The caption2 package*

Axel Sommerfeldt

`caption@sommerfee.de`

2005/11/02

This package is obsolete!

The caption2 package used to be an experimental side-version of the regular caption package. It was made public as beta test version without documentation in 1995 because of the demand for new features and adaptations to other packages like the longtable and subfigure package.

Even in the next years I found no time to reintegrate some of the well-tried features into the regular caption package. So I decided to release a version 2.1 of the caption2 package in 2002 instead, which included some minor bug fixes and adaptations to the new version 2.1 of the subfigure package. Furthermore I started to write a documentation for this package, but unfortunately did not get very far with this...

In 2003 I finally found some (more) time, so a new regular release 3.0 of the caption package could be build with massive help from Frank Mittelbach[5] and Steven Cochran[4]. It was released in December 2003 and superseded the neglected caption2 package.

(In parallel, Steven Cochran released the subfig package which superseded the subfigure package.)

So please don't use this package for new documents. It's old, it's obsolete and it starts to begin smell bad! Please ignore all hints in books or other documents which try to tell you that the caption2 package should be used instead of the caption package – these hints are outdated since December 2003.

How to migrate to the regular caption package?

Usually replacing caption2 by caption is sufficient because the caption package emulates most of the options and commands of the caption2 package. If you get some errors or wired results afterwards, please take a closer look at the caption package documentation which will hopefully help you clearing these problems. If all this should fail you can write me an e-mail asking for help.

---

*This package has version number v2.1d, last revised 2005/10/03.

What will happen to this package?

The caption2 package is still some kind of supported, that means it will be part of future releases and bugs will still be fixed so existing documents using this package will still compile. But it will *not* be enhanced in the future or adapted to future versions of foreign packages.

This means migrating to the actual caption package should not be necessary for old documents.

1 The Implementation

1.1 Identificaton

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption2}[2005/10/03 v2.1d Customising captions (AS)]
3 \PackageWarning{caption2}{%
4   THIS PACKAGE IS OBSOLETE, PLEASE USE caption INSTEAD}
```

1.2 Preliminary declarations

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

```
5 \newcommand*\captionfont{}
6 \newcommand*\captionlabelfont{}
```

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

```
7 \newcommand*\captionlabeldelim{}
8 \newcommand*\captionlabelsep{}
```

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

```
9 \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.

```
10 \newdimen\captionmargin
11 \newdimen\captionwidth
12 \newif\ifcaptionwidth
```

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

```
13 \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.

```
14 \newif\ifcaptionlabel\captionlabeltrue
15 \newif\ifonelinecaptions
16 \newif\ifignoreLTcapwidth
```

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

```
17 \newcommand*\setcaptionmargin{%
18   \captionwidthfalse
19   \setlength\captionmargin}
20 \newcommand*\setcaptionwidth{%
21   \captionwidthtrue
22   \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.

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

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

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

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

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

```
52      \expandafter\defcaptionstyle
53    \fi
54    {#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.

```
55 \newcommand*\dummycaptionstyle[2]{%
56   \defcaptionstyle{#1}{%
57     \expandafter\ifx\csname caption@@\caption@style\expandafter\endcsname%
58                   \csname caption@@#1\endcsname
59       \PackageError{caption2}{You can't use the caption style '#1' directy}{%
60         The caption style '#1' is only a dummy and does not really exists.%
61         \MessageBreak You have to redefine it (with \protect\renewcaptionstyle)
62         before you can select\MessageBreak it with \protect\captionstyle.
63         \space\caption@eh}%
64     \else
65       #2\usecaptionstyle\caption@style
66     \fi}}
```

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

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

style 'normal'
style 'center'
style 'centerlast'
style 'flushleft'
style 'flushright'
style 'hang'
style 'indent'

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

```
73 \newcaptionstyle{normal}{\caption@make{normal}}
74 \newcaptionstyle{center}{\caption@make{center}}
75 \newcaptionstyle{centerlast}{\caption@make{centerlast}}
76 \newcaptionstyle{flushleft}{\caption@make{flushleft}}
77 \newcaptionstyle{flushright}{\caption@make{flushright}}
78 \newcaptionstyle{hang}{\caption@make{hang}}
79 \newcaptionstyle{hang+center}{\caption@make{hang@center}}
80 \newcaptionstyle{hang+centerlast}{\caption@make{hang@centerlast}}
81 \newcaptionstyle{hang+flushleft}{\caption@make{hang@flushleft}}
82 \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.

