Customizing captions of floating environments using the caption package*

Axel Sommerfeldt
axel.sommerfeldt@f-m.fm

2011/08/11

Abstract

The caption package offers customization of captions in floating environments such `figure` and `table` and cooperates with many other packages.

Please note: Many document classes already have build-in options and commands for customizing captions. If these possibilities are sufficient for you, there is usually no need for you to use the caption package at all. And if you are just interested in using the command `\captionof`, loading of the very small capt-of package is usually sufficient.

Introduction

Within the standard LaTeX document classes captions haven't received the attention they deserve. Simply typeset as an ordinary paragraph there is no remarkable visual difference from the rest of the text, like here:

Figure 1: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

There should be possibilities to change this; for example, it would be nice to make the text of the caption a little bit smaller as the normal text, add an extra margin, typeset the caption label with the same font family and shape as your headings etc. Just like this one:

Figure 2 – White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

You can do this easily with this package as there are many ready-to-use caption formatting options, but you are free to define your very own stuff, too.

Please note that the caption package is only controlling the look & feel of the captions. It does *not* control the placement of the captions. (But you could do so by using other packages like the floatrow package[8].)

*This package has version number v3.2, last revised 2011/08/06.

Contents

1 Using this package

\usepackage

Insert

```
\usepackage[⟨options⟩]{caption}[2011/08/06]
```

into the preamble of your document, i.e. the part of your document between `\documentclass` and `\begin{document}`. The options control how your captions will look like; e.g.,

```
\usepackage[margin=10pt,font=small,labelfont=bf,
            labelsep=endash]{caption}
```

would result in captions looking like the second one in the introduction.

\captionsetup

For a later change of options the caption package provides the command

```
\captionsetup[⟨float type⟩]{⟨options⟩}
```

.

So

```
\usepackage[margin=10pt,font=small,labelfont=bf]{caption}
```

and

```
\usepackage{caption}
\captionsetup{margin=10pt,font=small,labelfont=bf}
```

are equal in their results.

It's good to know that `\captionsetup` has an effect on the current environment only. So if you want to change settings for the current `figure` or `table` only, just place the `\captionsetup` command inside the `figure` or `table` right before the `\caption` command. For example

```
\begin{figure}
  ...
  \captionsetup{singlelinecheck=off}
  \caption{...}
\end{figure}
```

switches the single-line-check off, but only for this `figure`, so all the other captions remain untouched.

(For a detailed description of `\captionsetup` see section 3.2: *Setting options*.)

2 Options

2.1 Formatting

format= A figure or table caption mainly consists of three parts: the caption label, which says if this object is a 'Figure' or 'Table' and what number is associated with it, the caption text itself, which is normally a short description of contents, and the caption separator which separates the text from the label.

The *caption format* determines how this information will be presented; it is specified with the option

`format=`⟨*format name*⟩ ,

having the name of the caption format as its argument.

There are two standard caption formats:

`plain`	Typesets the captions as a normal paragraph.
`hang`	Indents the caption text, so it will 'hang' under the first line of the text.
`...`	Own formats can be defined using `\DeclareCaptionFormat`. (See section 4: *Own enhancements*)

An example: Specifying the option

```
format=hang
```

yields captions like this:

Figure 3: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

indention= For both formats (`plain` and `hang`) you can setup an extra indention starting at the second line of the caption. You do this with the option

`indention=`⟨*amount*⟩ .

Two examples:

```
format=plain,indention=.5cm
```

Figure 4: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

```
format=hang,indention=-0.5cm
```

Figure 5: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

`labelformat=` With the option

`labelformat=`⟨*label format name*⟩

you specify how the caption label will be typeset. There are five standard caption label formats:

`default`	The caption label will be typeset as specified by the document class, usually this means the name and the number (like `simple`). (This is the default behaviour.)
`empty`	The caption label will be empty.
`simple`	The caption label will be typeset as a name and a number.
`brace`	The caption label will be closed with a single (right) brace.
`parens`	The number of the caption label will be typeset in parentheses.
`...`	Own label formats can be defined using `\DeclareCaption-LabelFormat`. (See section 4: *Own enhancements*)

An example: Using the options

```
format=plain,labelformat=parens,labelsep=quad
```

gives captions like this one:

Figure (6) White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

Note: Some environments, e.g. the `algorithm` environment offered by the algorithm2e package, might react allergic to a change of the caption label format.

`labelsep=` With the option

`labelsep=`⟨*label separator name*⟩

you specify what caption separator will be used.[1] You can choose one of the following:

`none`	There is no caption separator.
`colon`	The caption label and text will be separated by a colon and a space.
`period`	The caption label and text will be separated by a period and a space.
`space`	The caption label and text will be separated by a single space.
`quad`	The caption label and text will be separated by a `\quad`.
`newline`	The caption label and text will be separated by a line break (`\\`). Please note that this separator does not work with all caption formats (e.g. `format=hang`), and you'll get an error message trying such combination of options.

[1] If the caption label or the caption text is empty, no separator will be used.

`endash` The caption label and text will be separated by an en-dash, surrounded by spaces (␣--␣).

`...` Own separators can be defined using `\DeclareCaptionLabelSeparator`. (See section 4: *Own enhancements*)

Three examples:

```
format=plain,labelsep=period
```

Figure 7. White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

```
format=plain,labelsep=newline,singlelinecheck=false
```

Figure 8
White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

```
format=plain,labelsep=endash
```

Figure 9 – White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

`textformat=` With the option

`textformat=`⟨*text format name*⟩

you specify how the caption text will be typeset. There are two standard caption text formats:

`simple` The caption text will be typeset as it is.

`period` The caption text will be followed by a period.

`...` Own text formats can be defined using `\DeclareCaptionTextFormat`. (See section 4: *Own enhancements*)

2.2 Justification

justification= As addition to the caption format you could also specify a *caption justification*; it is specified with the option

`justification=`⟨*justification name*⟩ .

You can choose one of the following:

`justified`	Typesets the caption as a normal paragraph.
`centering`	Each line of the caption will be centered.
`centerlast`	The last line of each paragraph of the caption text will be centered.
`centerfirst`	Only the first line of the caption will be centered.
`raggedright`	Each line of the caption will be moved to the left margin.
`RaggedRight`	Each line of the caption will be moved to the left margin, too. But this time the command `\RaggedRight` of the ragged2e package will be used to achieve this. The main difference to `raggedright` is that the word breaking algorithm of TEX will work inside captions.[2]
`raggedleft`	Each line of the caption will be moved to the right margin.
`...`	Own justifications can be defined using `\DeclareCaptionJustification`. (See section 4: *Own enhancements*)

Three examples:

```
format=plain,justification=centerlast
```

Figure 10: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

```
format=hang,justification=raggedright
```

Figure 11: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

```
format=plain,labelsep=newline,justification=centering
```

Figure 12
White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

[2] The need for the ragged2e package will be detected at run-time, therefore you maybe need a second LaTeX run if this option is used for the first time.

singlelinecheck= The standard LaTeX document classes (article, report, and book) automatically center a caption if it fits in one single line:

Figure 13: A short caption.

The caption package adapts this behavior and therefore usually ignores the justification & indention you have set with `justification=` & `indention=` in such case. But you can switch this special treatment of such short captions off with the option

`singlelinecheck=`⟨*bool*⟩ .

Using `false`, `no`, `off` or `0` for ⟨*bool*⟩ switches the extra centering off:

```
singlelinecheck=false
```

Doing so the above short caption would look like

Figure 13: A short caption.

