

# The caption2 package<sup>\*</sup>

Axel Sommerfeldt

`caption@sommerfee.de`

2007/04/06

## **THIS PACKAGE IS OBSOLETE!**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 It was released in December 2003 and superseded the neglected caption2 package.

(In parallel, Steven Cochran released the subfig package which superseded the subfigure package.)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which try to tell you that the caption2 package should be used instead of the caption package – these hints are outdated since December 2003.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. You will also find a section called 'Compatibility to older versions' there which should help you with the migration process. If all this should fail you can write me an e-mail asking for help.

---

<sup>\*</sup>This package has version number v2.2, last revised 2007/04/06.



## **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 – they should still compile fine as they are. If they don't please don't hesitate to write me an e-mail asking for maintainance.



# 1 The Implementation

## 1.1 Identificaton

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption2}[2007/04/06 v2.2 Customising captions (AS)]
3 \PackageWarning{caption2}{%
4   *** THIS PACKAGE IS OBSOLETE, PLEASE USE caption ***}
```

## 1.2 Loading the caption3 kernel

```
5 \RequirePackage{caption3}[2007/03/04] % needs v3.0m or newer
6 \def\caption@package{caption2}
```

## 1.3 Check against the regular caption package

```
7 \@ifpackageloaded{caption}{%
8   \caption@error{You can't use both the caption *and* caption2 package}%
9   \endinput
10 }{}
```

## 1.4 Preliminary declarations

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

```
11 \newcommand*\captionlabeldelim{}
12 \newcommand*\captionlabelsep{}
```

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

```
13 \newcommand*\ifcaptionwidth{\ifdim\captionwidth>\z@}
14 \newif\ifcaption@temp
```

`\ifcaptionlabel` If `\ifcaptionlabel` is not set the caption label should be suppressed; we need this flag to support the `\caption*` command.

```
15 \def\ifcaptionlabel{%
16   \ifcaption@star
17     \caption@tempfalse
18   \else
19     \caption@temptrue
20   \fi
21   \ifcaption@temp}
22 \def\captionlabeltrue{\caption@starfalse}
23 \def\captionlabelfalse{\caption@startrue}
```



`\ifonelinecaptions` If `\ifonelinecaptions` 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.

```

24 \def\ifonelinecaptions{%
25   \caption@ifslc{\caption@temptrue}{\caption@tempfalse}%
26   \ifcaption@temp}
27 \def\onelinecaptionstrue{\caption@setbool{slc}{1}}
28 \def\onelinecaptionsfalse{\caption@setbool{slc}{0}}

```

`\ifignoreLTcapwidth` If `\ifignoreLTcapwidth` is set we ignore the `\LTcapwidth` of `longtable`.

```

29 \newif\ifignoreLTcapwidth

```

`\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`.

```

30 \newcommand*\normalcaptionparams{%
31   \let\captionsize\@empty
32   \renewcommand*\captionfont{\captionsize}%
33   \let\captionlabelfont\@empty
34   \renewcommand*\captionlabeldelim{:}%
35   \renewcommand*\captionlabelsep{\space}%
36   \setcaptionmargin\z@
37   \setlength\captionindent\z@
38   \onelinecaptionstrue}

```

`\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.

```

39 \newcommand*\defcaptionstyle[1]{%
40   \@namedef{caption@@#1}}
41 %
42 \newcommand*\newcaptionstyle[1]{%
43   \expandafter\ifx\csname caption@@#1\endcsname\relax
44     \expandafter\defcaptionstyle
45   \else
46     \PackageError{caption2}{Caption style `#1' already defined}{\caption@eh}%
47     \expandafter\@gobbletwo
48   \fi
49   {#1}}
50 %
51 \newcommand*\renewcaptionstyle[1]{%
52   \expandafter\ifx\csname caption@@#1\endcsname\relax
53     \PackageError{caption2}{Caption style `#1' undefined}{\caption@eh}%
54     \expandafter\@gobbletwo
55   \else
56     \expandafter\defcaptionstyle
57   \fi
58   {#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.

```

59 \newcommand*\dummycaptionstyle[2]{%
60   \defcaptionstyle{#1}{%
61     \expandafter\ifx\csname caption@@\caption@style\expandafter\endcsname%
62       \csname caption@@#1\endcsname
63     \PackageError{caption2}{You can't use the caption style '#1' directly}{%
64       The caption style '#1' is only a dummy and does not really exists.%
65       \MessageBreak You have to redefine it (with \protect\renewcaptionstyle)
66       before you can select\MessageBreak it with \protect\captionstyle.
67       \space\caption@eh}%
68   \else
69     #2\usecaptionstyle\caption@style
70   \fi}}

```

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

```

71 \newcommand*\captionstyle[1]{%
72   \expandafter\ifx\csname caption@@#1\endcsname\relax
73     \PackageError{caption2}{Undefined caption style '#1'}{\caption@eh}%
74   \else
75     \def\caption@style{#1}%
76   \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`.