5

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

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

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

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

```
100 \newcommand*\caption@@@center{%
101   \centering\caption@label\captiontext}%
```

\caption@@@centerlast   The 'centerlast' caption style. The idea how to do this was taken from Brüggemann-Klein[6], it is also mentioned in Kopka[7, p227].

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

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

```
109 \newcommand*\caption@@@flushleft{%
110   \raggedright\caption@label\captiontext}%
```

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

```
111 \newcommand*\caption@@@flushright{%
112   \raggedleft\caption@label\captiontext}%
```

\caption@@@hang
\caption@hangplus   The 'hang' caption style. This code was taken from The LATEX Companion[5, p155] and modified.

```
113 \newcommand*\caption@@@hang{%
114   \sbox\@tempboxa{\caption@label}%
```

```
115    \hangindent\wd\@tempboxa\noindent
116    \usebox\@tempboxa\caption@hangplus\captiontext}
117 %
118 \newcommand*\caption@hangplus{}
```

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

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

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

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

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

```
123 \newcommand*\caption@@@hang@flushleft{%
124    \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.

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

1.3 Options

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

```
128 \DeclareOption{normal}{\captionstyle{normal}}
129 \DeclareOption{center}{\captionstyle{center}}
130 \DeclareOption{centerlast}{\captionstyle{centerlast}}
131 \DeclareOption{flushleft}{\captionstyle{flushleft}}
132 \DeclareOption{flushright}{\captionstyle{flushright}}
133 \DeclareOption{anne}{\ExecuteOptions{centerlast}}
134 \DeclareOption{hang}{\captionstyle{hang}}
135 \DeclareOption{hang+center}{\captionstyle{hang+center}}
136 \DeclareOption{hang+centerlast}{\captionstyle{hang+centerlast}}
137 \DeclareOption{hang+flushleft}{\captionstyle{hang+flushleft}}
138 \DeclareOption{isu}{\ExecuteOptions{hang}}
139 \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
normalsize
large,Large

```
140 \DeclareOption{scriptsize}{\g@addto@macro\captionsize\scriptsize}
141 \DeclareOption{footnotesize}{\g@addto@macro\captionsize\footnotesize}
142 \DeclareOption{small}{\g@addto@macro\captionsize\small}
```

7

```
143 \DeclareOption{normalsize}{\g@addto@macro\captionsize\normalsize}
144 \DeclareOption{large}{\g@addto@macro\captionsize\large}
145 \DeclareOption{Large}{\g@addto@macro\captionsize\Large}
```

up,it,sl,sc
md,bf
rm,sf,tt

These options will set the caption label.

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

oneline
nooneline

These options will set the 'oneline' flag. ('oneline' is the default.)

```
155 \DeclareOption{oneline}{\onelinecaptionstrue}
156 \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.

```
157 \newcommand*\caption@package[1]{\@namedef{caption@pkt@#1}}
```

float
longtable
subfigure

These options will enable or suppress the support of the packages float, longtable, and subfigure.

```
158 \DeclareOption{float}{\caption@twozerofalse\caption@package{float}{1}}
159 \DeclareOption{longtable}{\caption@twozerofalse\caption@package{longtable}{1}}
160 \DeclareOption{subfigure}{\caption@twozerofalse\caption@package{subfigure}{1}}
```

none
all

These options will enable or suppress the support of all the above packages.

```
161 \DeclareOption{none}{\caption@twozerofalse
162   \caption@package{float}{0}\caption@package{longtable}{0}%
163   \caption@package{subfigure}{0}}
164 \DeclareOption{all}{\ExecuteOptions{float,longtable,subfigure}}
```

ruled
boxed

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

```
165 \newif\ifcaption@ruled
166 \DeclareOption{ruled}{\caption@ruledtrue}
167 \DeclareOption{boxed}{}
```

ignoreLTcapwidth

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

```
168 \DeclareOption{ignoreLTcapwidth}{\ignoreLTcapwidthtrue}
```

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

```
169 \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.)

```
170 \newif\ifcaption@debug
171 \newif\ifcaption@twozero
172 \normalcaptionparams
173 \ExecuteOptions{none,normal}
174 \caption@twozerotrue
175 \ProcessOptions*
176 \ifcaption@twozero
177   \PackageInfo{caption2}{Running in caption2 v2.0 compatibility mode}
178 \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.

