The **caption2** package[*]

Axel Sommerfeldt

`caption@sommerfeldt.net`

2004/07/16

This package is obsolete!

The **caption2** package used to be an experimental side-version of the regular **caption** package and has been superseed by the new release of the regular **caption** package version 3.0.

It is still some kind of supported, that means questions will be answered and bugs will still be fixed, but you should migrate to the new regular release **caption** v3.0 as soon as possible.

Please ignore all hints in other documents which try to tell you that the **caption2** package should be used instead of the **caption** package – these hints are outdated.

---

[*]This package has version number v2.1c, last revised 2004/05/10.

1 The Implementation

1.1 Identificaton

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption2}[2004/05/10 v2.1c Customising captions (AS)]
```

1.2 Preliminary declarations

\captionfont \captionfont and \captionlabelfont will hold the font specifications for the
\captionlabelfont caption.

```
3 \newcommand*\captionfont{}
4 \newcommand*\captionlabelfont{}
```

\captionlabeldelim \captionlabeldelim & \captionlabelsep will hold the iterim space between
\captionlabelsep caption label and text. (\captionlabeldelim will be typeset within \captionlabelfont,
\captionlabelsep not.)

```
5 \newcommand*\captionlabeldelim{}
6 \newcommand*\captionlabelsep{}
```

\captionsize The macro \captionsize is obsolete since v1.4 of the caption package, but we still
support it to provide backward compatibility.

```
7 \newcommand*\captionsize{}
```

\captionmargin Either \captionmargin (with specifies an extra margin) or \captionwidth (with
\captionwidth specifies an explicit width) can be set, therefore we need the flag \ifcaptionwidth
\ifcaptionwidth to determine with parameter we should pay attention to.

```
8 \newdimen\captionmargin
9 \newdimen\captionwidth
10 \newif\ifcaptionwidth
```

\captionindent \captionindent will be used in caption style indent and specifies the indention
after the first line.

```
11 \newdimen\captionindent
```

\ifcaptionlabel More flags. If \ifcaptionlabel is not set the caption label should be suppressed;
\ifonelinecaptions we need this flag to support the \caption* command. If \ifonelinecaptions
\ifignoreLTcapwidth is set we support the LaTeX base style 'one line captions', that means the caption
will be typeset centered if it fits to one line. If \ifignoreLTcapwidth is set we
ignore the \LTcapwidth of longtable.

```
12 \newif\ifcaptionlabel\captionlabeltrue
13 \newif\ifonelinecaptions
14 \newif\ifignoreLTcapwidth
```

\setcaptionmargin User-friendly commands to set the caption margin resp. width. Note that they
\setcaptionwidth additionally set the \ifcaptionwidth flag.

```
15 \newcommand*\setcaptionmargin{%
16   \captionwidthfalse
17   \setlength\captionmargin}
18 \newcommand*\setcaptionwidth{%
19   \captionwidthtrue
20   \setlength\captionwidth}
```

\normalcaptionparams   \normalcaptionparams resets all caption related parameters to it's normal default values. \captionfont will be set to \captionsize so setting the obsolete \captionsize will still work. Same story with \captiondelim and the obsolete \captionlabeldelim.

```
21 \newcommand*\normalcaptionparams{%
22   \let\captionsize\@empty
23   \renewcommand*\captionfont{\captionsize}%
24   \let\captionlabelfont\@empty
25   \renewcommand*\captionlabeldelim{:}%
26   \renewcommand*\captionlabelsep{\space}%
27   \setcaptionmargin\z@\setlength\captionindent\z@
28   \onelinecaptionstrue}
```

\caption@eh   Some commands will produce an error message, use this as help text.

```
29 \newcommand*\caption@eh{%
30   If you do not understand this error, please take a closer look\MessageBreak
31   at the documentation of the 'caption2' package.\MessageBreak
32   \@ehc}
```

\defcaptionstyle   These macros will define a new caption style. \newcaptionstyle and \renewcaptionstyle
\newcaptionstyle   will additionally check if the caption style already exists or not.
\renewcaptionstyle

```
33 \newcommand*\defcaptionstyle[1]{%
34   \@namedef{caption@@#1}}
35 %
36 \newcommand*\newcaptionstyle[1]{%
37   \expandafter\ifx\csname caption@@#1\endcsname\relax
38     \expandafter\defcaptionstyle
39   \else
40     \PackageError{caption2}{Caption style '#1' already defined}{\caption@eh}%
41     \expandafter\@gobbletwo
42   \fi
43   {#1}}
44 %
45 \newcommand*\renewcaptionstyle[1]{%
46   \expandafter\ifx\csname caption@@#1\endcsname\relax
47     \PackageError{caption2}{Caption style '#1' undefined}{\caption@eh}%
48     \expandafter\@gobbletwo
49   \else
50     \expandafter\defcaptionstyle
51   \fi
52   {#1}}
```

\dummycaptionstyle   This macro will also define a new caption style, but a one which is based on the actual set caption style. Therefore you can't set a caption style made with this command with \captionstyle – we check this to avoid an endless recursion.