You switch the extra centering on again by using `true`, `yes`, `on` or `1` for ⟨*bool*⟩. (The default is on.)

2.3 Fonts

font= labelfont= textfont= There are three font options which affects different parts of the caption: One affecting the whole caption (`font`), one which only affects the caption label and separator (`labelfont`) and at least one which only affects the caption text (`textfont`). You set them up using the options

`font={`⟨*font options*⟩`}` ,
`labelfont={`⟨*font options*⟩`}` , and
`textfont={`⟨*font options*⟩`}` ,

where ⟨*font options*⟩ is a list of comma separated font options.

And these are the available font options:

`scriptsize`	Very small size
`footnotesize`	The size usually used for footnotes
`small`	Small size
`normalsize`	Normal size
`large`	Large size
`Large`	Even larger size

`normalfont`	Normal shape & series & family
`up`	Upright shape
`it`	*Italic shape*
`sl`	*Slanted shape*
`sc`	SMALL CAPS SHAPE
`md`	Medium series
`bf`	**Bold series**
`rm`	Roman family
`sf`	Sans Serif family
`tt`	`Typewriter family`
`singlespacing`	Single spacing (See section 6.12: *setspace*)
`onehalfspacing`	One-and-a-half spacing (See section 6.12: *setspace*)
`doublespacing`	Double spacing (See section 6.12: *setspace*)
`stretch=`⟨*amount*⟩	`\setstretch{`⟨*amount*⟩`}` (See section 6.12: *setspace*)
`normalcolor`	`\normalcolor`
`color=`⟨*colour*⟩	`\color{`⟨*colour*⟩`}` (If the color or xcolor package is loaded, see section 4: *Own enhancements* for an example)
`normal`	The combination of the options `normalcolor`, `normalfont`, `normalsize`, and `singlespacing`
`...`	Own font options can be defined using `\DeclareCaptionFont`. (See section 4: *Own enhancements*)

If you use only one of these options you can omit the braces; e.g., the options `font={small}` and `font=small` will give the same result.

Three examples:

```
font=it,labelfont=bf
```

Figure 14: *White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.*

```
labelfont=bf,textfont=it
```

Figure 15: *White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.*

```
font={small,stretch=0.80}
```

Figure 16: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

font+=
labelfont+=
textfont+=

You can also add font options to the current ones, so for example

```
\captionsetup{font=small}
\captionsetup{font+=it}
```

is identical to

```
\captionsetup{font={small,it}}
```

2.4 Margins and further paragraph options

margin=
width=

For all captions you can specify *either* an extra margin *or* a fixed width: [3]

```
margin=⟨amount⟩                          –or–
margin={⟨left amount⟩,⟨right amount⟩}    –or–
 width=⟨amount⟩
```

If you specify just one ⟨*amount*⟩ for the margin, it will be used for both, the left and right margin, e.g. `margin=10pt` is equivalent to `margin={10pt,10pt}`. In two-side documents the left and right margin will be swapped on even pages. To prevent this you can specify the option `oneside` additionally, e.g. `\captionsetup{margin={0pt,10pt},oneside}`.

oneside
twoside

But if you are specifying a width, then both, the left and the right margin, will have the same amount.

Three examples will illustrating this:

```
margin=10pt
```

Figure 17: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

```
margin={1cm,0cm}
```

Figure 18: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

```
width=.75\textwidth
```

Figure 19: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

[3] Only fixed widths are supported here; if you are looking for a way to limit the width of the caption to the width of the figure or table, please take a look at the floatrow[8] or threeparttable[22] package.

Note: When the caption is placed beside the contents (for example in a `SCfigure` environment offered by the sidecap package[18]) or the figure is an in-text figure (for example in a `wrapfigure` environment offered by the wrapfig package[23]), the margin setting will be automatically reset to `0pt` at the very beginning of the environment. But if you really want to setup an extra margin for these environments, you can do so by setting this margin either inside the environment itself, of by specifying a margin for this particular environment, e.g. `\captionsetup[SCfigure]{margin=10pt}`.

`margin*=` There is also a starred variant of the `margin=` option, `margin*=`, which only changes the margin if no width was set.

`minmargin=` `maxmargin=` You can also set a minimum or maximum margin amount. This can be useful for limiting the margin amount in smaller environments, e.g. `minipages`. For example the SMF document classes limit the margin amount to `maxmargin=0.1\linewidth`. (See section 5.6: *SMF: smfart and smfbook*)

`parskip=` This option is useful for captions containing more than one paragraph. It specifies the extra vertical space inserted between them:

`parskip=`⟨*amount*⟩

One example:

`margin=10pt,parskip=5pt`

Figure 20: First paragraph of the caption. This one contains some test, just to show how these options affect the layout of the caption.

Second paragraph of the caption. This one contains some text, too, to show how these options affect the layout of the caption.

`hangindent=` The option

`hangindent=`⟨*amount*⟩

is for setting up a hanging indention starting from the second line of each paragraph. If the caption contains just a single paragraph, using this option leads to the same result as the option `indention=`⟨*amount*⟩ you already know about. But if the caption contains multiple paragraphs you will notice the difference:

`format=hang,indention=-.5cm`

Figure 21: First paragraph of the caption. This one contains some test, just to show how these options affect the layout of the caption.
Second paragraph of the caption. This one contains some text, too, to show how these options affect the layout of the caption.

`format=hang,hangindent=-.5cm`

Figure 22: First paragraph of the caption. This one contains some test, just to show how these options affect the layout of the caption.
Second paragraph of the caption. This one contains some text, too, to show how these options affect the layout of the caption.

Note: If your caption contains more than one paragraph, you have to specify an alternative caption for the list-of-figures using the optional argument of `\caption` or `\captionof`, otherwise you will get an error message.

2.5 Styles

style= A suitable combination of caption options is called *caption style*. You can compare them more or less to page styles which you set up with `\pagestyle`; the caption style provides all settings for a whole caption layout.

You switch to an already defined caption style with the option

`style=`⟨*style name*⟩ .

The caption package pre-defines two styles: `base` and `default`.

The `base` style puts all options you already know about to values reflecting the look of the captions when using one of the base LaTeX document classes article, report, and book. This means that specifying the option

```
style=base
```

has the same effect as specifying all these options:

```
format=plain,labelformat=default,labelsep=colon,
justification=justified,font={},labelfont={},
textfont={},margin=0pt,indention=0pt
parindent=0pt,hangindent=0pt,singlelinecheck=true
```

(But `justification=centering,indention=0pt` will be set if the caption fits into a single line.)

In contrast the `default` style follows the default values, reflecting the look of the captions given by the document class you actually use. This style is selected by default and represents these options:

```
format=default,labelformat=default,labelsep=default,
justification=default,font=default,labelfont=default,
textfont=default,margin=0pt,indention=0pt
parindent=0pt,hangindent=0pt,singlelinecheck=true
```

(But again `justification=centering,indention=0pt` will be set if the caption fits into a single line.)

So if you use one of the base LaTeX document classes article, report, or book, both caption styles, `base` and `default`, point to (nearly) the same settings.

Note: Own caption styles can be defined using `\DeclareCaptionStyle`.
(See section 4: *Own enhancements*)

2.6 Skips

skip= The vertical space between the caption and the figure or table contents is controlled by the option

`skip=`⟨*amount*⟩ .

The standard LaTeX document classes article, report and book preset it to `skip=10pt`, but other document classes may use a different amount.

