironment

1081 \newcommand*\caption@DeclareFloatingEnvironment[1]{%
1082 \caption@@DeclareFloatingEnvironment[lo#1]{#1}}
1083 \@onlypreamble\caption@DeclareFloatingEnvironment

1084 \def\caption@@DeclareFloatingEnvironment[#1]#2{%
1085 \ifundefined{c@chapter}%
1086 {\newcounter{#2}%

```

```

1087 \@namedef{the#2}{\arabic{#2}}%
1088 \@namedef{theH#2}{\arabic{#2}}}%
1089 {\newcounter{#2}[chapter]%
1090 \@namedef{the#2}{\ifnum\c@chapter>\z@ \thechapter.\fi \arabic{#2}}%
1091 \@namedef{theH#2}{\theHchapter.\arabic{#2}}}%
1092 \@namedef{fps@#2}{tbp}%
1093 \@ifundefined{c@float@type}%
1094 {\newcounter{float@type}%
1095 \setcounter{float@type}{\@ifundefined{c@figure}14}}%
1096 {}}%
1097 \caption@Debug{float type '#2'=\the\value{float@type}}%
1098 \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
1099 \addtocounter{float@type}{\value{float@type}}%
1100 \@namedef{ext@#2}{#1}%
1101 \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1102 \newenvironment{#2}{\@float{#2}}{\end@float}%
1103 \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1104 \expandafter\newcommand\csname listof#2s\endcsname{\begin@group
1105 \expandafter\let\expandafter\listfigurename\csname list#2name\endcsname
1106 \expandafter\let\expandafter\ext@figure\csname ext@#2\endcsname
1107 \let\caption@ORI@starttoc\@starttoc
1108 \renewcommand*\@starttoc[1]{%
1109 \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1110 \listoffigures
1111 \endgroup}%
1112 \@ifundefined{l@figure}%
1113 {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1114 {\expandafter\let\csname l@#2\endcsname\l@figure}%
1115 \expandafter\newcommand\csname #2name\endcsname{}%
1116 \edef\@tempa{%
1117 \def\noexpand\@tempa{\expandafter\noexpand\@car#2\@nil}}%
1118 \uppercase\expandafter{\@tempa}%
1119 \edef\@tempb{%
1120 \noexpand\g@addto@macro\noexpand\@tempa{\expandafter\@cdr#2\@nil}}%
1121 \@tempb
1122 \expandafter\let\csname #2name\endcsname\@tempa
1123 \expandafter\newcommand\csname list#2name\endcsname{}%
1124 \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%
1125 \@cons\caption@floatlist{{#2}}%
1126 \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#2}}\relax}
1127 \@onlypreamble\caption@@DeclareFloatingEnvironment
1128 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1129 \@namedef{#1name}{#2}%
1130 \@ifnextchar[{\caption@@@DeclareFloatingEnvironment{#1}}\relax}
1131 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1132 \def\caption@@@DeclareFloatingEnvironment#1[#2]{%
1133 \@namedef{list#1name}{#2}}
1134 \@onlypreamble\caption@@@DeclareFloatingEnvironment
1135 \newcommand*\caption@floatlist{}

\ForFloatingEnvironments \ForFloatingEnvironments{<code with #1>}
1136 \providecommand\ForFloatingEnvironments[1]{%
1137 \caption@AtBeginDocument{%

```

```

1138 \def\@elt##1{#1}%
1139 \caption@floatlist
1140 \let\@elt\relax}%
1141 \@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.

```

1142 \caption@AtBeginDocument{%
1143 \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1144 \ifx\subref\@tempa
1145 \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
1146 \let\caption@setfloattype\@gobble
1147 % \let\sf@ifinlist\caption@ifinlist
1148 \let\sf@ifpositiontop\caption@iftop
1149 \def\@dottedxxxline#1#2#3#4#5#6#7{%
1150 \begingroup
1151 \caption@setfloattype{#1}%
1152 \caption@setoptions{subfloat}%
1153 \caption@setoptions{sub#1}%
1154 \ifnum #3>\@nameuse{c@#2depth}\else
1155 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1156 \fi
1157 \endgroup}%
1158 \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.

```

1159 \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
1160 \caption@removefromoptlist{subfloat}%
1161 \caption@removefromoptlist{subfigure}%
1162 \caption@removefromoptlist{subtable}%
1163 \fi
1164 \let\sf@NEW@subfloat\@undefined}
1165 \def\sf@NEW@subfloat{%
1166 \begingroup
1167 \caption@setfloattype\@capttype
1168 \sf@ifpositiontop{%
1169 \maincaptiontoptrue
1170 }{%
1171 \maincaptiontopfalse
1172 }%
1173 \caption@setoptions{subfloat}%
1174 \caption@setoptions{sub\@capttype}%
1175 \let\sf@oldlabel=\label
1176 \let\label=\subfloat@label
1177 \ifmaincaptiontop\else
1178 \advance\@nameuse{c@\@capttype}\@ne
1179 \fi
1180 \refstepcounter{sub\@capttype}%

```

```
1181 \setcounter{sub\@cuptype @save}{\value{sub\@cuptype}}%
1182 \@ifnextchar [% %] match left bracket
1183 {\sf@@subfloat}%
1184 {\sf@@subfloat[\@empty]}}%
```

2 Main package

2.1 Identification

```
1185 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
1186 \ProvidesPackage{caption}[2007/10/17 v3.1c Customising captions (AR)]

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

1187 \newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
1188 \@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
1189 \RequirePackage{caption3}[2007/10/17] % needs v3.1c or newer
```

2.3 Check against incompatible packages

```
1190 \@ifpackageloaded{caption2}{%
1191   \caption@Error{%
1192     You can't use both, the (obsolete) caption2 *and*\MessageBreak
1193     the (current) caption package}%
1194   \endinput
1195 }{}

1196 \caption@AtBeginDocument{%
1197   \@ifpackageloaded{ftcap}{%
1198     \caption@IncompatiblePackage{ftcap}{}%
1199   }{}%
1200   \@ifpackageloaded{nonfloat}{%
1201     \caption@IncompatiblePackage{nonfloat}{. %
1202       Furthermore you could use \string\captionof\MessageBreak
1203       instead of \noexpand\figcaption or \string\tabcaption}%
1204   }{}%
1205   \@ifpackageloaded{topcapt}{%
1206     \caption@IncompatiblePackage{topcapt}{}%
1207   }{}%
1208 }

\caption@Incompatible Issue an 'incompatible' warning.

1209 \newcommand*\caption@IncompatiblePackage[2]{%
1210   \PackageWarningNoLine{caption}{%
1211     Usage of the #1 package together with the\MessageBreak
1212     caption package is strongly *not* recommended.\MessageBreak
1213     Instead of loading the #1 package you should\MessageBreak
1214     use the caption package option\MessageBreak
1215     \space\space'tableposition=top'\MessageBreak
1216     #2}}

1217 \@onlypreamble\caption@IncompatiblePackage
```

2.4 Check document class

```
1218 \caption@ifbool{documentclass}{}{}%
1219 \PackageWarningNoLine{caption}{%
1220   Unsupported document class (or package) detected,\MessageBreak
```

```

1221     usage of the caption package is *not* recommended}%
1222 \caption@Info{\string\@makecaption\space=\space\meaning\@makecaption}%
1223 }

```

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

```

1224 \@ifundefined{@captionheadfont}{}{}%
1225 \caption@Info{AMS or SMF document class}%
1226 \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
1227 }

```

2.6 Emulation of the KOMA-Script commands

```

1228 \@ifundefined{scr@caption}{}{}%
1229 \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 1230 \def\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
1231 \def\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}

1232 \if@tablecaptionabove
1233 \caption@AtBeginDocument{%
1234 \def\caption@tempa#1{%
1235 \ifpackageloaded{#1}{\caption@Error{%
1236 You can't use the global option\MessageBreak
1237 \space\space'tablecaptionabove'\MessageBreak
1238 and use the #1 package, too}}}%
1239 \caption@tempa{ftcap}%
1240 \caption@tempa{nonfloat}%
1241 \caption@tempa{topcapt}}%
1242 \@tablecaptionabovetrue
1243 \else
1244 \@tablecaptionabovefalse
1245 \fi

\onelinecaptionstrue
\onelinecaptionfalse 1246 \def\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
1247 \def\onelinecaptionfalse{\let\caption@ifslc\@secondoftwo}

1248 \ifonelinecaptions
1249 \onelinecaptionstrue
1250 \else
1251 \onelinecaptionfalse
1252 \fi

\@captionabovetrue
\@captionabovefalse 1253 \def\@captionabovetrue{\let\caption@position\@firstoftwo}
1254 \def\@captionabovefalse{\let\caption@position\@secondoftwo}

\setcapindent
1255 \let\caption@KOMA@setcapindent\@setcapindent
1256 \renewcommand*\@setcapindent[1]{%
1257 \caption@KOMA@setcapindent{#1}\caption@setcapindent}

```

```

1258 \let\caption@KOMA@@setcapindent\@@setcapindent
1259 \renewcommand*\@@setcapindent[1]{%
1260   \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
1261 \newcommand*\caption@setcapindent{%
1262   \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
1263 \@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.)
1264 \expandafter\let\expandafter\caption@KOMA@setcapwidth
1265   \csname\string\setcapwidth\endcsname
1266 \@namedef{\string\setcapwidth}[1]#2{%
1267   \caption@KOMA@setcapwidth[#1]{#2}\caption@setcapwidth{#1}}
1268 \newcommand*\caption@setcapwidth[1]{%
1269   \ifx\#1\\\else
1270     \@ifundefined{cap@margin}{}{%
1271       \def\@tempa{captionbeside}%
1272       \ifx\@tempa\@currenvir\else\caption@Warning{%
1273         Ignoring optional argument [1] of \string\setcapwidth\MessageBreak}%
1274       \fi}%
1275   \fi
1276   \captionsetup{width=\cap@width}}