```
53 \newcommand*\dummycaptionstyle[2]{%
54   \defcaptionstyle{#1}{%
55     \expandafter\ifx\csname caption@@\caption@style\expandafter\endcsname%
56                     \csname caption@@#1\endcsname
57       \PackageError{caption2}{You can't use the caption style '#1' directy}{%
58         The caption style '#1' is only a dummy and does not really exists.%
59         \MessageBreak You have to redefine it (with \protect\renewcaptionstyle)
60         before you can select\MessageBreak it with \protect\captionstyle.
```

```
61        \space\caption@eh}%
62     \else
63        #2\usecaptionstyle{\caption@style}%
64     \fi}}
```

\captionstyle \captionstyle sets the actual caption style. It includes a check if the given caption style is defined or not.

```
65 \newcommand*\captionstyle[1]{%
66     \expandafter\ifx\csname caption@@#1\endcsname\relax
67        \PackageError{caption2}{Undefined caption style '#1'}{\caption@eh}%
68     \else
69        \def\caption@style{#1}%
70     \fi}
```

style 'normal' The predefined caption styles 'normal', 'center', 'flushleft', 'flushright', 'centerlast',
style 'center' 'hang', 'hang+X', and 'indent'. Because they are quite similar they all are based
style 'centerlast' on the macro \caption@make.
style 'flushleft'
style 'flushright'
style 'hang'
style 'indent'

```
71 \newcaptionstyle{normal}{\caption@make{normal}}
72 \newcaptionstyle{center}{\caption@make{center}}
73 \newcaptionstyle{centerlast}{\caption@make{centerlast}}
74 \newcaptionstyle{flushleft}{\caption@make{flushleft}}
75 \newcaptionstyle{flushright}{\caption@make{flushright}}
76 \newcaptionstyle{hang}{\caption@make{hang}}
77 \newcaptionstyle{hang+center}{\caption@make{hang@center}}
78 \newcaptionstyle{hang+centerlast}{\caption@make{hang@centerlast}}
79 \newcaptionstyle{hang+flushleft}{\caption@make{hang@flushleft}}
80 \newcaptionstyle{indent}{\caption@make{indent}}
```

\caption@makecaption Our predefined caption styles. \caption@makecaption takes the style name as parameter, it does the common stuff and calls a macro (build out of the style name) to do the uncommon stuff if neccessary.

```
81 \newcommand*\caption@makecaption[1]{%
82     \usecaptionmargin
83 %
84     \ifcaptionlabel
85        \def\caption@label{%
86           {\captionlabelfont\captionlabel\captionlabeldelim}\captionlabelsep}%
87     \else
88        \let\caption@label\@empty
89     \fi
90 %
91     \captionfont
92     \onelinecaption
93        {\caption@label\captiontext}%
94        {\parbox[b]\captionlinewidth{\strut\@nameuse{caption@@@#1}\par}\par}}
95 \newcommand*\caption@make{\caption@makecaption}
```

\caption@@@normal The 'normal' caption style. Just typeset caption (label & text) as paragraph.

```
96 \newcommand*\caption@@@normal{%
97     \caption@label\captiontext}
```

\caption@@@center The 'center' caption style. Typeset the caption centered within a parbox.

```
98 \newcommand*\caption@@@center{%
99     \centering\caption@label\captiontext}%
```

`\caption@@@centerlast`	The 'centerlast' caption style. The idea how to do this was taken from Brüggemann-Klein[5], it is also mentioned in Kopka[6, p227].

```
100 \newcommand*\caption@centerlast{%
101    \advance\leftskip by 0pt plus 1fil%
102    \advance\rightskip by 0pt plus -1fil%
103    \parfillskip0pt plus 2fil\relax}
104 %
105 \newcommand*\caption@@@centerlast{%
106    \caption@centerlast\caption@label\captiontext}
```

`\caption@@@flushleft`	The 'flushleft' caption style. Typeset the caption raggedright within a parbox.