```

77 \newcaptionstyle{normal}{\caption@make{normal}}
78 \newcaptionstyle{center}{\caption@make{center}}
79 \newcaptionstyle{centerlast}{\caption@make{centerlast}}
80 \newcaptionstyle{flushleft}{\caption@make{flushleft}}
81 \newcaptionstyle{flushright}{\caption@make{flushright}}
82 \newcaptionstyle{hang}{\caption@make{hang}}
83 \newcaptionstyle{hang+center}{\caption@make{hang@center}}
84 \newcaptionstyle{hang+centerlast}{\caption@make{hang@centerlast}}
85 \newcaptionstyle{hang+flushleft}{\caption@make{hang@flushleft}}
86 \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 necessary.

```

87 \newcommand*\caption@makecaption[1]{%
88   \usecaptionmargin
89   \ifcaptionlabel
90     \def\caption@label{%

```



```

91      {\captionlabelfont\captionlabel\captionlabeldelim}\captionlabelsep}%
92  \else
93      \let\caption@label\@empty
94  \fi

95  \captionfont
96  \onelinecaption
97  {\caption@label\captiontext}%
98  {\parbox[b]\captionlinewidth{\strut\@nameuse{caption@@#1}\par}\par}}

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

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

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

101 \newcommand*\caption@@@center{%
102  \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].

103 \newcommand*\caption@@@centerlast{%
104  \centerlast\caption@label\captiontext}

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

105 \newcommand*\caption@@@flushleft{%
106  \raggedright\caption@label\captiontext}%

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

107 \newcommand*\caption@@@flushright{%
108  \raggedleft\caption@label\captiontext}%

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

109 \newcommand*\caption@@@hang{%
110  \sbox\@tempboxa{\caption@label}%
111  \hangindent\wd\@tempboxa\noindent
112  \usebox\@tempboxa\caption@hangplus\captiontext}

113 \newcommand*\caption@hangplus{ }

\caption@@@hang@center The ‘hang+flushleft’ caption style.

114 \newcommand*\caption@@@hang@center{%
115  \let\caption@hangplus\centering\caption@@@hang}

```



<code>\caption@@@hang@centerlast</code>	<p>The ‘hang+flushleft’ caption style.</p> <pre> 116 \newcommand*\caption@@@hang@centerlast{% 117   \let\caption@hangplus\centerlast\caption@@@hang} </pre>
<code>\caption@@@hang@flushleft</code>	<p>The ‘hang+flushleft’ caption style.</p> <pre> 118 \newcommand*\caption@@@hang@flushleft{% 119   \let\caption@hangplus\raggedright\caption@@@hang} </pre>
<code>\caption@@@indent</code>	<p>The ‘indent’ caption style. Is is quite like the ‘hang’ style but the indentation is given as <code>\captionindent</code>.</p> <pre> 120 \newcommand*\caption@@@indent{% 121   \hangindent\captionindent\noindent 122   \caption@label\captiontext} </pre>