1277 \def\caption@tempa{\hsize}%
1278 \ifx\caption@tempa\cap@width \else
1279   \caption@setcapwidth{?}
1280 \fi

\setcapmargin
1281 \expandafter\let\expandafter\caption@KOMA@setcapmargin
1282   \csname\string\@setcapmargin\endcsname
1283 \@namedef{\string\@setcapmargin}[1]#2{%
1284   \caption@KOMA@setcapmargin[#1]{#2}\caption@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 \newcommand*\caption@setcapmargin{%
1290   \begingroup
1291     \let\onelinecaptionsfalse\relax
1292     \def\@twoside{0}%
1293     \def\if@twoside{\def\@twoside{1}\iffalse}%
1294     \cap@margin
1295     \def\@tempa{\endgroup}%
1296     \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
1297       \def\hspace##1##{\@firstofone}%
1298       \edef\@tempa{\endgroup
1299         \noexpand\captionsetup{%
1300           twoside=\@twoside,singlelinecheck=0,%
1301           margin={\cap@left,\cap@right}}}%
1302     \fi\fi
1303   \@tempa}

```

```

1304 \ifx\cap@margin\relax \else
1305 \caption@setcapmargin
1306 \fi
1307 }

```

2.7 Option for configuration files

```

1308 \DeclareCaptionOption*{config}[caption]{%
1309 \InputIfFileExists{#1.cfg}%
1310 {\typeout{*** Local configuration file #1.cfg used ***}}%
1311 {\caption@Warning{Configuration file #1.cfg not found}}}

```

2.8 Options for figure and table

```

1312 \DeclareCaptionOption{figureposition}{%
1313 \captionsetup*[figure]{position=#1}}
1314 \@onlypreamble@key{caption}{figureposition}

1315 \DeclareCaptionOption{tableposition}{%
1316 \caption@AtBeginDocument{%
1317 \def\caption@tempa##1{%
1318 \ifpackageloaded{##1}{\caption@Error{%
1319 You can't use the caption package option\MessageBreak
1320 \space\space`tableposition=#1'\MessageBreak
1321 and use the ##1 package, too}}{}}}%
1322 \caption@tempa{ftcap}%
1323 \caption@tempa{nonfloat}%
1324 \caption@tempa{topcapt}}}%
1325 \captionsetup*[table]{position=#1}}
1326 \@onlypreamble@key{caption}{tableposition}

1327 \DeclareCaptionOption{figurename}{\caption@SetFloatName{figure}{#1}}
1328 \@onlypreamble@key{caption}{figurename}
1329 \DeclareCaptionOption{tablename}{\caption@SetFloatName{table}{#1}}
1330 \@onlypreamble@key{caption}{tablename}

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

1331 \newcommand*\caption@SetFloatName[2]{%
1332 \begingroup
1333 \def\@tempa{\endgroup\@firstofone}%
1334 \ifundefined{language\@tempa}{\@tempa}%
1335 \ifundefined{captions\@tempa}{\@tempa}%
1336 \def\@tempa{\endgroup
1337 \expandafter\addto\csname captions\@tempa\endcsname}}}%
1338 \@tempa{\@namedef{#1name}{#2}}}%
1339 \@onlypreamble\caption@SetFloatName

```

2.9 caption v1.x compatibility options

```

1340 \DeclareCaptionOption{compatibility}[1]{\caption@setbool{compatibility}{#1}}
1341 \@onlypreamble@key{caption}{compatibility}

1342 \DeclareCaptionOptionNoValue*{normal}{%
1343 \caption@setformat{normal}%
1344 \caption@setjustification{justified}}
1345 \DeclareCaptionOptionNoValue*{isu}{%

```



```

1346 \caption@setformat{hang}%
1347 \caption@setjustification{justified}}
1348 \DeclareCaptionOptionNoValue*{hang}{%
1349 \caption@setformat{hang}%
1350 \caption@setjustification{justified}}
1351 \DeclareCaptionOptionNoValue*{center}{%
1352 \caption@setformat{normal}%
1353 \caption@setjustification{centering}}
1354 \DeclareCaptionOptionNoValue*{anne}{%
1355 \caption@setformat{normal}%
1356 \caption@setjustification{centerlast}}
1357 \DeclareCaptionOptionNoValue*{centerlast}{%
1358 \caption@setformat{normal}%
1359 \caption@setjustification{centerlast}}

1360 \DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
1361 \DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
1362 \DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
1363 \DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
1364 \DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
1365 \DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}

1366 \DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
1367 \DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
1368 \DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
1369 \DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
1370 \DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
1371 \DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
1372 \DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
1373 \DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
1374 \DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}

1375 \DeclareCaptionOptionNoValue*{nooneline}{\caption@setbool{slc}{0}}

1376 \caption@setbool{ruled}{0}
1377 \DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}

```

2.10 caption2 v2.x compatibility options

```

1378 \DeclareCaptionOptionNoValue*{flushleft}{%
1379 \caption@setformat{normal}%
1380 \caption@setjustification{raggedright}}
1381 \DeclareCaptionOptionNoValue*{flushright}{%
1382 \caption@setformat{normal}%
1383 \caption@setjustification{raggedleft}}

1384 \DeclareCaptionOptionNoValue*{oneline}{\caption@setbool{slc}{1}}

1385 \DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
1386 \caption@Warning{Obsolete option 'ignoreLTcapwidth' ignored}}

```

2.11 Obsolete caption v3.0 options

```

1387 \DeclareCaptionOption*{caption}{%
1388 \caption@setbool{temp}{#1}%
1389 \caption@ifbool{temp}{}{%
1390 \caption@Error{%
1391 The package option 'caption=#1' is obsolete.\MessageBreak
1392 Please pass this option to the subfig package instead\MessageBreak

```

```
1393         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.

```
1394 \DeclareCaptionOption{FPlist}[1]{\caption@setFPoption{list}{#1}}
1395 \DeclareCaptionOption{FPref}[1]{\caption@setFPoption{ref}{#1}}
1396 \@onlypreamble@key{caption}{FPlist}
1397 \@onlypreamble@key{caption}{FPref}

1398 \newcommand*\caption@setFPoption[2]{%
1399   \edef\@tempa{\@car#2\@nil}%
1400   \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
1401 \@onlypreamble\caption@setFPoption

1402 \captionsetup{FPlist=caption,FPref=figure}
```

2.13 hyperref package support options

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

```
1403 \DeclareCaptionOption{hycap}[1]{\caption@setbool{hycap}{#1}}
1404 \DeclareCaptionOption{hycapSPACE}{\def\caption@hycapSPACE{#1}}

1405 \captionsetup{hycap=1,hycapSPACE=.5\baselineskip}
```

2.14 Processing of options

```
1406 \caption@ProcessOptions*{caption}
```

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

```
1407 \caption@AtBeginDocument{%
1408   \DeclareCaptionOption{type}{\caption@settype{#1}}
1409   \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).

```
1410 \newcommand*\caption@settype{%
1411   \caption@@settype{}}

1412 \newcommand*\caption@@settype[1]{%
1413   \caption@teststar{\caption@@settype{#1}}{\@gobble\@firstofone}}
```

```

1414 \newcommand*\caption@@@settype[3]{%
1415 % #1 = "" or "sub"
1416 % #2 = \@gobble in star form, \@firstofone otherwise
1417 % #3 = <type>, e.g. "figure" or "table"
1418 \caption@Debug{type=#3}%
1419 \caption@checkgrouplevel{#1}{\captionsetup{#1type#2\@gobble*=#3}}%
1420 \ifundefined{@#1capttype}{}{%
1421 \edef\@tempa{#3}%
1422 \expandafter\ifx\csname @#1capttype\endcsname\@tempa \else
1423 \caption@iftypewarning{\caption@Warning{%
1424 Caption type was already set to `#3'\MessageBreak}}{}%
1425 \fi}%
1426 \expandafter\edef\csname @#1capttype\endcsname{#3}%
1427 \let\caption@iftypewarning\caption@ifdebug
1428 \@nameuse{#1caption@settypehook}%
1429 \expandafter\caption@setoptions\csname @#1capttype\endcsname
1430 \let\caption@ifrefstepcounter\@secondoftwo
1431 #2{% \@gobble or \@firstofone
1432 \let\@currentlabel\caption@undefinedlabel
1433 % \let\@currentHlabel\@undefined
1434 \ifx\caption@ORI@label\@undefined
1435 \let\caption@ORI@label\label
1436 \def\label{\caption@withoptargs\caption@xlabel}%
1437 \fi
1438 \caption@start}}

```

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

```
1439 \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.

```

1440 \newcommand*\caption@iftype{%
1441 \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 ϵ -TeX)