`position=`

The `\caption` command offered by LaTeX has a design flaw: The command does not know if it stands on the beginning of the figure or table, or at the end. Therefore it does not know where to put the space separating the caption from the content of the figure or table. While the standard implementation always puts the space above the caption in floating environments (and inconsistently below the caption in `longtables`), the implementation offered by this package is more flexible: By giving the option

`position=top` or `position=above`

it's assumed that the caption is standing at the *top* of the environment and therefore the space setup with `skip=`⟨*amount*⟩ is placed below the caption. (Please note that `position=top` does *NOT* mean that the caption is actually placed at the top of the figure or table. Instead the caption is usually placed where you place the `\caption` command.) But with

`position=bottom` or `position=below`

it's assumed that the caption is standing at the *bottom* of the environment and therefore the space is placed above the caption. And finally with

`position=auto` (which is the default setting)

the caption package tries its best to determine the actual position of the caption on its own. Please note that while this is successfully in most cases, it could give wrong results under rare circumstances.

`figureposition=`
`tableposition=`

The `position` option is especially useful when used together with the optional argument of the `\captionsetup` command. (See also section 3.2: *Setting options*)

For example

```
\captionsetup[table]{position=above}
```

causes all captions within tables to be treated as captions *above* the table (regarding spacing around it). Because this is a very common setting, the caption package offers the abbreviating options `figureposition=`⟨*pos*⟩ and `tableposition=`⟨*pos*⟩, e.g.

```
\usepackage[...,tableposition=top]{caption}
```

is equivalent to

```
\usepackage[...]{caption}
\captionsetup[table]{position=top}
```

Please note that the options `skip=`, `position=`, `figureposition=`, and `tableposition=` do not always have an effect. Since it's a matter of the document class to supply the environments `figure` and `table`, it could use its very own spacing, and could decide for itself if the caption will be typeset as "top" or "bottom" caption. For example the KOMA-Script document classes support the `skip=` setting, but will always typeset `figure` captions as "bottom" captions, and `table` captions are dependent on the global option `tablecaptionsabove` resp. `tablecaptionsbelow`. (See section 5.4: *KOMA-Script: scrartcl, scrreprt, and scrbook*)

Furthermore some packages control the behavior of the spacing above and below the caption for themself, e.g. the float, the floatrow, and the supertabular package.

Internally the skip between caption and contents is represented by `\abovecaptionskip` (which is always set above the caption in LaTeXs implementation). But there is a second value `\belowcaptionskip` (usually set to `0pt` by default) which is set below the caption in LaTeXs implementation, but on the other side than `\abovecaptionskip` by this package. So technically speaking, this package swaps the meaning of these two skips when `position=top` is set. Please note that there are several packages around which do the same trick (like the ftcap, the nonfloat, and the topcap package), so the usage of the caption option `position=` is not supported if one of these packages will be used, too.

2.7 Lists

`list=`

The `\caption` command usually places an entry in the List of Figures resp. List of Tables. You can either suppress that individually by giving an empty optional argument to `\caption` (see section 3.1: *Typesetting captions*), or programmatically by saying

`list=no` (or any other boolean value instead of `no`) .[4]

`listformat=`

With the option

`listformat=`⟨*list format name*⟩

you can specify how the `figure` or `table` number will be typeset within the List of Figures resp. List of Tables. There are five standard caption list formats:

`empty`	No number will be typeset.
`simple`	The number will be typeset with label prefix.
`parens`	The number will be typeset in parentheses, with label prefix.
`subsimple`	Same as `simple`, but without label prefix. (default)
`subparens`	Same as `parens`, but without label prefix.
`...`	Own list formats can be defined using `\DeclareCaptionListFormat`. (See section 4: *Own enhancements*)

The prefix (=`\p@figure` resp. `\p@table`), which runs ahead of the number (=`\thefigure` resp. `\thetable`) in lists (e.g. List of Figure/Table) and references, is usually empty, so the list formats `simple` and `subsimple` gives equal results; same with `parens` and `subparens`. But this can be different for sub-figures or sub-tables listed in the List of Figures resp. List of Tables, or when the label prefix is redefined for a different purpose.[5]

2.8 Names & Numbering

`figurename=` `tablename=`

You can change the name of the figures and tables with the options

`figurename=`⟨*name*⟩ and `tablename=`⟨*name*⟩ .

[4] Please note that the subfig package[20] is not supporting this option, it uses the counters `lofdepth` & `lotdepth` for this purpose instead.

[5] Sub-figures and sub-tables can be typeset using the subcaption or subfig package.

The document class usually sets them to “Figure” and “Table”, but for example with

```
figurename=Fig.
```

you can change the figure name to “Fig”.

If you use the babel package, please load the caption package *after* the babel package.

`name=` In contrast to the options `figurename` and `tablename` the option

```
name=⟨name⟩
```

changes the name of the *current* environment. This could be useful in conjunction with the optional argument of `\captionsetup`, e.g.

```
\captionsetup[wrapfigure]{name=Fig.}
```

changes the name to “Fig.” for all `wrapfigure`s (while all the other figure captions will still have “Figure” as name).

`listfigurename=` `listtablename=` Analogous to the options `figurename` and `tablename` you can change the title of the lists here. So for example

```
listfigurename=List of images
```

changes the title “List of Figures” to “List of images”.

`figurewithin=` `tablewithin=` If a document class providing `\chapter` is used (e.g. report or book), usually the `figure` and `table` caption counters are dependent on the `chapter` counter. Otherwise usually the `figure` and `table` counters are continuous throughout the document. This can be changed with the options

```
figurewithin=  chapter or section or none  and
 tablewithin=  chapter or section or none  .
```

If “`none`” is given as value, the numbering of figures resp. tables will be continuous throughout the document. Otherwise the numbering will be dependent on the given counter, e.g. `figurewithin=section` gives figures a section-dependent numbering scheme.

The name and numbering is internally done by the commands `\⟨type⟩name` & `\the⟨type⟩`, e.g. `\figurename` & `\thefigure` for figures. While usually using the above options are sufficient, sometimes a direct re-definition of these two macros is necessary. For example:

```
\captionsetup{figurewithin=section}
\renewcommand\thefigure{\arabic{section}\alph{figure}}
```

gives you a section-dependent numbering scheme ⟨*section*⟩⟨*figure*⟩ with a numerical section counter and a alphanumerical figure counter, e.g. “`Figure 2b`”.

2.9 Types

type= The `\caption` command can typeset captions for different types, e.g. `figure` and `table`. If you try to use the `\caption` command outside these environments you will get an error message, because it does not know what kind of caption do you want to have here. But in such situations you can set the caption type manually with

`type=`⟨*float type*⟩

prior to the usage of the `\caption` command (and other commands like `\ContinuedFloat`, or `\subcaptionbox` offered by the subcaption package, or `\subfloat` offered by the subfig package[20]), for example within an non-floating environment like `minipage`:

```
\noindent\begin{minipage}{\textwidth}
  \captionsetup{type=figure}
  \subfloat{...}
  ...
  \caption{...}
\end{minipage}
```

There is also a starred variant of this option, `type*=`⟨*float type*⟩, which behaves different than `type=`⟨*float type*⟩ if the hyperref package[10] is loaded: While `type=` sets an hyperlink anchor (if `hypcap=true` is set), `type*=` does not. (See also section 6.5: *hyperref*)

Note: Please don't re-define the internal macro `\@captype` for yourself, like suggested by some documentations, always use `\captionsetup{type=...}` instead.