## 1.5 Options

<code>normal</code> <code>center</code> <code>centerlast, anne</code> <code>flushleft</code> <code>flushright</code> <code>hang, isu</code> <code>indent</code>	<p>These options will set the caption style. (‘normal’ is the default one.)</p> <p>The options ‘anne’ and ‘isu’ are for backward compatibility only.</p> <pre> 123 \DeclareOption{normal}{\captionstyle{normal}} 124 \DeclareOption{center}{\captionstyle{center}} 125 \DeclareOption{centerlast}{\captionstyle{centerlast}} 126 \DeclareOption{flushleft}{\captionstyle{flushleft}} 127 \DeclareOption{flushright}{\captionstyle{flushright}} 128 \DeclareOption{anne}{\ExecuteOptions{centerlast}} 129 \DeclareOption{hang}{\captionstyle{hang}} 130 \DeclareOption{hang+center}{\captionstyle{hang+center}} 131 \DeclareOption{hang+centerlast}{\captionstyle{hang+centerlast}} 132 \DeclareOption{hang+flushleft}{\captionstyle{hang+flushleft}} 133 \DeclareOption{isu}{\ExecuteOptions{hang}} 134 \DeclareOption{indent}{\captionstyle{indent}} </pre>
<code>scriptsize</code> <code>footnotesize</code> <code>small</code> <code>normalsize</code> <code>large, Large</code>	<p>These options will set the caption size. We use <code>\g@addto@macro</code> so more than one option can be set.</p> <pre> 135 \DeclareOption{scriptsize}{\g@addto@macro\captionsize\scriptsize} 136 \DeclareOption{footnotesize}{\g@addto@macro\captionsize\footnotesize} 137 \DeclareOption{small}{\g@addto@macro\captionsize\small} 138 \DeclareOption{normalsize}{\g@addto@macro\captionsize\normalsize} 139 \DeclareOption{large}{\g@addto@macro\captionsize\large} 140 \DeclareOption{Large}{\g@addto@macro\captionsize\Large} </pre>
<code>up, it, sl, sc</code> <code>md, bf</code> <code>rm, sf, tt</code>	<p>These options will set the caption label.</p> <pre> 141 \DeclareOption{up}{\g@addto@macro\captionlabelfont\upshape} 142 \DeclareOption{it}{\g@addto@macro\captionlabelfont\itshape} 143 \DeclareOption{sl}{\g@addto@macro\captionlabelfont\slshape} 144 \DeclareOption{sc}{\g@addto@macro\captionlabelfont\scshape} </pre>



```

145 \DeclareOption{md}{\g@addto@macro\captionlabelfont\mdseries}
146 \DeclareOption{bf}{\g@addto@macro\captionlabelfont\bfseries}
147 \DeclareOption{rm}{\g@addto@macro\captionlabelfont\rmfamily}
148 \DeclareOption{sf}{\g@addto@macro\captionlabelfont\sffamily}
149 \DeclareOption{tt}{\g@addto@macro\captionlabelfont\ttfamily}

online These options will set the ‘online’ flag. (‘online’ is the default.)
nooneline
150 \DeclareOption{online}{\onlinecaptionstrue}
151 \DeclareOption{nooneline}{\onlinecaptionsfalse}

\caption@setpackage 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.
152 \newcommand*\caption@setpackage[1]{\@namedef{caption@pkt@#1}}

float These options will enable or suppress the support of the packages float, longtable, and
longtable subfigure.
subfigure
153 \DeclareOption{float}{\caption@twozerofalse\caption@setpackage{float}{1}}
154 \DeclareOption{longtable}{\caption@twozerofalse\caption@setpackage{longtable}{1}}
155 \DeclareOption{subfigure}{\caption@twozerofalse\caption@setpackage{subfigure}{1}}

none These options will enable or suppress the support of all the above packages.
all
156 \DeclareOption{none}{\caption@twozerofalse
157 \caption@setpackage{float}{0}\caption@setpackage{longtable}{0}%
158 \caption@setpackage{subfigure}{0}}
159 \DeclareOption{all}{\ExecuteOptions{float,longtable,subfigure}}

ruled The option ‘ruled’ introduced in caption v1.2 is obsolete now, but we will still support it.
boxed The option ‘boxed’ was introduced in version 2.0 and is obsolete now, too.
160 \newif\ifcaption@ruled
161 \DeclareOption{ruled}{\caption@ruledtrue}
162 \DeclareOption{boxed}{}

ignoreLTcapwidth This option will make the caption code ignore the setting of \LTcapwidth and use the
setting of \setcaptionmargin or \setcaptionwidth instead.
163 \DeclareOption{ignoreLTcapwidth}{\ignoreLTcapwidthtrue}

debug This option will put additional debug information in the log file.
164 \DeclareOption{debug}{\captionsetup{debug}}

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.)
165 \newif\ifcaption@twozero
166 \normalcaptionparams

```



```

167 \ExecuteOptions{none,normal}
168 \caption@twozerotrue
169 \ProcessOptions*
170 \ifcaption@twozero
171   \PackageInfo{caption2}{Running in caption2 v2.0 compatibility mode}
172 \fi

```

## 1.6 More declarations

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

```

173 \def\captionof{\@ifstar{\caption@of{\caption*}}{\caption@of\caption}}
174 \newcommand*\caption@of[2]{\def\@capttype{#2}\#1}

```

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

```

175 \@ifundefined{abovecaptionskip}{%
176   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
177 \@ifundefined{belowcaptionskip}{%
178   \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.)