```

1442 \begingroup\expandafter\expandafter\expandafter\endgroup
1443 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
1444 \caption@Debug{TeX engine: TeX}
1445 \let\caption@checkgrouplevel\@gobbletwo
1446 \else
1447 \caption@Debug{TeX engine: e-TeX}
1448 \newcommand*\caption@checkgrouplevel[2]{%
1449 \ifundefined{#1caption@grouplevel}{%
1450 \ifundefined{caption@grouplevel}{\let\caption@grouplevel\z@}}{%
1451 \ifnum\currentgrouplevel>\caption@grouplevel\relax
1452 \expandafter\edef\csname #1caption@grouplevel\endcsname{%
1453 \the\currentgrouplevel}%
1454 \else
1455 \caption@Error{\string#2 outside group or environment}%
1456 \fi
1457 }{}%
1458 \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.

```

1459 \newcommand*\caption@undefinedlabel{??}

\caption@xlabel
1460 \newcommand*\caption@xlabel[2]{%
1461   \ifx\@currentlabel\caption@undefinedlabel
1462     \caption@Warning{\noexpand\label before \noexpand\caption ignored}%
1463   \else
1464     \caption@ORI@label#1{#2}%
1465   \fi}

\captionof \captionof{<type>}[<lst.entry>]{<heading>}
\captionof* [<lst.entry>]{<heading>}
1466 \def\captionof{\caption@teststar\caption@of{\caption*}\caption}
1467 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}

\captionlistentry \captionlistentry[<float type>]{<list entry>}
1468 \newcommand*\captionlistentry{%
1469   \@testopt\caption@listentry\@capttype}
1470 \def\caption@listentry[#1]#2{%
1471   \caption@refstepcounter{#1}%
1472   \caption@makecurrent{#1}{#2}%
1473   \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.
1474 \def\ContinuedFloat{%
1475   \ifnextchar\@ContinuedFloat{%
1476     \caption@iftype
1477       {\addtocounter\@capttype\m@ne
1478        \caption@ContinuedFloat\@capttype}%
1479     {\caption@Error{\noexpand\ContinuedFloat outside float}}}}
1480 \def\caption@ContinuedFloat#1{%
1481   \edef\@tempa{#1}%
1482   \ifx\@tempa\caption@CFtype \else
1483     \caption@Error{Continued ‘#1’ after ‘\caption@CFtype’}%
1484   \fi
1485   \stepcounter{ContinuedFloat}%
1486   \caption@@ContinuedFloat{#1}%
1487   \caption@setoptions{ContinuedFloat}%
1488   \caption@setoptions{continued#1}%
1489   \let\caption@resetContinuedFloat\@gobble}
1490 \def\@ContinuedFloat[#1]{\addtocounter{#1}\m@ne}

```

```

1491 \newcounter{ContinuedFloat}
1492 \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.)
1493 \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.)
1494 \newcommand*\caption@resetContinuedFloat[1]{%
1495   \@stpel{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.
1496 \newcommand*\caption@refstepcounter[1]{%
1497   \caption@resetContinuedFloat{#1}%
1498   \caption@@refstepcounter{#1}%
1499   \let\caption@ifrefstepcounter\@firstoftwo}

1500 \newcommand*\caption@@refstepcounter{\refstepcounter}
1501 \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[]{}).
1502 \newcommand\caption@dblarg[1]{\kernel@ifnextchar[{\#1}{\caption@xdblarg{#1}}}%
1503 \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)

1504 \newcommand*\caption@begin[1]{%
1505   \begingroup
1506   \caption@setfnum{#1}%
1507   \caption@fixposition
1508   \global\let\caption@fixedposition\caption@position}

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

5. Make an entry in the list-of-whatever.

6. Set \caption@ifempty according argument <heading>.

```

```

1509 \newcommand\caption@beginex[3]{%
1510   \caption@begin{#1}%
1511   \caption@addcontentsline{#1}{#2}%
1512   \caption@ifempty{#3}{}%

```

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

```

1513 \newcommand*\caption@end{%
1514   \endgroup
1515   \let\caption@position\caption@fixedposition}

```

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

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

```

1516 \newcommand*\caption@setfnum[1]{%
1517   \@ifundefined{fnum@#1}{\iftrue}{\ifx\caption@lfmt\caption@lfmt@default\else}%
1518   \@namedef{fnum@#1}{\caption@fnum{#1}}%
1519   \fi}

```

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

```

1520 \caption@AtBeginDocument*{%
1521   \@ifundefined{sf@ifpositiontop}{}{%
1522     \let\caption@kernel@addcontentsline\caption@addcontentsline
1523     \renewcommand*\caption@addcontentsline[2]{%
1524       \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
1525       \caption@kernel@addcontentsline{#1}{#2}%
1526       \sf@ifpositiontop{}{\@listsubcaptions{#1}}}%
1527   }}

```

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

```

\def\@parboxrestore{\@arrayparboxrestore\let\\\@normalcr}
\def\@arrayparboxrestore{%
  \let\if@nbreak\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`.

```

1528 \newcommand*\caption@boxrestore{%
1529   \let\if@nbreak\iffalse
1530   \let\if@noskipsec\iffalse
1531   \let\par\@par
1532   \let\-\@dischyph

```

```

1533 % \let'\@acci\let\'\@accii\let\=\@acciii
1534 \parindent\z@ \parskip\z@skip
1535 \everypar{ }%
1536 % \linewidth\hsize
1537 % \@totalleftmargin\z@
1538 \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
1539 \parfillskip\@flushglue \lineskip\normallineskip
1540 \baselineskip\normalbaselineskip
1541 \sloppy
1542 \let\\\@normalcr
1543 }

```

`\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).

```

1544 \newcommand*\caption@normalsize{%
1545 \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.

```
1546 \let\caption@setfloatcapt\@firstofone
```

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

```

\caption@@anchor 1547 \newcommand*\caption@makecurrent[2]{}
\caption@start 1548 \let\caption@@anchor\@firstofone
\caption@@start 1549 \let\caption@start\relax
\caption@freezeHref 1550 \let\caption@@start\relax
\caption@defrostHref 1551 \let\caption@freezeHref\relax
1552 \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 its own definition of `\caption*`, was quite common these days.

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

```

1553 \caption@AtBeginDocument{%
1554 \caption@setbool{incompatible}{0}%
1555 \caption@CheckCommand\caption{%
1556 % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1557 \def\caption{%
1558 \ifx\@capttype\@undefined
1559 \@latexerror{\noexpand\caption outside float}\@ehd
1560 \expandafter\@gobble

```

```

1561         \else
1562             \refstepcounter\@capttype
1563             \expandafter\@firstofone
1564         \fi
1565         {\@dblarg{\@caption\@capttype}}%
1566     } }%
1567 \caption@CheckCommand\caption{%
1568     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1569     \def\caption{
1570         \ifx\@capttype\@undefined
1571             \@latex@error{\noexpand\caption outside figure or table}\@ehd
1572             \expandafter\@gobble
1573         \else
1574             \refstepcounter\@capttype
1575             \expandafter\@firstofone
1576         \fi
1577         {\@dblarg{\@caption\@capttype}}%
1578     } }%
1579 \caption@CheckCommand\caption{%
1580     % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1581     \renewcommand\caption{%
1582         \ifx\@capttype\@undefined
1583             \@latex@error{\noexpand\caption outside float}\@ehd
1584             \expandafter\@gobble
1585         \else
1586             \refstepcounter\@capttype
1587             \let\@tempf\@caption
1588             \expandafter\ifx\csname @float@c@\@capttype\endcsname\relax\else
1589                 \expandafter\expandafter\let
1590                 \expandafter\@tempf\csname @float@c@\@capttype\endcsname
1591             \fi
1592             \fi
1593             \@dblarg{\@tempf\@capttype}}}%
1594 \caption@CheckCommand\caption{%
1595     % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1596     % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1597     % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1598     \def\caption{%
1599         \ifx\@capttype\@undefined
1600             \@latex@error{\noexpand\caption outside float}\@ehd
1601             \expandafter\@gobble
1602         \else
1603             \H@refstepcounter\@capttype
1604             \@ifundefined{fst@\@capttype}{%
1605                 \let\Hy@tempa\@caption
1606             }{%
1607                 \let\Hy@tempa\Hy@float@caption
1608             }%
1609             \expandafter\@firstofone
1610         \fi
1611         {\@dblarg{\Hy@tempa\@capttype}}%
1612     } }%
1613 \caption@CheckCommand\caption{%

```



```

1614 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1615 \def\caption{%
1616   \ifx\@capttype\undefined
1617     \@latex@error{\noexpand\caption outside float}\@ehd
1618     \expandafter\@gobble
1619   \else
1620     \H@refstepcounter\@capttype
1621     \let\Hy@tempa\@caption
1622     \@ifundefined{float@caption}{%
1623       }{%
1624         \expandafter\ifx\csname @float@c@\@capttype\endcsname\float@caption
1625         \let\Hy@tempa\Hy@float@caption
1626         \fi
1627       }%
1628     \expandafter\@firstofone
1629   \fi
1630   {\@dblarg{\Hy@tempa\@capttype}}%
1631 }%
1632 \caption@ifcheckcommand{}{%
1633   \caption@info{%
1634     Incompatible package detected (regarding \string\caption).\MessageBreak
1635     \string\caption\space=\space meaning\caption}%
1636   \caption@setbool{incompatible}{1}}%
1637 \caption@checkcommand\@caption{%
1638   % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
1639   \long\def\@caption#1[#2]#3{%
1640     \par
1641     \addcontentsline{\csname ext@#1\endcsname}{#1}%
1642     {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
1643     \begingroup
1644       \@parboxrestore
1645       \if@minipage
1646         \@setminipage
1647       \fi
1648       \normalsize
1649       \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
1650     \endgroup}%
1651 \caption@checkcommand\@caption{%
1652   % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
1653   \long\def\@caption#1[#2]#3{% second argument ignored
1654     \par\nobreak
1655     \begingroup
1656       \@parboxrestore
1657       \if@minipage
1658         \@setminipage
1659       \fi
1660       \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
1661     \endgroup}%
1662 % \caption@checkcommand\float@caption{%
1663 %   % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
1664 %   \long\def\float@caption#1[#2]#3{%
1665 %     \addcontentsline{\@nameuse{ext@#1}}{#1}%
1666 %     {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}

```