```
107 \newcommand*\caption@@@flushleft{%
108    \raggedright\caption@label\captiontext}%
```

`\caption@@@flushright`	The 'flushright' caption style. Typeset the caption raggedleft within a parbox.

```
109 \newcommand*\caption@@@flushright{%
110    \raggedleft\caption@label\captiontext}%
```

`\caption@@@hang` `\caption@hangplus`	The 'hang' caption style. This code was taken from The LaTeX Companion[4, p155] and modified.

```
111 \newcommand*\caption@@@hang{%
112    \sbox\@tempboxa{\caption@label}%
113    \hangindent\wd\@tempboxa\noindent
114    \usebox\@tempboxa\caption@hangplus\captiontext}
115 %
116 \newcommand*\caption@hangplus{}
```

`\caption@@@hang@center`	The 'hang+flushleft' caption style.

```
117 \newcommand*\caption@@@hang@center{%
118    \let\caption@hangplus\centering\caption@@@hang}
```

`\caption@@@hang@centerlast`	The 'hang+flushleft' caption style.

```
119 \newcommand*\caption@@@hang@centerlast{%
120    \let\caption@hangplus\caption@centerlast\caption@@@hang}
```

`\caption@@@hang@flushleft`	The 'hang+flushleft' caption style.

```
121 \newcommand*\caption@@@hang@flushleft{%
122    \let\caption@hangplus\raggedright\caption@@@hang}
```

`\caption@@@indent`	The 'indent' caption style. Is is quite like the 'hang' style but the indention is given as `\captionindent`.

```
123 \newcommand*\caption@@@indent{%
124    \hangindent\captionindent\noindent
125    \caption@label\captiontext}
```

1.3 Options

normal center centerlast,anne flushleft flushright hang,isu indent	These options will set the caption style. ('normal' is the default one.) The options 'anne' and 'isu' are for backward compatibility only.

```
126 \DeclareOption{normal}{\captionstyle{normal}}
127 \DeclareOption{center}{\captionstyle{center}}
128 \DeclareOption{centerlast}{\captionstyle{centerlast}}
```

```
129 \DeclareOption{flushleft}{\captionstyle{flushleft}}
130 \DeclareOption{flushright}{\captionstyle{flushright}}
131 \DeclareOption{anne}{\ExecuteOptions{centerlast}}
132 \DeclareOption{hang}{\captionstyle{hang}}
133 \DeclareOption{hang+center}{\captionstyle{hang+center}}
134 \DeclareOption{hang+centerlast}{\captionstyle{hang+centerlast}}
135 \DeclareOption{hang+flushleft}{\captionstyle{hang+flushleft}}
136 \DeclareOption{isu}{\ExecuteOptions{hang}}
137 \DeclareOption{indent}{\captionstyle{indent}}
```

scriptsize These options will set the caption size. We use `\g@addto@macro` so more that one
footnotesize option can be set.

small
```
138 \DeclareOption{scriptsize}{\g@addto@macro\captionsize\scriptsize}
```
normalsize
```
139 \DeclareOption{footnotesize}{\g@addto@macro\captionsize\footnotesize}
```
large,Large
```
140 \DeclareOption{small}{\g@addto@macro\captionsize\small}
141 \DeclareOption{normalsize}{\g@addto@macro\captionsize\normalsize}
142 \DeclareOption{large}{\g@addto@macro\captionsize\large}
143 \DeclareOption{Large}{\g@addto@macro\captionsize\Large}
```

up,it,sl,sc These options will set the caption label.

md,bf
```
144 \DeclareOption{up}{\g@addto@macro\captionlabelfont\upshape}
```
rm,sf,tt
```
145 \DeclareOption{it}{\g@addto@macro\captionlabelfont\itshape}
146 \DeclareOption{sl}{\g@addto@macro\captionlabelfont\slshape}
147 \DeclareOption{sc}{\g@addto@macro\captionlabelfont\scshape}
148 \DeclareOption{md}{\g@addto@macro\captionlabelfont\mdseries}
149 \DeclareOption{bf}{\g@addto@macro\captionlabelfont\bfseries}
150 \DeclareOption{rm}{\g@addto@macro\captionlabelfont\rmfamily}
151 \DeclareOption{sf}{\g@addto@macro\captionlabelfont\sffamily}
152 \DeclareOption{tt}{\g@addto@macro\captionlabelfont\ttfamily}
```

oneline These options will set the 'oneline' flag. ('oneline' is the default.)
nooneline
```
153 \DeclareOption{oneline}{\onelinecaptionstrue}
154 \DeclareOption{nooneline}{\onelinecaptionsfalse}
```

\caption@package A helper macro, a value of 1 within parameter #2 will activate the support of the
package given in parameter #1, a value of 0 will deactivate it.
```
155 \newcommand*\caption@package[1]{\@namedef{caption@pkt@#1}}
```

float These options will enable or suppress the support of the packages float, longtable,
longtable and subfigure.
subfigure
```
156 \DeclareOption{float}{\caption@twozerofalse\caption@package{float}{1}}
157 \DeclareOption{longtable}{\caption@twozerofalse\caption@package{longtable}{1}}
158 \DeclareOption{subfigure}{\caption@twozerofalse\caption@package{subfigure}{1}}
```

none These options will enable or suppress the support of all the above packages.
all
```
159 \DeclareOption{none}{\caption@twozerofalse
160   \caption@package{float}{0}\caption@package{longtable}{0}%
161   \caption@package{subfigure}{0}}
162 \DeclareOption{all}{\ExecuteOptions{float,longtable,subfigure}}
```

ruled The option 'ruled' introduced in caption v1.2 is obsolete now, but we will still
boxed support it. The option 'boxed' was introduced in version 2.0 and is obsolete now,
too.