Own float types can be defined with `\DeclareCaptionType` offered by this package, `\newfloat` offered by the float package[6], or `\DeclareNewFloatType` offered by the floatrow package[6].

Please note that you should use the option `type=` only *inside* boxes or environments (like `\parbox` or `minipage`), at best one where no page break could happen between contents and caption. Furthermore some visual side-effects (e.g. mixed-up figure and table settings regarding captions) could occur without using a box or environment, therefore a warning message will be issued if you try to do so.[6]

[6] You only get this warning message if you use ε-TEX as underlying TEX engine.

3 Commands

3.1 Typesetting captions

`\caption` The command

`\caption[`⟨*list entry*⟩`]{`⟨*heading*⟩`}`

typesets the caption inside a floating environment like `figure` or `table`. Well, you already know this, but the caption package offers an extension: If you leave the argument ⟨*list entry*⟩ empty, no entry in the list of figures or tables will be made. For example:

```
\caption[]{A figure without list entry.}
```

Please remember that the ⟨*heading*⟩ is a so-called *moving* argument, if no ⟨*list entry*⟩ has been given. But if a ⟨*list entry*⟩ is given, this argument is moving instead. "*Moving* argument" means that the argument will be written to the list-of file, make it appearing in the "List of Figures" resp. "List of Tables", too. *Moving* arguments are not allowed to contain *fragile* commands, everything must be *robust*, otherwise the argument could get *broken*, resulting in strange errors at the next LaTeX run. Some *fragile* commands could be *protected* by a leading `\protect`, own definitions could get defined with `\DeclareRobustCommand` instead of `\newcommand` to make them *robust*.

An example: `\caption{${}^{137}_{\phantom{1}55}$Cs}` will cause errors since `\phantom` is *fragile*. So we have either have to use the optional argument ⟨*list entry*⟩ (e.g. `\caption[${}^{137}_{55}$Cs]{${}^{137}_{\phantom{1}55}$Cs}`) or add `\protect` to prevent the `\phantom` command from getting *broken*: `\caption{${}^{137}_{\protect\phantom{1}55}$Cs}`.

But sometimes even this is not sufficient. The reason behind is the so-called single-line-check: It puts the ⟨*heading*⟩ into a horizontal box to determine the width of the caption, and this could cause error messages, too. An example: `\caption{A scheme. \[V_{C} \sim \left \{ \begin{array}{cc} E_{g} & \textrm{p-n} \\ e\phi_{B} & \textrm{M-S} \end{array} \right. \]}`. Using `\caption[A scheme]{...}` is not sufficient here, it's still leading to errors. ("`Missing $ inserted.`") So we have to put a `\captionsetup{singlelinecheck=off}` just in front of the `\caption` command additionally.

For more information about *moving* arguments and *fragile* & *robust* commands, take a closer look at your LaTeX manual or visit http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/teTeX/latex/latex2e-html/fragile.html.

`\caption*` The longtable package defines the command `\caption*` which typesets the caption without label and without entry in the list of tables. An example:

```
\begin{longtable}{...}
  \caption*{A table}\\
  ...
\end{longtable}
```

looks like

A table

	x	y
a	1	2
b	3	4

The caption package offers this feature, too, so you can use this command now within every floating environment like `figure` or `table`, like:

```
\begin{table}
  \caption*{A table}
  ...
\end{table}
```

\captionof
\captionof*

Sometimes you want to typeset a caption *outside* a floating environment, putting a figure within a non-floating `minipage` for instance. For this purpose the caption package offers the command

`\captionof{`⟨*float type*⟩`}[`⟨*list entry*⟩`]{`⟨*heading*⟩`}` .

Note that the first argument, the ⟨*float type*⟩, is mandatory here, because the `\captionof` command needs to know which name to put into the caption label (e.g. “Figure” or “Table”) and in which list to put the contents entry. An example:

```
\captionof{table}{A table}
```

typesets captions like this:

Table 1: A table

The star variant `\captionof*` has the same behavior as the `\caption*` command: It typesets the caption without label and without entry to the list of figures or tables.

Since `\captionof` uses the option `type` internally, the same restrictions as for the `type` option apply here, so you should use both `\captionof` and `\captionof*` only *inside* boxes or environments, too. (See section 2.9: *Types*)

\captionlistentry

Under certain circumstances it could be useful to make a list-of-figure (or table) entry on its own. This could be achieved with

`\captionlistentry[`⟨*float type*⟩`]{`⟨*list entry*⟩`}` .

One example: It’s quite easy to have a `longtable` with captions above the contents and a single list entry which points to the first page of the table:

```
\begin{longtable}{...}
  \caption{...}\\
\endfirsthead
  \caption[]{...}\\
\endhead
  ...
```

But since the longtable package does not offer an `\endfirstfoot` command, you cannot easily have captions *below* the table contents and a single list entry which points to the first page of the table. Here is were the `\captionlistentry` command could be used:

```
\begin{longtable}{...}
  \caption[]{...}\\
\endfoot
  \captionlistentry{...}
  ...
```

(Another example can be found in section 4.1: *Further examples*.)

There is also a starred variant, `\captionlistentry*`, which does not increment the ⟨*float type*⟩ counter. (Note that inside `longtable` environments `\captionlistentry` never increments the `table` counter. See also section 6.8: *longtable*.)

Please note that ⟨*list entry*⟩ is a *moving* argument, so everything it contains must be *robust*. (See also explanation of `\caption`)

3.2 Setting options

`\captionsetup` We already know the `\captionsetup` command (see section 1: *Using this package*), but this time we get enlighten about its optional argument ⟨*float type*⟩.

Remember, the syntax of this command is

`\captionsetup[`⟨*float type*⟩`]{`⟨*options*⟩`}` .

If a ⟨*float type*⟩ gets specified, all the ⟨*options*⟩ don't change anything at this time. Instead they only get marked for a later use, when a caption inside of a floating environment of the particular type ⟨*float type*⟩ gets typeset. For example

`\captionsetup[figure]{`⟨*options*⟩`}`

forces captions within a `figure` environment to use the given ⟨*options*⟩.

Here comes an example to illustrate this:

```
\captionsetup{font=small}
\captionsetup[figure]{labelfont=bf,textfont=it}
```

gives captions like this:

Figure 23: *A figure*

Table 2: A table

As you see the command `\captionsetup[figure]{...}` only changes the look of the `figure` caption labels, not touching the other ones.

As ⟨*float type*⟩ you can usually give one of these two only: `figure` or `table`. But as we will see later some LaTeX packages (like the floatrow, longtable, and sidecap package for example) and also this package offer additional environments with captions and these two commands can also be used with them. (See section 4: *Own enhancements* and section 6: *Package support*)

There is also a starred variant of `\captionsetup`:

`\captionsetup*[`⟨*float type*⟩`]{`⟨*options*⟩`}`

While the non-starred variant can give you warnings—for example if the ⟨*options*⟩ are actually not used throughout the document (e.g. a `\captionsetup[table]{font=sf}` without a `table`)—the starred variant will not.

`\clearcaptionsetup` If you want to get rid of these parameters marked for an automatic use within a particular environment you can use the command

`\clearcaptionsetup[`⟨*option*⟩`]{`⟨*float type*⟩`}` .