```

179 \newdimen\captionlinewidth
180 \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.

```

181 \newcommand*\usecaptionmargin{%
182   \ifcaptionwidth
183     \leftskip\captionlinewidth
184     \advance\leftskip by -\captionwidth
185     \divide\leftskip by 2
186     \rightskip\leftskip
187     \captionlinewidth\captionwidth
188   \else
189     \leftskip\captionmargin
190     \rightskip\captionmargin

```



```

191 \advance\captionlinewidth by -2\captionmargin
192 \fi
193 \realcaptionwidth\captionlinewidth}

```

`\onelinecaption` This macro definition helps setting captions the  $\text{\LaTeX}$  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.)

```

194 \newcommand\onelinecaption[1]{%
195 \let\next\@firstofone
196 \ifonelinecaptions
197 \sbox\@tempboxa{#1}%
198 \ifdim\wd\@tempboxa >\captionlinewidth
199 \else
200 \def\next{\centering\usebox\@tempboxa\par}\@gobble}%
201 \fi
202 \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.

```

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

```

`\@makecaption` This is the heart of the `caption2` package – the redefinition of the core caption code. It was taken from the  $\text{\LaTeX 2}_{\epsilon}$  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`.

```

216 \renewcommand\@makecaption[2]{%
217 \vskip\abovecaptionskip
218 \captionlinewidth\hsize
219 \realcaptionwidth\hsize
220 \def\captionlabel{#1}%
221 \def\captiontext{#2}%
222 \usecaptionstyle\caption@style
223 \vskip\belowcaptionskip}

```



## 1.7 Support of other packages

`\caption@ifpackage` 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.

```

224 \newcommand*\caption@ifpackage[3]{%
225   \if1\@nameuse{caption@pkt@#1}%
226   \ifundefined{#2}%
227     {\let\next\AtBeginDocument}%
228     {\let\next\@firstofone}%
229   \else\ifcaption@twozero
230     \ifundefined{#2}%
231     {\#3\let\next\@gobble}%
232     {\let\next\@firstofone}%
233   \else
234     \#3\let\next\@gobble
235   \fi\fi
236   \expandafter\let\csname caption@pkt@#1\endcsname\undefined
237   \caption@ifdebug{%
238     \ifx\next\@gobble\PackageInfo{caption2}{#1 => gobble}%
239     \else\ifx\next\@firstofone\PackageInfo{caption2}{#1 => firstofone}%
240     \else\ifx\next\AtBeginDocument\PackageInfo{caption2}{#1 => AtBeginDocument}%
241     \fi\fi\fi}{}%
242   \next}

```

### 1.7.1 Support of the float package

```

243 \caption@ifpackage{float}{floatc@plain}{}{%
244   \ifx\floatc@plain\relax
245     \PackageWarning{caption2}{%
246       Option 'float' was set but there is no float package loaded}
247   \else
248     \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`.

```

249   \newcommand\caption@floatc[3]{%
250     \ifx\captionlabelfont\@empty
251       \let\captionlabelfont\@fs@cfont
252     \fi
253     \captionlinewidth\hsize
254     \realcaptionwidth\hsize

```



```

255     \def\captionlabel{#2}%
256     \def\captiontext{#3}%
257     \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.

```

258     \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.

```

259     \ifcaption@ruled
260         \dummycaptionstyle{ruled}{\onelinecaptionsfalse\setcaptionmargin{\z@}}%
261     \else
262         \newcaptionstyle{ruled}{%
263             \ifcaptionlabel
264                 {\@fs@cfont\captionlabel}\space%
265             \fi\captiontext\par}%
266     \fi
267     \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 `\@capttype` 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@<floatttype>` 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.