```
163 \newif\ifcaption@ruled
164 \DeclareOption{ruled}{\caption@ruledtrue}
165 \DeclareOption{boxed}{}
```

ignoreLTcapwidth This option will make the caption code ignore the setting of `\LTcapwidth` and use the setting of `\setcaptionmargin` or `\setcaptionwidth` instead.

```
166 \DeclareOption{ignoreLTcapwidth}{\ignoreLTcapwidthtrue}
```

debug This option will put additional debug information in the log file.

```
167 \DeclareOption{debug}{\caption@debugtrue}
```

That's it! Now set the default values and start processing the options. (If `\caption@twozero` is set to true (default) we will emulate the package load algorithm of caption v2.0: If the package is already loaded patch it, otherwise do nothing.)

```
168 \newif\ifcaption@debug
169 \newif\ifcaption@twozero
170 \normalcaptionparams
171 \ExecuteOptions{none,normal}
172 \caption@twozerotrue
173 \ProcessOptions*
174 \ifcaption@twozero
175   \PackageInfo{caption2}{Running in caption2 v2.0 compatibility mode}
176 \fi
```

1.4   More declarations

\captionof
\captionof*
`\captionof` resp. `\captionof*` will just set `\@captype` and do the normal `\caption` resp. `\caption*`, so we can also typeset captions outside floating environments.

```
177 \def\captionof{\@ifstar{\caption@of{\caption*}}{\caption@of\caption}}
178 \newcommand*\caption@of[2]{\def\@captype{#2}#1}
```

\abovecaptionskip
\belowcaptionskip
Not all document classes define `\abovecaptionskip` and `\belowcaptionskip` (like ucthesis), so we do it here if not already done.

```
179 \@ifundefined{abovecaptionskip}{%
180   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
181 \@ifundefined{belowcaptionskip}{%
182   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}
```

\captionlinewidth
\captionlabel
\captiontext
These values are only set and used within the caption code itself. `\captionlinewidth` will be set to the given vertical space for the caption, normally this is `\linewidth`. (This value was called `\realcaptionwidth` within caption2 2.0, so we will offer this, too.)
`\captionlabel` and `\captiontext` will be set to the caption label resp. the caption text. (Because `\captionlabel` and `\captiontext` will be locally defined with `\def` we do not need to define them here.)

```
183 \newdimen\captionlinewidth
184 \newdimen\realcaptionwidth
```

\usecaptionmargin A helper macro for caption style authors: It calculates \leftskip and \rightskip out of \captionlinewidth and \captionmargin resp. \captionwidth. Also \captionlinewidth will be corrected to the appropriate value.