For example `\clearcaptionsetup{figure}` would clear all the extra handling for figures in the example above:

```
\captionsetup{font=small}
\captionsetup[figure]{labelfont=bf,textfont=it}
...
  \caption{A figure}
...
\clearcaptionsetup{figure}
...
  \caption{A figure}
...
```

Figure 24: *A figure*

Figure 25: A figure

If an optional argument ⟨*option*⟩ is given, only the settings regarding this particular ⟨*option*⟩ are cleared.[7] While the example above not only clears the options `labelfont=bf,textfont=it` for figures (but all options for figures instead), this one would only clear the `labelfont=bf` setting, leaving all other settings for figures intact:

```
\captionsetup{font=small}
\captionsetup[figure]{labelfont=bf,textfont=it}
...
  \caption{A figure}
...
\clearcaptionsetup[labelfont]{figure}
...
  \caption{A figure}
...
```

Figure 26: *A figure*

Figure 27: *A figure*

Analogous to `\captionsetup*` there is also a starred form `\clearcaptionsetup*` which suppresses warnings if the given ⟨*option*⟩ was not setup for the specified ⟨*float type*⟩.

`\showcaptionsetup` For debugging purposes the command

`\showcaptionsetup{`⟨*float type*⟩`}`

is offered. It generates a log file entry, showing the given options for the specified ⟨*float type*⟩. For example

```
\captionsetup[figure]{labelfont=bf,textfont=it}
\showcaptionsetup{figure}
```

[7] You can only specify *one* option here, not a list of options. If you want to clean more than one option, you need to use more than one `\clearcaptionsetup`.

gives the info:

```
Caption Info: Option list on 'figure'
Caption Data: {labelfont=bf,textfont=it} on input line 5.
```

3.3 Continued floats

\ContinuedFloat

Sometimes you want to split figures or tables without giving them their own reference number. This is what the command

```
\ContinuedFloat
```

is for; it should be used as first command inside the floating environment. It prevents the incrementation of the relevant counter (usually done by `\caption`), so a figure or table containing a `\ContinuedFloat` inside gets the same reference number as the figure or table before. An example:

```
\begin{table}
  \caption{A table}
  ...
\end{table}
...
\begin{table}\ContinuedFloat
  \caption{A table (cont.)}
  ...
\end{table}
```

gives the following result:

Table 3: A table

...

Table 3: A table (cont.)

...

Furthermore the `\ContinuedFloat` command executes options associated with the type name "`ContinuedFloat`". For example this can be used to switch to a different label format for continued figures or tables, as shown here:

```
\DeclareCaptionLabelFormat{continued}{Continued #1~#2}
\captionsetup[ContinuedFloat]{labelformat=continued}
...
\begin{table}\ContinuedFloat
  \caption{A table}
  ...
\end{table}
```

Continued Table 3: A table

...

(See section 4: *Own enhancements* for an explanation of `\DeclareCaptionLabelFormat`.)

There is also a LaTeX counter called `ContinuedFloat` which could be used for own purposes. For ordinary (floating) environments it's set to zero, to one for the first continued float, to two inside the second one, and so on. So every `\ContinuedFloat` increments this counter and a floating environment without `\ContinuedFloat` command resets this counter to zero. An example:

```
\DeclareCaptionLabelFormat{cont}{#1~#2\alph{ContinuedFloat}}
\captionsetup[ContinuedFloat]{labelformat=cont}
...
\begin{table}\ContinuedFloat
  \caption{A table}
  ...
\end{table}
```

Table 3c: A table

...

A reference to this table would still result in the output "Table 3" since only the caption label format was changed. If you would like to use the `ContinuedFloat` counter for the references, too, you could redefine the command `\theContinuedFloat` instead, which will be appended to the figure or table counter automatically in continued floats and is preset to be empty.

```
\renewcommand\theContinuedFloat{\alph{ContinuedFloat}}
...
\begin{table}\ContinuedFloat
  \caption{A table}
  ...
\end{table}
```

Table 3d: A table

...

A reference to that table would result in the output "Table 3d".

`\ContinuedFloat*`

Suppose you want to start the first figure or table of such a series with a kind of "Figure 7a" and not with "Figure 7" (and the second one with "Figure 7b" instead of "Figure 7a"). This is possible, too, by using the starred variant `\ContinuedFloat*` which—just like `\ContinuedFloat`—executes options associated with the type name "`ContinuedFloat`" and increments the LaTeX counter `ContinuedFloat`, *but* marks the first figure or table of a series instead:

```
\renewcommand\theContinuedFloat{\alph{ContinuedFloat}}
...
\begin{figure}\ContinuedFloat*
  ...
  \caption{First figure of a series}
\end{figure}
...
\begin{figure}\ContinuedFloat
  ...
  \caption{Second figure of a series}
```

```
\end{figure}
...
\begin{figure}\ContinuedFloat
  ...
  \caption{Third figure of a series}
\end{figure}
```

...

Figure 28a: First figure of a series

...

Figure 28b: Second figure of a series

...

Figure 28c: Third figure of a series

Note: Unfortunately `\ContinuedFloat*` is not available if the subfig package[20] is loaded.

A note about longtables

If you want to have a different caption label in `longtables` (offered by the longtable package[13]) after a page break, this can *not* be archived by using `\ContinuedFloat`, but instead you could write something like:

```
\DeclareCaptionLabelFormat{continued}{Continued #1~#2}
...
\begin{longtable}{...}
  \caption{A longtable}\\
\endfirsthead
  \captionsetup{labelformat=continued}
  \caption[]{A longtable}\\
\endhead
  ...
\end{longtable}
```

4 Own enhancements

A family of commands is provided to allow users to define their own formats etc. This enables information on separators, justification, fonts, and styles to be associated with a name and kept in one place. (These commands need to appear in the document preamble, this is the part between `\documentclass` and `\begin{document}`.)

`\DeclareCaption-Format`

You can define your own caption formats using the command

`\DeclareCaptionFormat{`⟨*name*⟩`}{`⟨*code using #1, #2 and #3*⟩`}` .

At usage the system replaces #1 with the caption label, #2 with the separator and #3 with the text. So the standard format `plain` is pre-defined by the caption package as

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

There is also a starred variant, `\DeclareCaptionFormat*`, which causes the code being typeset in TEXs vertical (instead of horizontal) mode, but does not support the `indention=` option.

`\DeclareCaption-LabelFormat`

Likewise you can define your own caption label formats:

`\DeclareCaptionLabelFormat{`⟨*name*⟩`}{`⟨*code using #1 and #2*⟩`}`

At usage #1 gets replaced with the name (e.g. "figure") and #2 gets replaced with the reference number (e.g. "12"). An example:

```
\DeclareCaptionLabelFormat{bf-parens}{(\textbf{#2})}
\captionsetup{labelformat=bf-parens,labelsep=quad}
```

(**29**) White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

`\bothIfFirst` `\bothIfSecond`

If you define your own caption label formats and use the subcaption or subfig[20] package, you should take care of empty caption label names. For this purpose the commands

`\bothIfFirst{`⟨*first arg*⟩`}{`⟨*second arg*⟩`}` and
`\bothIfSecond{`⟨*first arg*⟩`}{`⟨*second arg*⟩`}`

are offered. `\bothIfFirst` tests if the first argument exists (means: is not empty), `\bothIfSecond` tests if the second argument exists. If yes, both arguments get typeset, otherwise none of them.

For example the standard label format `simple` is *not* defined as

```
\DeclareCaptionLabelFormat{simple}{#1~#2} ,
```

because this could cause an extra space if #1 is empty. Instead `simple` is defined as

```
\DeclareCaptionLabelFormat{simple}%
                    {\bothIfFirst{#1}{~}#2} ,
```

causing the space to appear only if the label name is present.