```

268     \renewcommand*\caption@of[2]{\def\@capttype{#2}%
269         \@ifundefined{fst@#2}{}{%
270             \@nameuse{fst@#2}%
271             \@ifundefined{@float@setevery}{}{\@float@setevery{#2}}%
272             \let\caption@fs@capt\@fs@capt
273             \let\@fs@capt\caption@of@float}%
274     #1}

```



```

275 \newcommand\caption@of@float[2]{\egroup
276 \vskip\abovecaptionskip
277 \normalsize\caption@fs@capt{#1}{#2}%
278 \vskip\belowcaptionskip
279 \bgroup}%

280 \fi}

```

### 1.7.2 Support of the longtable package

```

281 \caption@ifpackage{longtable}{LT@makecaption}{\}%
282 \ifx\LT@makecaption\relax
283 \PackageWarning{caption2}{%
284 Option 'longtable' was set but there is no longtable package loaded}
285 \else
286 \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.)

```

287 \dummycaptionstyle{longtable}{\}

288 \renewcommand\LT@makecaption[3]{%
289 \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\hsize{%
290 \ifignoreLTcapwidth
291 \else
292 \setcaptionwidth\LTcapwidth
293 \fi
294 \captionlinewidth\hsize
295 \realcaptionwidth\hsize
296 \captionlabelfalse#1\captionlabeltrue
297 \def\captionlabel{#2}%
298 \def\captiontext{#3}%
299 \usecaptionstyle{longtable}%
300 \endgraf\vskip\baselineskip}%
301 \hss}}}

302 \fi}

```



### 1.7.3 Support of the subfigure package

Some of the following code will not work within `\if`, because of the (yet) undefined `\ifxxs`. 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.

```

303 \newcommand*\setsubcapstyle{%
304   \@ifundefined{subcapraggedright}{false}%
305   \newif\ifsubcapraggedright}%
306 \ifsubcaphang
307   \ifsubcapcenter
308     \subcapstyle{hang+center}%
309   \else\ifsubcapcenterlast
310     \subcapstyle{hang+centerlast}%
311   \else\ifsubcapraggedright
312     \subcapstyle{hang+flushleft}%
313   \else
314     \subcapstyle{hang}%
315   \fi\fi\fi
316 \else\ifsubcapcenter
317   \subcapstyle{center}%
318 \else\ifsubcapcenterlast
319   \subcapstyle{centerlast}%
320 \else\ifsubcapraggedright
321   \subcapstyle{flushleft}%
322 \else
323   \subcapstyle{normal}%
324 \fi\fi\fi\fi

```

`\caption@makesubcaption` This will typeset the subcaption. We just set all our `\captionxxx` values to the values of `\subcapxxx` 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
  {\captionlabelfont\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.)

```

325 \newcommand\caption@makesubcaption[2]{%
326   \renewcommand*\captionfont{\subcapsize\subcapfont}%
327   \renewcommand*\captionlabelfont{\normalfont\subcapsize\subcaplabelfont}%
328   \let\captionlabel\captionlabel
329   \let\captionlabelsep\subcaplabelsep
330   \ifsubfigcapwidth\captionwidthtrue\else\captionwidthfalse\fi
331   \setlength\captionmargin\subfigcapmargin
332   \setlength\captionwidth\subfigcapwidth
333   \captionindent\subcaptionindent
334   \ifsubcapnoonline\onelinecaptionsfalse\else\onelinecaptionstrue\fi
335   \hbox to\@tempdima{%
336     \caption@subfig@hss\parbox[t]\@tempdima{%
337       \captionlinewidth\@tempdima
338       \realcaptionwidth\@tempdima
339       \captionlabeltrue
340       \def\captionlabel{#1}%
341       \def\captiontext{\ignorespaces #2}%
342       \usecaptionstyle\caption@substyle}%
343     \caption@subfig@hss}}

```

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

```

344 \caption@ifpackage{subfigure}{\@makesubfigurecaption}{%
345   \let\setsubcapstyle\undefined
346   \let\caption@makesubcaption\undefined}{%
347   \ifx\@makesubfigurecaption\relax
348     \PackageWarning{caption2}{%
349       Option 'subfigure' was set but there is no subfigure package loaded}
350   \let\setsubcapstyle\undefined
351   \let\caption@makesubcaption\undefined
352   \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.