```
185 \newcommand*\usecaptionmargin{%
186   \ifcaptionwidth
187     \leftskip\captionlinewidth
188     \advance\leftskip by -\captionwidth
189     \divide\leftskip by 2
190     \rightskip\leftskip
191     \captionlinewidth\captionwidth
192   \else
193     \leftskip\captionmargin
194     \rightskip\captionmargin
195     \advance\captionlinewidth by -2\captionmargin
196   \fi
197   \realcaptionwidth\captionlinewidth}
```

\onelinecaption This macro definition helps setting captions the LaTeX base classes way: If \ifonelinecaptions is set and the 1st argument fits within \captionlinewidth, we typeset it centered – otherway we typeset the 2nd argument. (We use the savebox \@tempboxa as helper for this.)

```
198 \newcommand\onelinecaption[1]{%
199   \let\next\@firstofone
200   \ifonelinecaptions
201     \sbox\@tempboxa{#1}%
202     \ifdim\wd\@tempboxa >\captionlinewidth
203     \else
204       \def\next{{\centering\usebox{\@tempboxa}\par}\@gobble}%
205     \fi
206   \fi\next}
```

\usecaptionstyle First we check if we are inside a caption – if \captiontext is undefined we are not. If we are we call the appropriate caption definition.

```
207 \newcommand*\usecaptionstyle[1]{%
208   \@ifundefined{captiontext}{%
209     \PackageError{caption2}{You can't use \protect#1
210       in normal text}{The usage of \protect#1 is only
211       allowed inside code declared with\MessageBreak \protect\defcaptionstyle,
212       \protect\newcaptionstyle \space or \protect\renewcaptionstyle.
213       \space\caption@eh}
214   }{%
215     \@ifundefined{caption@@#1}%
216       {\PackageError{caption2}{Caption style '#1' undefined}{\caption@eh}}%
217       {\@nameuse{caption@@#1}}%
218   }}
```

\@makecaption This is the heart of the caption2 package – the redefinition of the core caption code. It was taken from the LaTeX 2ε standard classes and modified. It's very easy – apart from using \abovecaptionskip and \belowcaptionskip we just set \captionlinewidth, \captionlabel and \captiontext to its appropriate values and using the code of the actual caption style via \usecaptionstyle.

```
219 \renewcommand\@makecaption[2]{%
```

```
220    \vskip\abovecaptionskip
221    \captionlinewidth\hsize
222    \realcaptionwidth\hsize
223    \def\captionlabel{#1}%
224    \def\captiontext{#2}%
225    \usecaptionstyle{\caption@style}%
226    \vskip\belowcaptionskip}
```

1.5   Support of other packages

\caption@package   This macro will execute the code needed to support the package named within
argument #1.   The parameter #2 is the command which shows if the package
is loaded – it is defined, it is already loaded, otherwise not.   The parameter #3
contains code which will be executed if no support is required – this is for cleanup
purposes.   The final parameter #4 contains the code itself.

```
227 \renewcommand*\caption@package[3]{%
228    \if1\@nameuse{caption@pkt@#1}%
229      \@ifundefined{#2}%
230        {\let\next\AtBeginDocument}%
231        {\let\next\@firstofone}%
232    \else\ifcaption@twozero
233      \@ifundefined{#2}%
234        {#3\let\next\@gobble}%
235        {\let\next\@firstofone}%
236    \else
237      #3\let\next\@gobble
238    \fi\fi
239    \expandafter\let\csname caption@pkt@#1\endcsname\undefined
240    \ifcaption@debug
241      \ifx\next\@gobble\PackageInfo{caption2}{#1 => gobble}%
242      \else\ifx\next\@firstofone\PackageInfo{caption2}{#1 => firstofone}%
243      \else\ifx\next\AtBeginDocument\PackageInfo{caption2}{#1 => AtBeginDocument}%
244      \fi\fi\fi
245    \fi
246    \next}
```

1.5.1   Support of the float package

```
247 \caption@package{float}{floatc@plain}{}{%
248    \ifx\floatc@plain\relax
249      \PackageWarning{caption2}{%
250        Option 'float' was set but there is no float package loaded}
251    \else
252      \PackageInfo{caption2}{float package v1.2 (or newer) detected}
```

\caption@floatc   First we define a helper macro to typeset the caption via \usecaptionstyle, the
1st parameter is the caption style name, the 2nd and 3rd are the caption label
and text.
caption2 has the goal not to modify the output just by loading it (without options),
therefore we have to be tricky here to support \@fs@cfont which is in fact the
same as our \captionlabelfont. So we test if a \captionlabelfont has been
set by the user – if not \@fs@cfont will be used, otherwise \captionlabelfont.