`\DeclareCaption-TextFormat`

Likewise you can define your own caption text formats:

`\DeclareCaptionTextFormat{`⟨*name*⟩`}{`⟨*code using #1*⟩`}`

At usage #1 gets replaced with the caption text.

\DeclareCaption-LabelSeparator

You can define your own caption label separators with

```
\DeclareCaptionLabelSeparator{⟨name⟩}{⟨code⟩}
```

.

Again an easy example taken from the caption package itself:

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

There is also a starred variant, `\DeclareCaptionLabelSeparator*`, which causes the code being typeset without using the `labelfont=` setting. So for example the label separators `quad`, `newline`, and `endash` are defined in this way.

\DeclareCaption-Justification

You can define your own caption justifications with

```
\DeclareCaptionJustification{⟨name⟩}{⟨code⟩}
```

.

The ⟨*code*⟩ simply gets typeset just before the caption. E.g. using the justification `raggedright`, which is defined as

```
\DeclareCaptionJustification{raggedright}{\raggedright}
```

,

typesets captions with all lines moved to the left margin.

\DeclareCaption-Font

You can define your own caption fonts with

```
\DeclareCaptionFont{⟨name⟩}{⟨code⟩}
```

.

For example this package defines the options `small` and `bf` as

```
\DeclareCaptionFont{small}{\small}
```

and

```
\DeclareCaptionFont{bf}{\bfseries}
```

.

An example which brings color into life:

```
\usepackage{color}
\DeclareCaptionFont{red}{\color{red}}
\DeclareCaptionFont{green}{\color{green}}
\DeclareCaptionFont{blue}{\color{blue}}
\captionsetup{labelfont={blue,bf},textfont=green}
```

Figure 30: White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

But since the caption package already includes the tricky definition

```
\DeclareCaptionFont{color}{\color{#1}}
```

you can get the same result simply with

```
\usepackage{color}
\captionsetup{labelfont={color=blue,bf},
              textfont={color=green}}
```

.

\DeclareCaption-Style

You can define your own caption styles with

\DeclareCaptionStyle{⟨*name*⟩}[⟨*additional options*⟩]{⟨*options*⟩} .

Remember, caption styles are just a collection of suitable options, saved under a given name. You can wake up these options at any time with the option `style=`⟨*style name*⟩.

All caption styles are based on the `base` set of options. (See section 2.5: *Styles* for a complete list.) So you only need to specify options which are different to them.

If you specify ⟨*additional options*⟩ they get used in addition when the caption fits into a single line and this check was not disabled with the option `singlelinecheck=off`.

Again a very easy example taken from the core of this package: The caption style `base` is predefined as

```
\DeclareCaptionStyle{base}%
        [justification=centering,indention=0pt]{} .
```

Something more exciting:

```
\DeclareCaptionStyle{mystyle}%
        [margin=5mm,justification=centering]%
        {font=footnotesize,labelfont=sc,margin={10mm,0mm}}
\captionsetup{style=mystyle}
```

gives you caption like these ones:

FIGURE 31: A short caption.

FIGURE 32: A long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long caption.

\DeclareCaption-ListFormat

You can define your own caption list formats with

\DeclareCaptionListFormat{⟨*name*⟩}{⟨*code using #1 and #2*⟩} .

At usage #1 gets replaced with the label prefix (e.g. `\p@figure`), and #2 gets replaced with the reference number (e.g. `\thefigure`).

\DeclareCaptionType

You can define your own floating environments and caption types with

\DeclareCaptionType[⟨*options*⟩]{⟨*type*⟩}[⟨*name*⟩][⟨*list name*⟩]

where ⟨*options*⟩ are one or more of

`fileext=`⟨*file extension*⟩	(e.g. `fileext=lod`)
`placement=`⟨*combination of* `htbp`⟩	(e.g. `placement=tbp`)
`within=`⟨*"within" counter*⟩ *or* `none` .	(e.g. `within=section`)

If no ⟨*options*⟩ are given, "lo⟨*type*⟩" will be used as ⟨*file extension*⟩ for the list, "`tbp`" as ⟨*placement*⟩ specifier, and "`chapter`" resp. "`none`" as ⟨*"within" counter*⟩, i.e., the counter which resets the numbering. (If "`none`" was specified as value, the numbering of the new type will be continuous throughout the document.)

If no ⟨*name*⟩ is given, the name of the newly created type will be set to ⟨*type*⟩, but with an uppercase first letter. If no ⟨*list name*⟩ is given, the list name will be set to "List of ⟨*name*⟩s".

The list will be typeset using the command `\listof`⟨*type*⟩`s`, analogous to `\listoffigures` and `\listoftables`. If the fltpage package is loaded, an environment called `FP`⟨*type*⟩ will be defined, same for `sideways`⟨*type*⟩ (rotating package), `SC`⟨*type*⟩ (sidecap package), and `wrap`⟨*type*⟩ (wrapfig package).

So for example

```
\DeclareCaptionType{diagram}
```

will define a new caption type and floating environment called `diagram`, the list will be stored in a file with the extension `lodiagram`, the name (used for the caption) will be "Diagram" and the list name "List of Diagrams". The list could be typeset with `\listofdiagrams`. Dependent on which packages are loaded, the environments `FPdiagram`, `sidewaysdiagram`, `SCdiagram`, and `wrapdiagram` will be defined additionally.

4.1 Further examples

Example 1

If you would like to have a line break between caption label and text you could define it this way:

```
\DeclareCaptionFormat{myformat}{#1#2\\#3}
```

If you select this format with `\captionsetup{format=myformat}` you get captions like this:

Figure 33:
White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

You could even use an indention with this caption format:

```
\captionsetup{format=myformat,indention=1cm}
```

This would look like this:

Figure 34:
White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

Instead, you would like to have an indention only of the first line of the caption text? No problem, e.g.

```
\newlength\myindention
\DeclareCaptionFormat{myformat}%
                {#1#2\\\hspace*{\myindention}#3}
...
\setlength\myindention{1cm}
\captionsetup{format=myformat}
```

would give you captions like

Figure 35:
White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

But you want to have an caption option for this indention, so you can use it for example with `\captionsetup[figure]{myindention=...}`? This could be done, too:

```
\newlength\myindention
\DeclareCaptionOption{myindention}%
                {\setlength\myindention{#1}}
```

```
\DeclareCaptionFormat{myformat}%
                {#1#2\\\hspace*{\myindention} #3}
...
\captionsetup{format=myformat,myindention=1cm}
```

would give the same result as the example above.

Example 2

You want captions to look like this:

> White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.
>
> *(Figure 36)*

You could do it this way:

```
\DeclareCaptionFormat{reverse}{#3#2#1}
\DeclareCaptionLabelFormat{fullparens}%
                {(\bothIfFirst{#1}{~}#2)}
\DeclareCaptionLabelSeparator{fill}{\hfill}
\captionsetup{format=reverse,labelformat=fullparens,
              labelsep=fill,font=small,labelfont=it}
```

Example 3

The caption text should go into the left margin? A possible solution would be:

```
\DeclareCaptionFormat{llap}{\llap{#1#2}#3\par}
\captionsetup{format=llap,labelsep=quad,singlelinecheck=no}
```

As a result you would get captions like this:

Figure 37 White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

If the indention into the margin shall be fixed, you could use a combination of `\llap` and `\makebox`, for example:

```
\DeclareCaptionFormat{llapx}%
                {\llap{\makebox[5em][l]{#1}}#3\par}
\captionsetup{format=llapx,singlelinecheck=off}
```

Figure 38 White sand beaches. The pink smoothness of the conch shell. A sea abundant with possibilities. Duty-free shops filled with Europe's finest gifts and perfumes. Play your favorite game of golf amidst the tropical greens on one of the many championship courses.