```

353   \ifx\subcapfont\undefined
354     \PackageInfo{caption2}{subfigure package v2.0 detected}

```



<code>\subcapfont</code>	<p>We define <code>\subcapfont</code> here so we can use it later in common code for subfigure v2.0 and v2.1 (or newer).</p> <pre> 355      \let\subcapfont\@empty </pre>
<code>\subfigcapwidth</code> <code>\setsubcapmargin</code> <code>\setsubcapwidth</code>	<p>Analogous to <code>\captionwidth</code>, <code>\setcaptionmargin</code>, and <code>\setcaptionwidth</code> we define <code>\subfigcapwidth</code>, <code>\setsubcapmargin</code>, and <code>\setsubcapwidth</code>.</p> <p><b>Note:</b> <code>\subfigcapmargin</code> is a command in v2.0 of subfigure. So we make <code>\subfigcapwidth</code> a command, too.</p> <pre> 356      \newcommand\subfigcapwidth{\z@} 357      \newcommand\setsubcapmargin{% 358          \subfigcapwidthfalse 359          \renewcommand*\subfigcapmargin} 360      \newcommand\setsubcapwidth{% 361          \subfigcapwidthtrue 362          \renewcommand*\subfigcapwidth} </pre>
<code>\subcaplabelsep</code>	<p>Analogous to <code>\captionlabelsep</code> we define <code>\subcaplabelsep</code>.</p> <pre> 363      \newcommand\subcaplabelsep{\space} </pre>
<code>\caption@subfig@hss</code>	<p>This will be uses within the caption code itself.</p> <pre> 364      \let\caption@subfig@hss\hfil  365      \else 366      \PackageInfo{caption2}{subfigure package v2.1 (or newer) detected} </pre>
<code>\subfigcapwidth</code> <code>\setsubcapmargin</code> <code>\setsubcapwidth</code>	<p>Analogous to <code>\captionwidth</code>, <code>\setcaptionmargin</code>, and <code>\setcaptionwidth</code> we define <code>\subfigcapwidth</code>, <code>\setsubcapmargin</code>, and <code>\setsubcapwidth</code>.</p> <p><b>Note:</b> <code>\subfigcapmargin</code> is a length in v2.1 of subfigure. So we make <code>\subfigcapwidth</code> a length, too.</p> <pre> 367      \newdimen\subfigcapwidth 368      \newcommand\setsubcapmargin{% 369          \subfigcapwidthfalse 370          \setlength\subfigcapmargin} 371      \newcommand\setsubcapwidth{% 372          \subfigcapwidthtrue 373          \setlength\subfigcapwidth} </pre>
<code>\subcaplabelsep</code>	<p>Analogous to <code>\captionlabelsep</code> we define <code>\subcaplabelsep</code>.</p> <pre> 374      \newcommand\subcaplabelsep{\hskip\subfiglabelskip} </pre>
<code>\caption@subfig@hss</code>	<p>This will be uses within the caption code itself.</p> <pre> 375      \let\caption@subfig@hss\hss  376      \fi </pre>



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
377 \newif\ifsubfigcapwidth
378 \newdimen\subcapindent
379 \newcommand\subcaplabeldelim{}

\subcapstyle Analogous to \captionstyle we define \subcapstyle and set it (via \setsubcapstyle)
to a appropriate value.
380 \newcommand*\subcapstyle[1]{%
381 \expandafter\ifx\csname caption@@#1\endcsname\relax
382 \PackageError{caption2}{Undefined caption style '#1'}{\caption@eh}%
383 \else
384 \def\caption@substyle{#1}%
385 \fi}
386 \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.

387 \renewcommand*\@thesubfigure{\thesubfigure}
388 \renewcommand*\@thesubtable{\thesubtable}

\@makesubfigurecaption Now we are ready to redefine \@makesubfigurecaption.
\@makesubtablecaption
389 \let\@makesubfigurecaption\caption@makesubcaption
390 \let\@makesubtablecaption\caption@makesubcaption

391 \fi}

That's all folks!

392 \let\caption@ifpackage\undefined

```

## 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 L<sup>A</sup>T<sub>E</sub>X*, 1997/09/26
- [4] Steven Douglas Cochran: *The subfigure package*, 2002/02/14



- [5] Michel Goossens, Frank Mittelbach and Alexander Samarin: *The L<sup>A</sup>T<sub>E</sub>X Companion*, Addison-Wesley, Reading, Massachusetts, 1994.
- [6] Anne Brüggemann-Klein: *Einführung in die Dokumentverarbeitung*, B.G. Teubner, Stuttgart, 1989
- [7] Helmut Kopka: *L<sup>A</sup>T<sub>E</sub>X– Erweiterungsmöglichkeiten*, 3. überarbeitete Auflage, Addison-Wesley, Bonn, 1991