```
253      \newcommand\caption@floatc[3]{%
```

9

```
254        \ifx\captionlabelfont\@empty
255          \let\captionlabelfont\@fs@cfont
256        \fi
257        \captionlinewidth\hsize
258        \realcaptionwidth\hsize
259        \def\captionlabel{#2}%
260        \def\captiontext{#3}%
261        \usecaptionstyle{#1}}
```

\floatc@plain Now we can redefine the caption code of the float package. Here we redefine `\floatc@plain` to use our caption code, so `plain` and `boxed` float types will use the actual caption style set by the user.

```
262        \renewcommand*\floatc@plain{\caption@floatc{\caption@style}}
```

\floatc@ruled The support of the `ruled` float type is a little more complex. First we define a caption style 'ruled' so the end-user can change this caption style afterwards. If the (obsolete) option 'ruled' is set, we define it in a caption v1.x compatible way, otherwise we define it in a float compatible way.

Then we redefine `\floatc@ruled` so the caption style 'ruled' will be used.

```
263        \ifcaption@ruled
264          \dummycaptionstyle{ruled}{\onelinecaptionsfalse\setcaptionmargin{\z@}}%
265        \else
266          \newcaptionstyle{ruled}{%
267            \ifcaptionlabel
268              {\@fs@cfont\captionlabel}\space%
269            \fi\captiontext\par}%
270        \fi
271 %
272        \renewcommand*\floatc@ruled{\caption@floatc{ruled}}}
```

\caption@of Typesetting captions outside floats is not so easy with redefined floats, because

- The caption code of the float package needs not only `\@captype` defined, but `\@fs@capt` (the command which will typeset the caption itself) either.

- The caption is only saved within a `\vbox`, so the float package can typeset the caption later at it's float style specific place (that means at top or at the bottom of the float).

Here is the new code: First we check if it's a restyled float by checking if `\fst@<floattype>` is defined. If yes, we use this command (it will define `\@fs@capt`). Then we execute `\@float@setevery`, if it exists (that means we are dealing with the float package 1.3 or newer here). Now comes the basic trick: We redefine the caption typesetting command `\@fs@capt`, so it will close the `\vbox`, typeset the caption outside the vbox and finally start the group again so the original `\@fs@capt` is happy with closing the group.

```
273        \renewcommand*\caption@of[2]{\def\@captype{#2}%
274          \@ifundefined{fst@#2}{}{%
275            \@nameuse{fst@#2}%
276            \@ifundefined{@float@setevery}{}{\@float@setevery{#2}}%
277            \let\caption@fs@capt\@fs@capt
278            \let\@fs@capt\caption@of@float}%
279          #1}
```

```
280 %
281     \newcommand\caption@of@float[2]{\egroup
282       \vskip\abovecaptionskip
283       \normalsize\caption@fs@capt{#1}{#2}%
284       \vskip\belowcaptionskip
285       \bgroup}%

286   \fi}
```

1.5.2 Support of the longtable package

```
287 \caption@package{longtable}{LT@makecaption}{}{%
288   \ifx\LT@makecaption\relax
289     \PackageWarning{caption2}{%
290       Option 'longtable' was set but there is no longtable package loaded}
291   \else
292     \PackageInfo{caption2}{longtable package v3.15 (or newer) detected}
```

\LT@makecaption   David Carlisle was so kind to introduce a macro called \LT@makecaption in version 3.15 of the longtable package which typeset the caption and can be easily redefined. This is the original definition:

```
    \def\LT@makecaption#1#2#3{%
      \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\LTcapwidth{%
        ⟨typeset #1{#2: }#3 as caption⟩
        \endgraf\vskip\baselineskip}%
      \hss}}}
```

So we do here: First we define a new (dummy) caption style 'longtable', than we redefine \LT@makecaption so this style will be used. (Remember: #1 is \@gobble in star form of \caption, and \@firstofone otherwise.)

```
293     \dummycaptionstyle{longtable}{}
294 %
295     \renewcommand\LT@makecaption[3]{%
296       \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\hsize{%
297         \ifignoreLTcapwidth
298         \else
299           \setcaptionwidth\LTcapwidth
300         \fi
301         \captionlinewidth\hsize
302         \realcaptionwidth\hsize
303         \captionlabelfalse#1\captionlabeltrue
304         \def\captionlabel{#2}%
305         \def\captiontext{#3}%
306         \usecaptionstyle{longtable}%
307         \endgraf\vskip\baselineskip}%
308       \hss}}}

309   \fi}
```

1.5.3 Support of the subfigure package

Some of the following code will not work within \if, because of the (yet) undefined \if*xxx*s. So we simply define the critical code within the helper commands \setsubcapstyle and \caption@makesubcaption already here.

\setsubcapstyle   This sets the subcaptionstyle to a appropriate value.

If `\ifsubcapraggedright` is undefined (it was introduced into v2.1 of the subfigure package) we define it first.