Example 4

This example puts a figure aside a table, but uses only one common caption for both. This will be achieved by a combination of `\DeclareCaptionLabelFormat` and `\captionlistentry`:

```
\DeclareCaptionLabelFormat{andtable}%
                {#1~#2 \& \tablename~\thetable}
...
```

```
\begin{figure}
  \centering
  \includegraphics{...}%
  \qquad
  \begin{tabular}[b]{\ldots}
    ...
  \end{tabular}
  \captionlistentry[table]{...}
  \captionsetup{labelformat=andtable}
  \caption{...}
\end{figure}
```

	x	y
a	1	2
b	3	4

Figure 39 & Table 4: A figure and a table with a common caption [8]

(Please remember that `\captionlistentry` increments the figure resp. table counter.)

[8] The picture was taken with permission from the LaTeX Companion[1] examples.

5 Document classes & Babel support

This section will give you an overview of the document classes the caption supports, what do they already offer regarding captions, what side effects will occur when using the caption package with them, and what are the 'default' settings for them.

The 'default' settings depend on the document class you use; they represent how the class author wanted the captions to look like. So for example setting `format=default` can give you different visual designs when used with different document classes.

If you don't find the document class you use in this section, you usually don't have to worry: Many document classes (e.g. the octavo class) are derived from one of the standard document classes article, report, or book, and behave the same regarding captions. The caption package automatically does a compatibility check against the document class used and will give you the clear warning

```
Package caption Warning: Unsupported document class (or package) detected,
(caption)                usage of the caption package is not recommended.
See the caption package documentation for explanation.
```

if such an incompatibility was detected. If you don't get such warning everything is fine, but if you get it the usage of the caption package is not recommended and especially not supported.

If you get such a compatibility warning but decide to use the caption package anyway, you should watch carefully what side-effects occur, usually the look and feel of your captions will change by just including the caption package without options, meaning they do not look like as intended by the author of the document class. If this is fine for you, you should first specify the option `style=base` via `\usepackage[style=base]{caption}` or `\captionsetup{style=base}` to set the caption package into a well-defined state. Afterwards you can start setting your own options additionally and keep your fingers crossed.

5.1 Standard LaTeX: article, report, and book

Option	`default` value
`format=`	`plain`
`labelformat=`	`simple`
`labelsep=`	`colon`
`justification=`	`justified`
`font=`	*none*
`labelfont=`	*none*
`textfont=`	*none*

(This also applies to document classes derived from them.)

5.2 $\mathcal{AMS}$: amsart, amsproc, and amsbook

Option	`default` value
`format=`	`plain`
`labelformat=`	`simple`
`labelsep=`	`.\enspace`
`justification=`	`justified`
`font=`	`\@captionfont`
`labelfont=`	`\@captionheadfont`
`textfont=`	`\@captionfont\upshape`

(`\@captionfont` will be set to `\normalfont`, and `\@captionheadfont` to `\scshape` by the $\mathcal{AMS}$ document classes.)

Furthermore the margin will be set to `\captionindent` for more-than-one-line captions (which will be set to `3pc` by the $\mathcal{AMS}$ classes), the margin for single-line captions will be set to the

half of it instead. If you want to use a common margin for both, insert `\clearcaptionsetup[margin*]{singleline}` into the preamble of your document, after loading the caption package.

Additionally the options `figureposition=b,tableposition=t` will be set. You can override these settings by specifying other values for `figureposition=` or `tableposition=` in the option list while loading the caption package.

5.3 beamer

Option	`default` value
`format=`	`plain`
`labelformat=`	*not numbered*
`labelsep=`	`colon`
`justification=`	`raggedright`
`font=`	beamer "`caption`" *settings*
`labelfont=`	beamer "`caption name`" *settings*
`textfont=`	*none*

Build-in features, and side-effects

You can setup font and color settings with `\setbeamerfont{caption}{`⟨*options*⟩`}` and `\setbeamerfont{caption name}{`⟨*options*⟩`}`. This will still work, unless you set a different font with `\captionsetup{font=`⟨*options*⟩`}` or `\captionsetup{labelfont=`⟨*options*⟩`}`.

Furthermore the beamer classes offer different caption templates which can be chosen with `\setbeamertemplate{caption}[`⟨*template*⟩`]`. Since the caption package replaces this caption template mechanism, `\defbeamertemplate*{caption}{`⟨*template code*⟩`}` and `\setbeamertemplate{caption}[`⟨*template*⟩`]` will have no effect when the caption package is used. (Exception: Selecting the template `default`, `numbered`, or `caption name own line` will be recognized by the caption packageand be mapped to corresponding options.)

5.4 KOMA-Script: scrartcl, scrreprt, and scrbook

Option	`default` value
`format=`	*uses* `\setcapindent` *&* `\setcaphanging` *settings*
`labelformat=`	*like* `simple`*, but with support of "autodot"*
`labelsep=`	`\captionformat`
`justification=`	`justified`
`font=`	`\setkomafont{caption}` *settings*
`labelfont=`	`\setkomafont{captionlabel}` *settings*
`textfont=`	*none*

Build-in features

The KOMA-Script document classes offer many ways to customize the look and feel of the captions. For an overview and a full description please take a look at the KOMA-Script documentation, section 'Tables and Figures'.

Side effects

The optional argument of `\setcapwidth` is not supported and will be ignored if used in conjunction with the caption package. Furthermore the KOMA-Script options `tablecaption-`

above & tablecaptionbelow and the commands \captionabove & \captionbelow are stronger than the position= setting offered by the caption package.

5.5 NTG: artikel, rapport, and boek

Option	default value
format=	plain
labelformat=	simple
labelsep=	colon
justification=	justified
font=	*none*
labelfont=	\CaptionLabelFont
textfont=	\CaptionTextFont

Build-in features, and side-effects

\CaptionLabelFont and \CaptionTextFont can be set either directly or by using \CaptionFonts. Both is still supported unless you use one of the two options labelfont= or textfont= offered by the caption package.

5.6 SMF: smfart and smfbook

Since the SMF document classes are derived from the $\mathcal{AMS}$ document classes the same default values are valid here.

Additionally the margin is limited up to the tenth of the \linewidth. If you don't like this limitation, you can switch it off with the option maxmargin=off or maxmargin=false (which both means the same).

5.7 thesis

Option	default value
format=	hang
labelformat=	*like* simple*, but with short name*
labelsep=	colon
justification=	justified
font=	*none*
labelfont=	\captionheaderfont
textfont=	\captionbodyfont

Build-in features, and side-effects

The caption label font can be set with \captionheaderfont, the caption text font with \captionbodyfont. Both is still supported unless you use one of the two options labelfont= or textfont= offered by the caption package.

5.8 frenchb babel option

If you use the frenchb option of the babel package with one of the three standard LaTeX classes (or a one derived from them) the default labelsep= will be set to \CaptionSeparator (offered by frenchb), overriding the default value set by the document class. So redefining \CaptionSeparator will still work, unless you don't select a different labelsep= than the default one.