```
179 \def\captionof{\@ifstar{\caption@of{\caption*}}{\caption@of\caption}}
180 \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.

```
181 \@ifundefined{abovecaptionskip}{%
182   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
183 \@ifundefined{belowcaptionskip}{%
184   \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.)

```
185 \newdimen\captionlinewidth
186 \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.

9

```
187 \newcommand*\usecaptionmargin{%
188   \ifcaptionwidth
189     \leftskip\captionlinewidth
190     \advance\leftskip by -\captionwidth
191     \divide\leftskip by 2
192     \rightskip\leftskip
193     \captionlinewidth\captionwidth
194   \else
195     \leftskip\captionmargin
196     \rightskip\captionmargin
197     \advance\captionlinewidth by -2\captionmargin
198   \fi
199   \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.)

```
200 \newcommand\onelinecaption[1]{%
201   \let\next\@firstofone
202   \ifonelinecaptions
203     \sbox\@tempboxa{#1}%
204     \ifdim\wd\@tempboxa >\captionlinewidth
205     \else
206       \def\next{{\centering\usebox\@tempboxa\par}\@gobble}%
207     \fi
208   \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.

```
209 \newcommand*\usecaptionstyle[1]{%
210   \@ifundefined{captiontext}{%
211     \PackageError{caption2}{You can't use \protect#1
212       in normal text}{The usage of \protect#1 is only
213       allowed inside code declared with\MessageBreak \protect\defcaptionstyle,
214       \protect\newcaptionstyle \space or \protect\renewcaptionstyle.
215       \space\caption@eh}
216   }{%
217     \@ifundefined{caption@@#1}%
218       {\PackageError{caption2}{Caption style '#1' undefined}{\caption@eh}}%
219       {\@nameuse{caption@@#1}}%
220   }}
```

\@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.

```
221 \renewcommand\@makecaption[2]{%
222   \vskip\abovecaptionskip
223   \captionlinewidth\hsize
224   \realcaptionwidth\hsize
225   \def\captionlabel{#1}%
226   \def\captiontext{#2}%
227   \usecaptionstyle\caption@style
228   \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 de-
fined, 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.

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

1.5.1   Support of the float package

```
249 \caption@package{float}{floatc@plain}{}{%
250   \ifx\floatc@plain\relax
251     \PackageWarning{caption2}{%
252       Option 'float' was set but there is no float package loaded}
253   \else
254     \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.

11

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`.

```
255    \newcommand\caption@floatc[3]{%
256      \ifx\captionlabelfont\@empty
257        \let\captionlabelfont\@fs@cfont
258      \fi
259      \captionlinewidth\hsize
260      \realcaptionwidth\hsize
261      \def\captionlabel{#2}%
262      \def\captiontext{#3}%
263      \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.

```
264    \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.

```
265    \ifcaption@ruled
266      \dummycaptionstyle{ruled}{\onelinecaptionsfalse\setcaptionmargin{\z@}}%
267    \else
268      \newcaptionstyle{ruled}{%
269        \ifcaptionlabel
270          {\@fs@cfont\captionlabel}\space%
271        \fi\captiontext\par}%
272    \fi
273 %
274    \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.

```
275    \renewcommand*\caption@of[2]{\def\@captype{#2}%
276      \@ifundefined{fst@#2}{}{%
277        \@nameuse{fst@#2}%
278        \@ifundefined{@float@setevery}{}{\@float@setevery{#2}}%
279        \let\caption@fs@capt\@fs@capt
280        \let\@fs@capt\caption@of@float}%
281      #1}
282    \newcommand\caption@of@float[2]{\egroup
283      \vskip\abovecaptionskip
284      \normalsize\caption@fs@capt{#1}{#2}%
285      \vskip\belowcaptionskip
286      \bgroup}%
287  \fi}
```

1.5.2 Support of the longtable package

```
288 \caption@package{longtable}{LT@makecaption}{}{%
289   \ifx\LT@makecaption\relax
290     \PackageWarning{caption2}{%
291       Option 'longtable' was set but there is no longtable package loaded}
292   \else
293     \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.)

```
294      \dummycaptionstyle{longtable}{}
295 %
296    \renewcommand\LT@makecaption[3]{%
297      \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\hsize{%
298        \ifignoreLTcapwidth
299        \else
```

```
300          \setcaptionwidth\LTcapwidth
301        \fi
302        \captionlinewidth\hsize
303        \realcaptionwidth\hsize
304        \captionlabelfalse#1\captionlabeltrue
305        \def\captionlabel{#2}%
306        \def\captiontext{#3}%
307        \usecaptionstyle{longtable}%
308        \endgraf\vskip\baselineskip}%
309      \hss}}}

310   \fi}
```