```
310 \newcommand*\setsubcapstyle{%
311   \@ifundefined{subcapraggedrightfalse}{%
312     \newif\ifsubcapraggedright}{}%
313   \ifsubcaphang
314     \ifsubcapcenter
315       \subcapstyle{hang+center}%
316     \else\ifsubcapcenterlast
317       \subcapstyle{hang+centerlast}%
318     \else\ifsubcapraggedright
319       \subcapstyle{hang+flushleft}%
320     \else
321       \subcapstyle{hang}%
322     \fi\fi\fi
323   \else\ifsubcapcenter
324     \subcapstyle{center}%
325   \else\ifsubcapcenterlast
326     \subcapstyle{centerlast}%
327   \else\ifsubcapraggedright
328     \subcapstyle{flushleft}%
329   \else
330     \subcapstyle{normal}%
331   \fi\fi\fi\fi}
```

\caption@makesubcaption   This will typeset the subcaption. We just set all our `\caption`*xxx* values to the values of `\subcap`*xxx* and typeset the caption like subfigure within a `\hbox`, but with the help of `\usecaptionstyle`.

But this is not as easy as it seems. We typeset the caption like this:

```
\captionfont
  {\capionlabelfont\captionlabel\captionlabeldelim}%
\captionlabelsep\captiontext
```

Within subfigure 2.0 the caption will be set quite similar to:

```
\subcapsize
  {\subcaplabelfont\captionlabel}%
\space\captiontext
```

But within subfigure 2.1 this has changed to:

```
\subcapsize
  {\subcaplabelfont\captionlabel}%
\hskip\subfiglabelskip
  {\subcapfont\captiontext}}
```

So we have to be tricky here: We set `\captionlabelfont` to `\normalfont` plus `\subcapsize` & `\subcaplabelfont`, so the font setting in `\captionfont` will not affect the caption label in subfigure captions.

Note that `\hfil` has changed to `\hss` from subfigure 2.0 to 2.1, so we use `\caption@subfig@hss` instead. (We will define this later on.)

```
332 \newcommand\caption@makesubcaption[2]{%
```

```
333    \renewcommand*\captionfont{\subcapsize\subcapfont}%
334    \renewcommand*\captionlabelfont{\normalfont\subcapsize\subcaplabelfont}%
335    \let\captionlabeldelim\subcaplabeldelim
336    \let\captionlabelsep\subcaplabelsep
337    \ifsubfigcapwidth\captionwidthtrue\else\captionwidthfalse\fi
338    \setlength\captionmargin\subfigcapmargin
339    \setlength\captionwidth\subfigcapwidth
340    \captionindent\subcapindent
341    \ifsubcapnooneline\onelinecaptionsfalse\else\onelinecaptionstrue\fi
342    \hbox to\@tempdima{%
343      \caption@subfig@hss\parbox[t]{\@tempdima}{%
344        \captionlinewidth\@tempdima
345        \realcaptionwidth\@tempdima
346        \captionlabeltrue
347        \def\captionlabel{#1}%
348        \def\captiontext{\ignorespaces #2}%
349        \usecaptionstyle{\caption@substyle}}%
350      \caption@subfig@hss}}
```

If the subfigure support is not needed, we throw the helper macros in the garbage can.

```
351 \caption@package{subfigure}{@makesubfigurecaption}{%
352    \let\setsubcapstyle\undefined
353    \let\caption@makesubcaption\undefined}{%
354    \ifx\@makesubfigurecaption\relax
355      \PackageWarning{caption2}{%
356        Option 'subfigure' was set but there is no subfigure package loaded}
357      \let\setsubcapstyle\undefined
358      \let\caption@makesubcaption\undefined
359    \else
```

Some stuff has changed from version 2.0 to 2.1 of the subfigure package, so we make a branch here. If \subcapfont is undefined we assume v2.0, otherwise we assume v2.1 or newer.

```
360      \ifx\subcapfont\undefined
361        \PackageInfo{caption2}{subfigure package v2.0 detected}
```

\subcapfont   We define \subcapfont here so we can use it later in common code for subfigure v2.0 and v2.1 (or newer).