```

1667 %      \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
1668 %      \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
1669 %      \ifnextchar[{\float@ccon}{\egroup}}%
1670 %      \long\def\float@ccon[#1]{#1\par\egroup}}%

1671 \caption@CheckCommand\@caption{%
1672 % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
1673 \long\def\@caption#1[#2]#3{%
1674   \hyper@makecurrent{\@cuptype}%
1675   \def\@currentlabelname{#2}%
1676   \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1677     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1678   }%
1679   \begingroup
1680     \@parboxrestore
1681     \if@minipage
1682       \setminipage
1683     \fi
1684     \normalsize
1685     \@makecaption{\csname fnum@#1\endcsname}{%
1686       \ignorespaces
1687       \ifHy@nesting
1688         \hyper@@anchor{\@currentHref}{#3}%
1689       \else
1690         \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1691       \fi
1692     }%
1693     \par
1694   \endgroup
1695 }%

1696 \caption@CheckCommand\@caption{%
1697 % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
1698 % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
1699 % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
1700 \long\def\@caption#1[#2]#3{%
1701   \expandafter\ifx\csname if@capstart\expandafter\endcsname
1702     \csname iftrue\endcsname
1703     \global\let\@currentHref\hc@currentHref
1704   \else
1705     \hyper@makecurrent{\@cuptype}%
1706   \fi
1707   \def\@currentlabelname{#2}%
1708   \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
1709     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
1710   }%
1711   \begingroup
1712     \@parboxrestore
1713     \if@minipage
1714       \setminipage
1715     \fi
1716     \normalsize
1717     \expandafter\ifx\csname if@capstart\expandafter\endcsname
1718       \csname iftrue\endcsname
1719     \global\@capstartfalse

```

```

1720         \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
1721     \else
1722         \@makecaption{\csname fnum@#1\endcsname}{%
1723             \ignorespaces
1724             \ifHy@nesting
1725                 \hyper@@anchor{\@currentHref}{#3}%
1726             \else
1727                 \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
1728             \fi
1729         }%
1730     \fi
1731 \par
1732 \endgroup
1733 }}%

1734 \caption@CheckCommand\@caption{%
1735     % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
1736     \long\def\@caption#1[#2]{%
1737         \def\@currentlabelname{#2}%
1738         \NR@@caption{#1}[{#2}]%
1739     }%

1740 \caption@CheckCommand\@caption{%
1741     % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
1742     \long\def\@caption#1[#2]#3{%
1743         \ifundefined{if#1topcap}%
1744             {\subfig@oldcaption{#1}[{#2}]{#3}}%
1745             {\@nameuse{if#1topcap}%
1746                 \@listsubcaptions{#1}%
1747                 \subfig@oldcaption{#1}[{#2}]{#3}%
1748             \else
1749                 \subfig@oldcaption{#1}[{#2}]{#3}%
1750                 \@listsubcaptions{#1}%
1751             \fi}}}%

1752 \caption@CheckCommand\@caption{%
1753     % subfig.sty [2005/06/28 ver: 1.3 subfig package]
1754     \def\@caption{\caption@}%
1755     \long\def\caption@#1[#2]#3{%
1756         \ifundefined{caption@setfloattype}%
1757             \caption@settype
1758             \caption@setfloattype
1759             \@capttype
1760         \sf@ifpositiontop{%
1761             \@listsubcaptions{#1}%
1762             \sf@old@caption{#1}[{#2}]{#3}%
1763         }{%
1764             \sf@old@caption{#1}[{#2}]{#3}%
1765             \@listsubcaptions{#1}%
1766         }%
1767     }%

1768 \caption@IfCheckCommand{}{%
1769     \caption@Info{%
1770         Incompatible package detected (regarding \string\@caption).\MessageBreak
1771         \string\@caption\space=\space\meaning\@caption}%
1772     \caption@setbool{incompatible}{1}}%

```

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

```

1773 \ifundefined{caption@ifcompatibility}%
1774   {\let\caption@ifcompatibility\caption@ifincompatible
1775    \let\@tempa\@firstofone}%
1776   {\let\@tempa\@gobble}% suppress warning
1777 \caption@ifcompatibility{%
1778   \@tempa{%
1779     \caption@WarningNoLine{%
1780       \noexpand\caption will not be redefined since it's already\MessageBreak
1781       redefined by a document class or package which is\MessageBreak
1782       unknown to the caption package.\MessageBreak
1783       As a result some new features of the caption\MessageBreak
1784       package will not work as expected}%
1785   }%

```

`\ContinuedFloat` is not supported in compatibility mode.

```

1786 \renewcommand*\caption@ContinuedFloat[1]{%
1787   \caption@Error{Not available in compatibility mode}}%

```

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

```

1788 \caption@AtBeginDocument*{%
1789   \let\caption@start\relax
1790   \ifundefined{caption@ORI@capstart}{}{}%
1791   \caption@Debug{%
1792     Undo redefinition of \string\capstart\@gobble}%
1793   \let\capstart\caption@ORI@capstart
1794   \let\caption@ORI@capstart\@undefined}%
1795   \ifundefined{Hy@float@makebox}{}{}%
1796   \caption@Debug{%
1797     Restore hyperref redefinition of \string\float@makebox\@gobble}%
1798   \let\float@makebox\Hy@float@makebox
1799   \let\Hy@float@makebox\@undefined}%
1800 }%

```

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

```

1801 \newcommand*\caption@star[2]{#1#2}%
1802 }{%
1803 \caption@ifincompatible{%
1804   \caption@WarningNoLine{%
1805     Forced redefinition of \noexpand\caption since the\MessageBreak
1806     unsupported(!) package option 'compatibility=false'\MessageBreak
1807     was given. Please fasten your seat belts, as we may\MessageBreak
1808     experience turbulence}%
1809   }{}%

```

`\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.

```

1810 \def\caption{%
1811   \caption@iftyp
1812   {\caption@checkgrouplevel\@empty\caption

```

```

1813         \caption@star
1814         {\caption@refstepcounter\@capttype}%
1815         {\caption@dblarg{\@caption\@capttype}}}%
1816         {\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.
1817     \newcommand*\caption@star[2]{%
1818         \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.)
1819     \long\def\@caption#1[#2]#3{%
1820         \ifcaption@star \else
1821             \caption@makecurrent{#1}{#2}%
1822             \caption@ifhycap{\caption@@start}{}%
1823         \fi
1824         \par
1825         \caption@beginex{#1}{#2}{#3}%
1826         \caption@setfloatcapt{%
1827             \caption@boxrestore
1828             \caption@normalsize
1829             \ifcaption@star
1830                 \let\caption@@anchor\@firstofone
1831             \fi
1832             \@makecaption{\csname fnum@#1\endcsname}%
1833             {\ignorespaces\caption@@anchor{#3}}\par}%
1834         \caption@end}%
1835     }%

\@xfloat We redefine \@xfloat so inside floating environments our type-specific options will be
used, a hyperref anchor will be set etc.
1836     \let\caption@ORI@xfloat\@xfloat
1837     \def\@xfloat#1[#2]{%
1838         \caption@ORI@xfloat{#1}[#2]%
1839         \caption@settype{#1}}%
1840 }

\@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.
1841 \long\def\@makecaption#1#2{%
1842     \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
1843     \caption@@make{#1}{#2}%
1844     \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

```
1845 \ifundefined{smf@makecaption}{}{\let\smf@makecaption\@makecaption}
```

2.18.2 The beamer class

```
1846 \ifclassloaded{beamer}{%
1847   \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.

```
1848   \captionsetup{listof=false}
1849   \undefine@key{caption}{listof}
1850   \undefine@key{caption}{listofformat}
1851 }{ }
```

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.

```
1852 \ifundefined{scr@caption}{}{%
1853   \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)

```
1854 \ifundefined{caption@frenchb}{}{\caption@frenchb}
```

2.18.5 The frenchle/pro package

```
1855 \caption@AtBeginDocument{\ifundefined{frenchTeXmods}{}{%
1856   \caption@Info{frenchle/pro package is loaded}%
1857   \let\captionfont@ORI\captionfont
1858   \let\captionlabelfont@ORI\captionlabelfont
1859   \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`.

```
1860   \ifundefined{GOfrench}%
1861     {\let\@tempa\@firstofone}%
1862     {\def\@tempa{\g@addto@macro\GOfrench}}%
1863   \@tempa{%
1864     \let\captionfont\captionfont@ORI
1865     \let\captionfont@ORI\undefined
1866     \let\captionlabelfont\captionlabelfont@ORI
1867     \let\captionlabelfont@ORI\undefined
1868     \let\@makecaption\@makecaption@ORI
1869     \let\@makecaption@ORI\undefined
```

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

```
1870     \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.
1871 \let\caption@tcORI\@tablescaption
1872 \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.
1873 \let\@eatDP\@undefined
1874 \let\@tempa\@empty
1875 \ifx\ffrench\figure
1876 \l@addto@macro\@tempa{\let\fnum@figure\ffrench}%
1877 \fi
1878 \ifx\tfrench\table
1879 \l@addto@macro\@tempa{\let\fnum@table\tfrench}%
1880 \fi
1881 \def\ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
1882 \def\tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
1883 \@tempa

1884 }%
1885 }

```

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.
1886 \newcommand\caption@ifpackageloaded[1]{%
1887 \testopt{\caption@ifpackageloaded{#1}}{}%
1888 \@onlypreamble\caption@ifpackageloaded
1889 \long\def\caption@ifpackageloaded#1[#2]#3#4{%
1890 \ifpackageloaded{#1}\@firstofone{%
1891 \caption@Debug{#1 package is not loaded (yet)\@gobble}%
1892 \caption@AtBeginDocument}{%
1893 \caption@ifpackageloaded{#1}[#2]{#3}{#4}}%
1894 \@onlypreamble\caption@ifpackageloaded
1895 \newcommand\caption@ifpackageloaded[1]{%
1896 \testopt{\caption@ifpackageloaded{#1}}{}%
1897 \@onlypreamble\caption@ifpackageloaded
1898 \long\def\caption@ifpackageloaded#1[#2]{%
1899 \ifpackageloaded{#1}{%
1900 \caption@Info{#1 package is loaded}%
1901 \ifpackagelater{#1}[#2]\@firstoftwo{%
1902 \caption@Error{%
1903 For a successful cooperation we need at least version\MessageBreak
1904 `#2' of package #1,\MessageBreak
1905 but only version\MessageBreak

```

```

1906         '\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
1907         is available}%
1908         \@secondoftwo}%
1909     }\@secondoftwo}}
1910 \@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.