1.5.3 Support of the subfigure package

Some of the following code will not work within \if, because of the (yet) unde-
fined \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.

```
311 \newcommand*\setsubcapstyle{%
312   \@ifundefined{subcapraggedrightfalse}{%
313     \newif\ifsubcapraggedright}{}%
314   \ifsubcaphang
315     \ifsubcapcenter
316       \subcapstyle{hang+center}%
317     \else\ifsubcapcenterlast
318       \subcapstyle{hang+centerlast}%
319     \else\ifsubcapraggedright
320       \subcapstyle{hang+flushleft}%
321     \else
322       \subcapstyle{hang}%
323     \fi\fi\fi
324   \else\ifsubcapcenter
325     \subcapstyle{center}%
326   \else\ifsubcapcenterlast
327     \subcapstyle{centerlast}%
328   \else\ifsubcapraggedright
329     \subcapstyle{flushleft}%
330   \else
331     \subcapstyle{normal}%
332   \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.)

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

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

```
352 \caption@package{subfigure}{@makesubfigurecaption}{%
353   \let\setsubcapstyle\undefined
```

```
354    \let\caption@makesubcaption\undefined}{%
355    \ifx\@makesubfigurecaption\relax
356      \PackageWarning{caption2}{%
357        Option 'subfigure' was set but there is no subfigure package loaded}
358      \let\setsubcapstyle\undefined
359      \let\caption@makesubcaption\undefined
360    \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.

```
361      \ifx\subcapfont\undefined
362        \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).

```
363        \let\subcapfont\@empty
```

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

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

\subcaplabelsep   Analogous to \captionlabelsep we define \subcaplabelsep.

```
371        \newcommand*\subcaplabelsep{\space}
```

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

```
372        \let\caption@subfig@hss\hfil
```

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

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

```
375        \newdimen\subfigcapwidth
376        \newcommand*\setsubcapmargin{%
```

16

```
377        \subfigcapwidthfalse
378        \setlength\subfigcapmargin}
379      \newcommand*\setsubcapwidth{%
380        \subfigcapwidthtrue
381        \setlength\subfigcapwidth}
```

\subcaplabelsep Analogous to \captionlabelsep we define \subcaplabelsep.

```
382        \newcommand*\subcaplabelsep{\hskip\subfiglabelskip}
```

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

```
383        \let\caption@subfig@hss\hss
```

```
384      \fi
```

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

\ifsubfigcapwidth Analogous to \ifcaptionwidth, \captionindent & \captionlabeldelim
\subcapindent we define \ifsubfigcapwidth, \subcapindent & \subcaplabeldelim
\subcaplabeldelim
```
385      \newif\ifsubfigcapwidth
386      \newdimen\subcapindent
387      \newcommand*\subcaplabeldelim{}
```

\subcapstyle Analogous to \captionstyle we define \subcapstyle and set it (via \setsubcapstyle)
to a appropriate value.
```
388      \newcommand*\subcapstyle[1]{%
389        \expandafter\ifx\csname caption@@#1\endcsname\relax
390          \PackageError{caption2}{Undefined caption style '#1'}{\caption@eh}%
391        \else
392          \def\caption@substyle{#1}%
393        \fi}
394      \setsubcapstyle
```

\@thesubfigure The subfigure package makes use of \subcaplabelfont and \subfiglabelskip
\@thesubtable within its \@thesubxxx macros. This is totally in contrast to the way the caption2
package handle these settings. So we redefine the \@thesubxxx to be just the plain
label and nothing else.
```
395      \renewcommand*\@thesubfigure{\thesubfigure}
396      \renewcommand*\@thesubtable{\thesubtable}
```

\@makesubfigurecaption Now we are ready to redefine \@makesubfigurecaption.
\@makesubtablecaption
```
397      \let\@makesubfigurecaption\caption@makesubcaption
398      \let\@makesubtablecaption\caption@makesubcaption
```

```
399    \fi}
```

That's all folks!

```
400 \let\caption@package\undefined
```

17

References

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

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

[3] Sebastian Rahtz and Leonor Barroca: *A style option for rotated objects in LaTeX*, 1997/09/26

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

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

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

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