```
362        \let\subcapfont\@empty
```

\subfigcapwidth   Analogous to \captionwidth, \setcaptionmargin, and \setcaptionwidth we
\setsubcapmargin   define \subfigcapwidth, \setsubcapmargin, and \setsubcapwidth.
\setsubcapwidth   Note: \subfigcapmargin is a command in v2.0 of subfigure. So we make
               \subfigcapwidth a command, too.

```
363        \newcommand*\subfigcapwidth{\z@}
364        \newcommand*\setsubcapmargin{%
365          \subfigcapwidthfalse
366          \renewcommand*\subfigcapmargin}
367        \newcommand*\setsubcapwidth{%
368          \subfigcapwidthtrue
369          \renewcommand*\subfigcapwidth}
```

\subcaplabelsep Analogous to \captionlabelsep we define \subcaplabelsep.

370   \newcommand*\subcaplabelsep{\space}

\caption@subfig@hss This will be uses within the caption code itself.

371   \let\caption@subfig@hss\hfil

372   \else
373   \PackageInfo{caption2}{subfigure package v2.1 (or newer) detected}

\subfigcapwidth Analogous to \captionwidth, \setcaptionmargin, and \setcaptionwidth we
\setsubcapmargin define \subfigcapwidth, \setsubcapmargin, and \setsubcapwidth.
\setsubcapwidth Note: \subfigcapmargin is a length in v2.1 of subfigure. So we make
\subfigcapwidth a length, too.

374   \newdimen\subfigcapwidth
375   \newcommand*\setsubcapmargin{%
376   \subfigcapwidthfalse
377   \setlength\subfigcapmargin}
378   \newcommand*\setsubcapwidth{%
379   \subfigcapwidthtrue
380   \setlength\subfigcapwidth}

\subcaplabelsep Analogous to \captionlabelsep we define \subcaplabelsep.

381   \newcommand*\subcaplabelsep{\hskip\subfiglabelskip}

\caption@subfig@hss This will be uses within the caption code itself.

382   \let\caption@subfig@hss\hss

383   \fi

Here starts the common code for subfigure v2.0 and v2.1.

\ifsubfigcapwidth Analogous to \ifcaptionwidth, \captionindent & \captionlabeldelim we de-
\subcapindent fine \ifsubfigcapwidth, \subcapindent & \subcaplabeldelim
\subcaplabeldelim 384   \newif\ifsubfigcapwidth
385   \newdimen\subcapindent
386   \newcommand*\subcaplabeldelim{}

\subcapstyle Analogous to \captionstyle we define \subcapstyle and set it (via \setsubcapstyle)
to a appropriate value.

387   \newcommand*\subcapstyle[1]{%
388   \expandafter\ifx\csname caption@@#1\endcsname\relax
389   \PackageError{caption2}{Undefined caption style '#1'}{\caption@eh}%
390   \else
391   \def\caption@substyle{#1}%
392   \fi}
393   \setsubcapstyle

\@thesubfigure The subfigure package makes use of \subcaplabelfont and \subfiglabelskip
\@thesubtable within its \@thesub*xxx* macros. This is totally in contrast to the way the caption2
package handle these settings! So we redefine the \@thesub*xxx* to be just the plain
label and nothing else.

394   \renewcommand*\@thesubfigure{\thesubfigure}
395   \renewcommand*\@thesubtable{\thesubtable}

`\@makesubfigurecaption`
`\@makesubtablecaption`

Now we are ready to redefine `\@makesubfigurecaption`.

```
396        \let\@makesubfigurecaption\caption@makesubcaption
397        \let\@makesubtablecaption\caption@makesubcaption
```

```
398    \fi}
```

That's all folks!

```
399 \let\caption@package\undefined
```

References

[1] Anselm Lingnau: *An Improved Environment for Floats*, 2001/11/08

[2] David Carlisle: *The longtable package*, 2000/10/22

[3] Steven Douglas Cochran: *The subfigure package*, 2002/02/14

[4] Michel Goossens, Frank Mittelbach and Alexander Samarin: *The LaTeX Companion*, Addison-Wesley, Reading, Massachusetts, 1994.

[5] Anne Brüggemann-Klein: *Einführung in die Dokumentverarbeitung*, B.G. Teubner, Stuttgart, 1989

[6] Helmut Kopka: *LaTeX– Erweiterungsmöglichkeiten*, 3. überarbeitete Auflage, Addison-Wesley, Bonn, 1991