```

1911 \newcommand*\caption@clearmargin{%
1912     \setcaptionmargin\z@
1913     \let\caption@minmargin\@undefined}

1914 \caption@setbool{needfreeze}{0}
1915 \caption@AtBeginDocument*{%
1916     \caption@ifneedfreeze{%

```

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

```

1917 \newcommand*\caption@freeze{%
1918     \caption@teststar\caption@@freeze\@gobble\@firstofone}%

1919 \newcommand*\caption@@freeze[1]{%
1920     \global\let\caption@SCcontinued\relax
1921     \global\let\caption@SCsetup\@undefined
1922     \global\let\caption@SClentry\@undefined
1923     \global\let\caption@SCtext\@undefined
1924     \global\let\caption@SClabel\@undefined

1925     \let\caption@ORI@ContinuedFloat\ContinuedFloat
1926     \def\ContinuedFloat{%
1927         \let\caption@ORI@setcounter\setcounter
1928         \let\caption@ORI@addtocounter\addtocounter
1929         \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
1930         \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
1931         \caption@ORI@ContinuedFloat
1932         \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
1933         \let\setcounter\caption@ORI@setcounter
1934         \let\addtocounter\caption@ORI@addtocounter}%
1935     \def\captionsetup##1{%
1936         \caption@g@addto@list\caption@SCsetup{##1}%
1937         \caption@setup{##1}}%
1938     \let\caption@ORI\caption
1939     \def\caption{%
1940         \def\caption{\caption@Error{%
1941             Only one \noexpand\caption can be placed in this environment}}%
1942         \let\captionsetup\caption@setup
1943         \let\caption@@refstepcounter\caption@l@stepcounter
1944         \caption@ORI}%
1945     \long\def\@caption##1[##2]##3{%
1946         \gdef\caption@SClentry{##2}\gdef\caption@SCtext{##3}%
1947         \ignorespaces}%
1948     #1{% is \@gobble in star form, and \@firstofone otherwise
1949         \def\label##1{\gdef\caption@SClabel{##1}\ignorespaces}}%

1950 }%

```



```

\caption@defrost \caption@defrost
1951 \newcommand*\caption@defrost{%
1952   \ifx\caption@ORI@ContinuedFloat\@undefined
1953     \caption@SCcontinued
1954     \ifx\caption@SCsetup\@undefined \else
1955       \expandafter\captionsetup\expandafter{\caption@SCsetup}%
1956     \fi
1957     \ifx\caption@SCtext\@undefined \else
1958       \expandafter\expandafter\expandafter\caption
1959       \expandafter\expandafter\expandafter[%
1960       \expandafter\expandafter\expandafter{%
1961       \expandafter\caption@SClentry\expandafter}\expandafter]%
1962       \expandafter{\caption@SCtext}%
1963     \fi
1964     \ifx\caption@SClabel\@undefined \else
1965       \expandafter\label\expandafter{\caption@SClabel}%
1966     \fi
1967   \else
1968     \caption@Error{Internal Error:\MessageBreak
1969     \noexpand\caption@defrost in same group as \string\caption@freeze}%
1970   \fi}%

1971 }{}%
1972 \caption@undefbool{needfreeze}}

```

2.19.1 The float package

The float package usually do not use the L^AT_EX 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.)

```

1973 \caption@ifPackageLoaded{float}[2001/11/08 v1.3d]{%
1974 \@ifpackageloaded{floatrow}{%
1975 \caption@ifpackageloaded{floatrow}[2007/08/24 v0.2a]{}{}%
1976 }{}%

```

`\@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>]`.

```

1977 \let\caption@ORI@float@setevery\@float@setevery
1978 \def\@float@setevery#1{%
1979   \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.

```
1980      \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.

```
1981      \renewcommand\caption@setfloatcapt{%
1982      \let\@makecaption\caption@make
1983      \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.

```
1984      \float@getstyle\float@style{#1}%
1985      \caption@setstyle*\float@style
1986      \caption@setoptions\float@style
1987      }{}%
1988      \caption@freezeHref % will be defrosted in \float@makebox
1989      \caption@ORI@float@setevery{#1}}%
```

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

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

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

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

```
1990      \g@addto@macro\caption@settypehook{%
1991      \expandafter\ifx\csname #1name\endcsname\relax
1992      \expandafter\let\csname #1name\expandafter\endcsname
1993      \csname fname@#1\endcsname
1994      \fi}%
```

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

```
1995      \g@addto@macro\fs@plaintop{\def\@fs@mid{\vspace\abovecaptionskip\relax}}%
1996      \g@addto@macro\fs@boxed{\def\@fs@mid{\kern\abovecaptionskip\relax}}%
```

```
\float@ifstyle \float@ifstyle{⟨type⟩}{⟨if-clause⟩}{⟨else-clause⟩}
```

Checks if the given $\langle type \rangle$ (e.g. figure) is associated with a float style (e.g. boxed).

```
1997 \providecommand*\float@ifstyle[1]{%
1998   \expandafter\ifx\csname fst@#1\endcsname\relax
1999   \expandafter\@secondoftwo
2000   \else
2001   \expandafter\@firstoftwo
2002   \fi}%
```

$\backslash float@getstyle$ $\backslash float@getstyle\{\langle cmd \rangle\}\{\langle type \rangle\}$

Determining the float style is not so easy because the only hint provided by the float package is the macro $\backslash fst@\langle float type \rangle$ which points to the macro which represents the float style. So for example after

```
\floatstyle{ruled}
\newfloat{Program}{tbp}{lop}
```

$\backslash fst@Program$ will be defined as

```
\def\fst@Program{\fs@ruled} .
```

So here is what we do: We make the first level expansion of $\backslash fst@\langle float type \rangle$ a string so we can gobble the first four tokens (= $\backslash fs@$), so only the the name of the float style is left.

TODO: We need to convert the catcodes here.

```
2003 \providecommand*\float@getstyle[2]{%
2004   \edef#1{%
2005     \noexpand\expandafter\noexpand\@gobblefour\noexpand\string
2006     \expandafter\expandafter\expandafter\noexpand
2007     \csname fst@#2\endcsname}%
2008   \edef#1{#1}%
2009   \caption@Debug{floatstyle{#2} = '#1'}}%
```

$\backslash float@setstyle$ $\backslash float@setstyle\{\langle type \rangle\}\{\langle style \rangle\}$

Sets or changes the float style associated with $\langle type \rangle$.

```
2010 \providecommand*\float@setstyle[2]{%
2011   \expandafter\edef\csname fst@#1\endcsname{%
2012     \expandafter\noexpand\csname fs@#2\endcsname}}%
```

$\backslash float@dostyle$ $\backslash float@dostyle\{\langle type \rangle\}$

```
2013 \providecommand*\float@dostyle[1]{%
2014   \@nameuse{fst@#1}\@float@setevery{#1}}%
```

$\backslash float@ifcaption$ $\backslash float@ifcaption\{\langle type \rangle\}\{\langle if-clause \rangle\}\{\langle else-clause \rangle\}$

Here we determine if the user has used $\backslash newfloat$ resp. $\backslash restylefloat$, or $\backslash restylefloat*$. This is quite easy: If $\backslash @float@c@\langle captype \rangle$ is the same as $\backslash float@caption$, the user has used $\backslash newfloat$ or $\backslash restylefloat$, otherwise we assume he has used $\backslash restylefloat*$. (This test will fail if some package re-defines $\backslash float@caption$, so we have to assume that there is no one.)

```
2015 \providecommand*\float@ifcaption[1]{%
2016   \expandafter\ifx\csname @float@c@#1\endcsname\float@caption
2017   \expandafter\@firstoftwo
2018   \else
2019   \expandafter\@secondoftwo
2020   \fi}%
```

```

2021 }}{%
2022 \providecommand*\float@ifstyle[1]{\@secondoftwo}%
2023 \providecommand*\float@ifcaption[1]{\@secondoftwo}%
2024 % \clearcaptionsetup{boxed}% used by the floatrow package?
2025 }

```

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

```

2026 \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’.

```

2027 \caption@ifbool{ruled}{%
2028   \captionsetup[ruled]{margin=0pt,minmargin=0,singlelinecheck=0}%
2029 }{%
2030   \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
2031 }
2032 \caption@undefbool{ruled}

```

2.19.2 The floatflt package

```

2033 \caption@ifPackageLoaded{floatflt}[1996/02/27 v1.3]{%

```

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

```

2034 \let\caption@ORI@floatingfigure\floatingfigure
2035 \def\floatingfigure{%
2036   \caption@floatflt{figure}%
2037   \caption@ORI@floatingfigure}%

```

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

```

2038 \let\caption@ORI@floatingtable\floatingtable
2039 \def\floatingtable{%
2040   \caption@floatflt{table}%
2041 %   \caption@setautoposition b%
2042   \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.

```

2043 \newcommand*\caption@floatflt[1]{%
2044   \caption@settype{#1}%
2045   \caption@clearmargin
2046   \caption@setoptions{floating#1}%
2047   \let\caption@boxrestore\@parboxrestore}%
2048 }{}

```

2.19.3 The fltpage package

```
2049 \caption@ifPackageLoaded{fltpage}[1998/10/29 v.0.3]{%
2050   \caption@setbool{needfreeze}{1}%
```

\FP@helpNote **Original code:**

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

2051   \renewcommand\FP@helpNote[2]{%
2052     \begingroup % save \caption@thepage
2053       \caption@pageref{#2}%
2054       \typeout{FP#1 is inserted on page \caption@thepage!}%
2055     \endgroup}%
```

\FP@floatBegin **Original code:**

```
\newcommand{\FP@floatBegin}[1]{%
  \gdef\@captive{#1}%
  \global\let\FP@savedCaptionCommand\caption%
  \global\let\FP@savedLabelCommand\label%
  \ifthenelse{\equal{\@captive}{figure}}%
    {\global\let\old@Fnum\Fnum@figure}%
    {\global\let\old@Fnum\Fnum@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}%
}%

2056   \renewcommand*\FP@floatBegin[1]{%
2057     \def\@captive{#1}%
2058     \let\FP@LabelText\@empty
2059     \begin{lrbox}{\FP@floatCorpusBOX}%
2060       \caption@ifFPrefcap
2061         {\caption@freeze\relax}%
2062         {\def\label##1{\gdef\FP@LabelText{##1}\ignorespaces}}%
2063       \caption@freeze*}%%
```

\FP@floatEnd **Original code:**

```
\newcommand{\FP@floatEnd}{%
  \end{lrbox}%
  \global\setbox\FP@floatCorpusBOX=\box\FP@floatCorpusBOX
  \stepcounter{FP@\@captive C}%
  \FP@savedLabelCommand{\FP@positionLabel}%
  \FP@helpNote{\@captive}{\FP@positionLabel}%
  \FP@float
    {\FP@positionLabel}% location label test
  {\begin{\@captive}[p!]
    \usebox{\FP@floatCorpusBOX}}%
```

```

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

2064 \renewcommand*\FP@floatEnd{%
2065 \end{lrbox}%

2066 \stepcounter{FP@\@capytype C}%
2067 \caption@label\FP@positionLabel
2068 \FP@helpNote\@capytype\FP@positionLabel

2069 \let\FP@ContinuedFloat\relax
2070 \FP@float
2071 {\FP@positionLabel}% location label test
2072 {\begin\@capytype[p!]}%
2073 \FP@ContinuedFloat
2074 \usebox\FP@floatCorpusBOX
2075 \caption@ifFPlistcap
2076 {\caption@refstepcounter\@capytype
2077 \expandafter\caption@makecurrent\expandafter\@capytype
2078 \expandafter{\caption@SClentry}}%
2079 {\expandafter\caption@listentry\expandafter{\caption@SClentry}}%
2080 \caption@@anchor\relax
2081 \ifx\FP@LabelText\@empty \else
2082 \expandafter\label\expandafter{\FP@LabelText}%
2083 \fi
2084 \end\@capytype}%
2085 {\let\FP@ContinuedFloat\ContinuedFloat
2086 \caption@ifcompatibility
2087 {\let\caption@ContinuedFloat\@gobble}%
2088 {\caption@clearsetup{ContinuedFloat}%
2089 \caption@clearsetup{continued\@capytype}}}%
2090 {\begin\@capytype[b!]}%
2091 \FP@ContinuedFloat
2092 \let\FP@savedSetfnumCommand\caption@setfnum
2093 \def\caption@setfnum##1{%
2094 \FP@savedSetfnumCommand{##1}%
2095 \ifx\FP@guide\@empty \else
2096 \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ {\FP@guide}}%

```

```

2097         \fi}%
2098         \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
2099         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
2100         \caption@setoptions{FP\@capttype}%
2101         \FP@separatorCaption
2102         \caption@ifFPlistcap{}\{\caption@set@bool\caption@iflof0}%
2103         \caption@defrost
2104     \end\@capttype}%
2105 }%

2106 \ForFloatingEnvironments{%
2107     \newenvironment{FP#1}\FP@floatBegin{#1}\FP@floatEnd}%
2108 }{%
2109     \let\caption@ifFPlistcap\@undefined
2110     \let\caption@ifFPPrefcap\@undefined
2111 }

```

2.19.4 The hyperref package

```

2112 \caption@ifPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
2113     \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
2114         \caption@WarningNoLine{%
2115             Hyperref support is turned off\MessageBreak
2116             because hyperref has stopped early}%
2117     }{%
2118         \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`.

```

2119     \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`.

```

2120     \renewcommand*\caption@makecurrent[2]{%
2121         \caption@makecurrentHref{#1}%
2122         \caption@Debug{hyperref current=\@currentHref}%
2123         \def\@currentlabelname{#2}}%
2124     \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`.

```

2125     \renewcommand\caption@@anchor[1]{%
2126         \caption@Debug{hyperref anchor: \@currentHref}%
2127         % If we cannot have nesting, the anchor is empty.
2128         \ifHy@nesting
2129             \hyper@@anchor{\@currentHref}{#1}%
2130         \else
2131             \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#1%
2132         \fi}%
2133     \g@addto@macro\caption@prepareslc{\let\caption@@anchor\@firstofone}%

```

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

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

will be used instead of

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

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

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

```

2134 \renewcommand*\caption@@ContinuedFloat[1]{%
2135 \ifundefined{theH#1}}{%
2136 \expandafter\l@addto@macro\csname theH#1\endcsname{%
2137 \@alph\c@ContinuedFloat}}}%

```

The `hypcap` option

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

2138 `\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`.

2139 `\def\caption@start{\caption@ifhypcap{%`
Generate the `hyperref` label and set the `hyperref` anchor, usually (if `hypcap=false`) both is done inside `\@caption`.

```

2140 \caption@makestart\@captype
2141 \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.)

```

2142 \global\@capstarttrue
2143 \let\hc@currentHref\@currentHref
2144 \def\caption@makecurrentHref##1{%
2145 \global\@capstartfalse
2146 \global\let\@currentHref\hc@currentHref}%

```

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

```

2147 \let\caption@@anchor\@firstofone
2148 }{}%

```

`\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.

```
2149 \newcommand*\caption@makestart[1]{%
2150 \begingroup
2151 \Hy@hypertextnamesfalse
2152 % \gdef\@currentHlabel{}%
2153 \hyper@makecurrent{#1.caption}%
2154 \endgroup
2155 \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.

```
2156 \newcommand*\caption@startanchor[1]{%
2157 \caption@Debug{hycap anchor: #1}%
2158 \ifvmode\begingroup
2159 \@tempdima\prevdepth
2160 \nointerlineskip
2161 \vspace*{-\caption@hycapspace}%
2162 \caption@anchor{#1}%
2163 \vspace*{\caption@hycapspace}%
2164 \prevdepth\@tempdima
2165 \endgroup\else
2166 \caption@Warning{Internal Warning: Vertical TeX mode expected}%
2167 \caption@anchor{#1}%
2168 \fi}%%
```

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

```
2169 \newcommand*\caption@anchor[1]{%
2170 \ifmeasuring@ \else
2171 \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2172 \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`.)

```
2173 \ifx\HyperRaiseLinkLength\@tempdima
2174 \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2175 \else
2176 \let\caption@raisedlink\Hy@raisedlink
2177 \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.

```

2178 \def\caption@@start{%
2179 \ifundefined{hc@currentHref}{%
2180 \caption@Warning{%
2181 The option 'hycap=true' will be ignored for this\MessageBreak
2182 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`.

```

2183 \def\caption@freezeHref{%
2184 \let\caption@ORI@start\caption@start
2185 \def\caption@start{\let\caption@start\caption@ORI@start}%
2186 % \let\caption@ORI@@start\caption@@start
2187 % \l@addto@macro\subcaption@settypehook{%
2188 % \let\caption@@start\caption@ORI@@start}%
2189 \global\let\caption@currentHref\@undefined
2190 \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2191 \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2192 \renewcommand*\caption@setfloatcapt{%
2193 \ifx\caption@currentHref\@undefined \else
2194 \let\caption@@anchor\@firstofone
2195 \fi
2196 \caption@ORI@setfloatcapt}}%

```

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

```

2197 \def\caption@defrostHref{%
2198 \ifx\caption@currentHref\@undefined \else
2199 \caption@startanchor\caption@currentHref
2200 \global\let\caption@currentHref\@undefined
2201 \fi}%

```

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

```

2202 \ifundefined{HyOrg@float@makebox}{}{%
2203 \caption@Debug{%
2204 Redefining \noexpand\float@makebox (again)\@gobble}%
2205 \let\Hy@float@makebox\float@makebox
2206 \renewcommand\float@makebox[1]{%
2207 \HyOrg@float@makebox{#1}\relax \caption@defrostHref}}%
2208 }%
2209 }{}

```

2.19.5 The hycap package

```

2210 \caption@IfPackageLoaded{hycap}{% v1.0
2211 \ifx\caption@start\relax \else % hyperref hasn't stopped early

```

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

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

```
2212 \let\caption@ORI@capstart\capstart
2213 \let\capstart\caption@start
2214 \let\caption@start\relax
2215 \let\caption@@start\relax
```

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

```
2216 \caption@@setbool\caption@ifhypcap1%
2217 \renewcommand*\caption@hypcapspace{\hypcapspace}%
```

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

```
2218 \ifundefined{Hy@float@makebox}{}{%
2219 \caption@Debug{%
2220 Restore hyperref redefinition of \string\float@makebox\@gobble}%
2221 \let\float@makebox\Hy@float@makebox
2222 \let\Hy@float@makebox\@undefined}%
2223 \fi}}
```

2.19.6 The listings package

```
2224 \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`.

```
2225 \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
2226 \def\lst@MakeCaption#1{%
2227 \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.

```
2228 \caption@setautoposition{#1}%
2229 \caption@begin{lstlisting}%
2230 \caption@ORI@lst@MakeCaption{#1}%
2231 \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.

```
2232 \providecommand*\ext@lstlisting{lol}%
2233 }}}
```

2.19.7 The longtable package

`\LTcapytype` `\LTcapytype` is preset to `table`.

```
2234 \providecommand*\LTcapytype{table}
```

```

2235 \caption@ifpackageloaded{longtable}[1995/05/24 v3.14]{%
2236   \RequirePackage{ltcaption}[2007/09/01]%
2237   \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.

2238 \caption@AtBeginDocument{%
2239   \let\caption@ORI@LT@array\LT@array
2240   \renewcommand*\LT@array{%

\captionsetup for longtable:

2241   \global\let\caption@opt@@longtable\@undefined
2242   \def\captionsetup##1{\noalign{\LT@captionsetup{##1}}}%
2243   \def\LT@captionsetup##1{%
2244     \@captionsetup@gobble[@longtable]{##1}%
2245     \global\let\caption@opt@@longtable\caption@opt@@longtable}%

\captionabove & \captionbelow for longtable: (KOMA-Script document class)

2246   \def\@captionabovetrue{\LT@captionsetup{position=t}}%
2247   \def\@captionabovefalse{\LT@captionsetup{position=b}}%

\captionlistentry for longtable:

2248   \def\captionlistentry##1{\noalign{%
2249     \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

2250 %   \caption@ifhyhpcap{%
2251 %     \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
2252 %     \def\hyper@makecurrent##1{%
2253 %       \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
2254 %       \caption@makestart{##1}%
2255 %       \let\Hy@LT@currentHlabel\@currentHlabel
2256 %       \let\Hy@LT@currentHref\@currentHref
2257 %       \def\hyper@makecurrent####1{%
2258 %         \let\@currentHlabel\Hy@LT@currentHlabel
2259 %         \let\@currentHref\Hy@LT@currentHref}}%
2260 %     \let\caption@ORI@ContinuedFloat\ContinuedFloat
2261 %     \def\ContinuedFloat{\noalign{%
2262 %       \gdef\caption@setContinuedFloat{%
2263 %         \let\caption@resetContinuedFloat\@gobble}%
2264 %       \def\caption@setoptions####1{%
2265 %         \g@addto@macro\caption@setContinuedFloat{%
2266 %           \caption@setoptions{####1}}}%
2267 %       \let\@capttype\LTcapttype
2268 %       \caption@ORI@ContinuedFloat}}%
2269 %   }{%
2270 %     \def\ContinuedFloat{\noalign{%
2271 %       \caption@Error{%
2272 %         \noexpand\ContinuedFloat inside longtables\MessageBreak
2273 %         is only available with 'hyhpcap=true'}}}%

```

```

2274 %      }%
2275 %      \global\let\caption@setContinuedFloat\@empty
2276 %      \def\ContinuedFloat{\noalign{%
2277 %          \caption@Error{\noexpand\ContinuedFloat outside float}}}%
2278 %      \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}`:

```

2279 \def\LT@c@ption#1[#2]#3{%
2280   \LT@makecaption#1{\csname fnun@\LTcapttype\endcsname}{#3}%
2281   \def\@currentlabelname{#2}%
2282   \caption@addcontentsline\LTcapttype{#2}}%

```

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

The original definition:

```

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

```

Our definition:

```

2283 \renewcommand\LT@makecaption[3]{%
2284   \caption@LT@make{%

```

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

```

2285   \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`.)

```

2286   \caption@settype*\LTcapttype
2287   \ifdim\LTcapwidth=4in \else
2288     \setcaptionwidth\LTcapwidth
2289   \fi
2290   \caption@setoptions{longtable}%

```

```

2291 %      \caption@setContinuedFloat
2292      \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!

```

2293      \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
2294      \caption@resetContinuedFloat\LTcapttype
2295      \caption@begin\LTcapttype
2296      \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...)

```

2297      \vskip-\ht\strutbox

```

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

```

2298      \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
2299      \caption@@make{#2}{#3}\endgraf
2300      \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
2301      \caption@end}}%
2302 }{ }

```

2.19.8 The picinpar package

```

2303 \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.

```

2304 \long\def\figwindow[#1,#2,#3,#4] {%
2305   \caption@window{figure}%
2306   \caption@setoptions{figwindow}%
2307   \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
2308 \long\def\tabwindow[#1,#2,#3,#4] {%
2309   \caption@window{table}%
2310   \caption@setoptions{tabwindow}%
2311   \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.

```

2312 \newcommand*\caption@window[1]{%
2313   \let\caption@boxrestore\@parboxrestore
2314   \let\@makecaption\caption@@make
2315   \caption@setautoposition b%

```

```

2316 \caption@settype{#1}%
2317 \caption@clearmargin}%

```

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

```

2318 \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.

```

```

2319 \ifdim\picwd=\z@
2320 \let\caption@makecurrent\@gobbletwo
2321 \let\caption@@start\relax
2322 \caption@prepareslc
2323 \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.

```

2324 \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
2325 \if\@tempa*
2326 \let\@tempa\@firstofone
2327 \else\if\@tempa[%
2328 \let\@tempa\@firstofone
2329 \else
2330 \let\@tempa\@empty
2331 \fi\fi
2332 \expandafter\caption\@tempa{#1}}%
2333 {}

```

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.

```

2334 \newcommand*\piccaptiontype[1]{\def\@piccaptiontype{#1}}

```

```

2335 \caption@IfPackageLoaded{picins}{%

```

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

```

2336 \@ifundefined{@piccaptiontype}{%
2337 \caption@iftype{%
2338 \let\@piccaptiontype\@capytype
2339 }{%
2340 \def\@piccaptiontype{figure}%
2341 }%
2342 }{}%
2343 \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.

```
2344 \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.

```
2345 \let\caption@ORI@make@piccaption\make@piccaption
2346 \def\make@piccaption{%
2347   \let\caption@ORI\caption
2348   \long\def\caption[##1]##2{%
2349     \caption@freezeHref % will be defrosted in \ivparpic
2350     \caption@settype\@piccapttype
2351     \ifnum\c@piccaptionpos>2\relax
2352       \caption@clearmargin
2353     \else
2354       \captionwidth\z@ % do not use "width=" setting
2355     \fi
2356     \caption@setoptions{parpic}%
2357     \let\caption@boxrestore\@parboxrestore
2358     \caption@setautoposition b%
2359     \expandafter\expandafter\expandafter\caption@ORI
2360       \expandafter\expandafter\expandafter[%
2361       \expandafter\expandafter\expandafter{%
2362       \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
```



```

2363 \caption@ORI@make@piccaption
2364 \let\caption\caption@ORI}%

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

2365 \let\caption@ORI@ivparpic\ivparpic
2366 \def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
2367 \let\caption@ORI@noindent\noindent
2368 \def\noindent{%
2369 \caption@defrostHref
2370 \let\noindent\caption@ORI@noindent
2371 \noindent}%
2372 \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
2373 \let\noindent\caption@ORI@noindent}%

2374 }{%
2375 \let\piccaptiontype\@undefined
2376 }

```

2.19.10 The rotating package

```

2377 \caption@IfPackageLoaded{rotating}[1995/08/22 v2.10]{%

\rotcaption Make \rotcaption* work.

2378 \def\rotcaption{\let\@makecaption\@makerotcaption\caption}%
2379 % \let\@rotcaption\@undefined

\rotcaptionof Make \rotcaptionof(*) work.

2380 \def\rotcaptionof{%
2381 \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.)

```

2382 \long\def\@makerotcaption#1#2{%
2383 \ifdim\captionwidth=\z@
2384 \setcaptionwidth{.8\textheight}%
2385 \caption@slc{#1}{#2}{.8\vsiz}%
2386 \let\caption@makerot\caption@@make
2387 \caption@clearmargin

```

```

2388 %      \long\def\captionbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
2389 %      (not needed because \rotatebox uses an \hbox anyway)
2390      \let\captionbox\@secondoftwo}%
2391      \caption@set@bool\caption@ifslc0% been there, done that
2392      \fi
2393      \rotatebox{90}{\caption@makerot{#1}{#2}}%
2394      \nobreak\hspace{12pt}}}%
2395 \newcommand\caption@makerot[2]{%
2396   \begin{minipage}\captionwidth\caption@make{#1}{#2}\end{minipage}}%
2397 \ForFloatingEnvironments{%
2398   \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
2399   \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
2400 }{}

```

2.19.11 The sidecap package

```

2401 \caption@ifPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
2402   \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.)
2403   \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.
2404   \let\caption@ORI@SC@zfloat\SC@zfloat
2405   \def\SC@zfloat#1#2#3[#4]{%
              First we use the original definition, but save & restore \caption so \caption@freeze
              will work correctly.
2406       \let\caption@ORI\caption
2407       \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
2408       \let\caption\caption@ORI
              Since the sidecap package uses our \caption code outside the environment the reg-
              ular \captionsetup will not work. So we need a special version here which saves
              the given argument list which will be executed later on. Furthermore we need to make
              \caption* work.
2409       \caption@settype*{#2}%
2410       \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.
2411   \let\caption@ORI@endSC@FLOAT\endSC@FLOAT
2412   \def\endSC@FLOAT{%
              Note: \@cptype isn't defined here, this will be done inside the original definition of
              \endSC@FLOAT. But \SC@cptype is defined and can be used here, if needed.
2413       \let\caption@ORI@settype\caption@settype
2414       \def\caption@settype##1{% will be done in \@xfloat
2415         \caption@ORI@settype*{##1}% do not change \@currentlabel
2416         \caption@setSC@justify

```

```

2417 %%% \caption@setoptions{SCfloat}%
2418 \caption@setoptions{SC\@capttype}%
2419 \caption@start}%

```

Before we can typeset the caption we need to set the margin to zero because any extra margin would only be disturbing here.

(We don't need to take care about the caption position because the sidecap package set both `\abovecaptionskip` and `\belowcaptionskip` to a skip of zero anyway.)

Furthermore `\SC@justify` will override the caption justification, if set. The usage of `\SC@justify` differs from version to version of the sidecap package:

Version 1.4: `\SC@justify` is not defined

Version 1.5: `\SC@justify` is `\relax` when not set

Version 1.6: `\SC@justify` is `\@empty` when not set

```

2420 \def\caption@setSC@justify{%
2421 \caption@clearmargin
2422 \@ifundefined{SC@justify}{}{%
2423 \ifx\SC@justify\@empty \else
2424 \let\caption@hj\SC@justify
2425 \let\SC@justify\@empty
2426 \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!

```

2427 \let\caption\SC@orig@caption
2428 \def\SC@orig@caption[#1]##2{\caption@defrost}%

```

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

```

2429 \caption@setSC@justify % for compatibility mode
2430 \caption@ORI@endSC@FLOAT}%

2431 \ForFloatingEnvironments{%
2432 \newenvironment{SC#1}%
2433 {\SC@float[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@float}%
2434 \newenvironment{SC#1*}%
2435 {\SC@dblfloat[\@nameuse{SC@#1@vpos}]{#1}}{\endSC@dblfloat}}%
2436 {} }

```

2.19.12 The subfigure package

```

2437 \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.

```

2438 \def\sfc@ifpositiontop{%
2439 \ifx\@capttype\@undefined
2440 \expandafter\@gobbletwo
2441 \else\ifx\@capttype\relax
2442 \expandafter\expandafter\expandafter\@gobbletwo
2443 \else
2444 \expandafter\expandafter\expandafter\sfc@if@position@top
2445 \fi\fi}

```

```

2446 \def\sif@if@position@top{%
2447   \@ifundefined{if\@capttype topcap}%
2448     {\@gobbletwo}%
2449     {\@nameuse{if\@capttype topcap}%
2450       \expandafter\@firstoftwo
2451     \else
2452       \expandafter\@secondoftwo
2453     \fi}}
2454 }{}

```

2.19.13 The supertabular package

```

2455 \caption@IfPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
\tablecaption Make \topcaption* and \bottomcaption* work.
2456 \let\caption@ORI@tablecaption\tablecaption
2457 \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}

2458 \long\def\ST@caption#1[#2]#3{\par%
2459   \caption@settype*{#1}%
2460   \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.

2461 \def\caption@fixposition{%
2462   \caption@setposition{\if@topcaption t\else b\fi}}%

2463 \caption@beginex{#1}{#2}{#3}%
2464 \caption@boxrestore
2465 \caption@normalsize
2466 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
2467 \caption@end}%

2468 }{%
2469 \let\caption@setSTposition\@undefined
2470 }

```

2.19.14 The threeparttable package

2471 \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.

```
2472 \let\caption@ORI@threeparttable\threeparttable
2473 \renewcommand*\threeparttable{%
2474   \caption@settype{table}%
2475   \caption@setposition a% ?
2476   \caption@clearmargin
2477   \caption@setoptions{threeparttable}%
2478   \caption@ORI@threeparttable}%
```

\measuredfigure Same here...

```
2479 \let\caption@ORI@measuredfigure\measuredfigure
2480 \renewcommand*\measuredfigure{%
2481   \caption@settype{figure}%
2482   \caption@setposition a% ?
2483   \caption@clearmargin
2484   \caption@setoptions{measuredfigure}%
2485   \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}

2486 \def\TPT@caption#1[#2]#3{%
2487   \gdef\TPT@docapt{%
2488     \global\let\TPT@docapt\@undefined
2489     \caption@setautoposition\caption@TPT@position
2490     \TPT@LA@caption{#1}[\#2][\#3]}%
2491   \ifx\TPT@hsize\@empty
2492     \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
2493     \gdef\caption@TPT@position{t}%
2494     \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
2495   \else
2496     \def\caption@TPT@position{b}%
2497     \TPT@docapt
2498   \fi
2499   \ignorespaces}%

2500 %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
2501 \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima}%

2502 {} }
```

2.19.15 The wrapfig package

2503 \caption@ifPackageLoaded{wrapfig}{% ver 3.3 (Oct 12, 1999)
 \float@ifstyle \float@ifstyle{<type>}{<if-clause>}{<else-clause>}}

(see float package support for details)

```
2504 \providecommand*\float@ifstyle[1]{%
2505   \expandafter\ifx\csname fst@#1\endcsname\relax
2506   \expandafter\@secondoftwo
2507   \else
2508   \expandafter\@firstoftwo
2509   \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.

```
2510 \newcommand*\caption@restylewrapfloat[1]{%
2511   \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
2512   \csname wrap#1\endcsname
2513   \caption@Error{%
2514     For a successful cooperation of the 'wrapfig' package\MessageBreak
2515     with the 'float' package you should load the 'wrapfig'\MessageBreak
2516     package *after* the 'float' package}%
2517   \else
2518     \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
2519     \csname wrap#1\endcsname
2520     \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
2521     \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
2522     \csname wrap#1\endcsname
2523   \fi}%
```

`\caption@wrapfloat`

```
2524 \newcommand*\caption@wrapfloat[1]{%
2525   \caption@settype*{#1}%
2526   \float@ifstyle{#1}{%
2527     \ifx\WF@floatstyhook\@undefined
2528     \caption@Error{%
2529       For a successful cooperation of the 'wrapfig' package\MessageBreak
2530       with the 'float' package you should use at least\MessageBreak
2531       'wrapfig' version 3.6}%
2532     \else
2533     \float@dostyle{#1}%
2534     \fi}{}%
2535   \caption@clearmargin
2536   %%% \caption@setoptions{wrapfloat}%
2537   \caption@setoptions{wrap#1}%
2538   \@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`.)

```

2539 \caption@restylewrapfloat{figure}%
2540 \caption@restylewrapfloat{table}%

2541 \ForFloatingEnvironments{%
2542   \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
2543   \caption@restylewrapfloat{#1}}%
2544 \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.
2545 \ifundefined{float@restyle}{}{%
2546   \toks@=\expandafter{\float@restyle{#1}}% (env may or may not be defined)
2547   \caption@restylewrapfloat{#1}}%
2548   \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
2549   \@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.
2550 \let\caption@ORI@wrapfloat\wrapfloat
2551 \def\wrapfloat#1{%
2552   \float@ifstyle{#1}{%
2553     \caption@Error{%
2554       For a successful cooperation of the 'wrapfig' package\MessageBreak
2555       with the 'float' package you should load the 'wrapfig'\MessageBreak
2556       package *right after* the 'float' package}}{%
2557     \caption@ORI@wrapfloat{#1}}%

2558 \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:
2559 \def\WF@rapt[#1]#2{% final two args: #1 = overhang, #2 = width,
2560   \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
2561   \global\setbox\WF@box\top\bgrou \setlength\hsize{#2}%
2562   \caption@start
2563   \ifdim\hsize>\z@ \@parboxrestore \else
2564   \setbox\z@\hbox\bgrou \let\wf@caption\caption \let\caption\wf@caption
2565   \ignorespaces \fi}%

2566 {}

```

References

- [1] Frank Mittelbach and Michel Goossens:
The L^AT_EX Companion (2nd. Ed.),
Addison-Wesley, 2004.
- [2] Till Tantau:
User Guide to the Beamer Class, Version 3.07,
March 11, 2007
- [3] Markus Kohm & Jens-Uwe-Morawski:
KOMA-Script – a versatile L^AT_EX 2_ε bundle,
2007-01-09
- [4] Victor Eijkhout:
An introduction to the Dutch L^AT_EX document classes,
3 September 1989
- [5] Anselm Lingnau:
An Improved Environment for Floats,
2001/11/08
- [6] Mats Dahlgren:
Welcome to the floatflt package,
1998/06/05
- [7] Olga Lapko:
The floatrow package documentation,
2007/08/24
- [8] Sebastian Gross:
Welcome to the beta test of fltpage package!,
1998/11/13
- [9] Sebastian Rahtz & Heiko Oberdiek:
Hypertext marks in L^AT_EX,
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 \LaTeX ,
1997/09/26
- [16] Rolf Niepraschk & Hubert Gäßlein:
The sidecap package,
2003/06/06
- [17] Steven D. Cochran:
The subfigure package,
2002/07/02
- [18] Steven D. Cochran:
The subfig package,
2005/07/05
- [19] Johannes Braams and Theo Jurriens:
The supertabular environment,
2002/07/19
- [20] Donald Arseneau:
Three part tables: title, tabular environment, notes,
2003/06/13
- [21] Donald Arseneau:
WRAPFIG.STY ver 3.6,
2003/01/31
- [22] Anne Brüggemann-Klein:
Einführung in die Dokumentverarbeitung,
B.G. Teubner, Stuttgart, 1989
- [23] Heiko Oberdiek:
The refcount package,
2006/02/20