Please load the caption package *after* the babel package.

5.9 frenchle and frenchpro packages

If you use the frenchle or frenchpro package, the default `labelsep=` will be set to `\captionseparator` (offered by frenchle/pro) plus `\space`, overriding the default value set by the document class. So redefining `\captionseparator` will still work, unless you don't select a different `labelsep=` than the default one.

Furthermore the default `textfont=` will be set to `textfont=it`, since this emulates the default setting of `\captionfont` defined by the frenchle or frenchpro package. Please note that the command `\captionfont` is used by the caption package internally for a different purpose, so you should not change it (anymore).

The command `\unnumberedcaptions{`⟨*figure* or *table*⟩`}` will still work, but only unless you don't select a different `labelformat=` than the default one.

Please load the caption package *after* the frenchle or frenchpro package.

6 Package support

The caption package was adapted to the following packages which deals with captions, too:

> float, floatflt, fltpage, hyperref, hypcap, listings, longtable, picinpar, picins, rotating, setspace, sidecap, subfigure, supertabular, threeparttable, wrapfig, and xtab

Furthermore the floatrow package[8], the subcaption package (which is part of the caption package bundle), and the subfig package[20] support the caption package and use its `\captionsetup` interface.

If a package (or document class) unknown to the caption package redefines the `\caption` command as well, this redefinition will be preferred over the one this package offers, providing maximum compatibility and avoiding conflicts. If such a potential incompatibility is detected, you will see this warning message: [9]

```
Package caption Warning: \caption will not be redefined since it's already
(caption)                redefined by a document class or package which is
(caption)                unknown to the caption package.
See the caption package documentation for explanation.
```

As a result, the following features offered by the caption package will not be available:

- the options `labelformat=`, `position=auto`, `list=`, and `listformat=`
- `\caption*` (to produce a caption without label)
- `\caption[]{...}` (to produce no entry in the List of Figures or Tables)
- `\caption{}` (to produce an empty caption without label separator)
- `\ContinuedFloat`
- correctly justified captions in environments like `wide` and `addmargin` which add extra margins
- the hypcap feature (See section 6.5: *hyperref*)
- the sub-caption feature (See subcaption package documentation)

`compatibility=`

You can override this compatibility mode by specifying the option

```
compatibility=false
```

when loading the caption package. But please note that using this option is neither recommended nor supported since unwanted side-effects or even errors could occur afterwards. (For that reason you will get a warning about this.)

[9] You can suppress this warning by specifying the option `compatibility=true` when loading the caption package.

6.1 algorithms

The algorithms package bundle[5] provides two environments: The `algorithmic` environment provides a possibility for describing algorithms, and the `algorithm` environment provides a "float" wrapper for algorithms.

Since the `algorithm` environment is implemented via `\newfloat` provided by the float package[6], please see section 6.2: *float*.

6.2 float

The float package[6] introduces the commands `\restylefloat` to give existing floating environments a new look & feel and `\newfloat` to define new floating environments. It also provides the "H" float placement option which places the environments "here" instead of letting them floating around.

For floating environments defined with `\newfloat` or `\restylefloat` the `position` option has no effect on the main caption anymore, since its placement and spacing will be controlled by the selected float style instead.

A caption style and options defined with the name of the float style will be executed additionally to the regular ones. Using this mechanism the caption package emulates the default look & feel of the `ruled` captions: It defines the caption style

```
\DeclareCaptionStyle{ruled}%
       {labelfont=bf,labelsep=space,strut=off}  .
```

So to change this you need either define your own caption style called `ruled` or use `\captionsetup[ruled]{`⟨*options*⟩`}` to specify additional options.

Also by using this mechanism the skip between a `boxed` float and its caption is specified, overriding its global value:

```
\captionsetup[boxed]{skip=2pt}
```

For changing this, just use `\captionsetup[boxed]{skip=`⟨*value*⟩`}` with an appropriate value. Or if you want to use the global `skip` setting instead, you can remove the usage of the local setting for these floats with `\clearcaptionsetup[skip]{boxed}`.

Note: Only one single caption can be typeset inside environments defined with `\newfloat` or `\restylefloat`, furthermore these environments are not behaving exactly like the pre-defined floats `figure` and `table`. As a consequence many packages do not cooperate well with these. Furthermore the float package has some caveats & limitations, so if you just want to define a new simple floating environment—behaving like `figure` or `table`—I recommend using `\DeclareCaptionType` instead. And for defining non-simple floating environments and customization I recommend using `\DeclareNewFloatType` offered by the floatrow package[8].

6.3 floatflt

The floatflt package[7] offers figures and tables which do not span the full width of a page and are float around by text.

If you want to setup special options for the `floatingfigure` and `floatingtable` environments you can use

```
\captionsetup[floatingfigure]{⟨options⟩}   and
\captionsetup[floatingtable]{⟨options⟩}   .
```

These options will be executed additionally to the regular ones for `figure` resp. `table`.

Note: The `margin` resp. `width` setting will not be used for these figures resp. tables, unless you set it explicit with `\captionsetup[floatingfigure]{...}` resp. `\captionsetup[floatingtable]{...}`.

6.4 fltpage

The fltpage package[9] offers the outhouse of the caption for `figures` or `tables` which needs the whole page for its contents. This will be done by placing the caption on the bottom of the previous or next page.

Two options control the links to the environments `FPfigure` and `FPtable`:

`FPlist=caption` or `FPlist=figure`

If set to `caption`, the list entry will link to the caption; if set to `figure`, it will link to the figure contents. (The default setting is `FPlist=caption`.)

`FPref=caption` or `FPref=figure`

If set to `caption`, a `\ref`, `\pageref`, `\autoref`, or `\nameref` will link to the caption; if set to `figure`, it will link to the figure contents. (The default setting is `FPref=figure`.)

If you want to setup special options for the `FPfigure` and `FPtable` environments you can use

`\captionsetup[FPfigure]{`⟨*options*⟩`}` *and*
`\captionsetup[FPtable]{`⟨*options*⟩`}` .

These options will be executed additionally to the regular ones for `figure` or `table`.

Furthermore `\DeclareCaptionType{`⟨*name*⟩`}` will also define an environment called `FP`⟨*name*⟩ which behaves like `FPfigure` and `FPtable`.

6.5 hyperref

The hyperref package[10] is used to handle cross referencing commands in LaTeX to produce hypertext links in the document.

There are two options available to control the placement of hyperlink anchors: [10]

`hypcap=true` or `hypcap=false`

If set to `true` all hyperlink anchors – where entries in the List of Figures, `\ref`, and `\autoref` will link to – are placed at the beginning of the (floating) environment.

If set to `false` the hyperlink anchor is (usually) placed at the caption.

(The default setting is `hypcap=true`.)

`hypcapspace=`⟨*amount*⟩

Because it looks poor if the hyperlink points exactly at the top of the figure or table, you can specify a vertical distance between the hyperlink anchor and the (floating) environment itself, e.g. `hypcapspace=0pt` removes this distance.

(The default setting is `hypcapspace=0.5\baselineskip`.)

Both settings have no effect in `lstlistings` (provided by the listings package), `longtables` (provided by the longtable package), `supertabulars` (provided by the supertabular package), and `xtabulars` (provided by the xtab package), within these environments hyperlink anchors will always be placed as if `hypcap=true` and `hypcapspace=0pt` would be set.

Please note:

`\captionof{`⟨*type*⟩`}{...}` vs. `\captionsetup{type=`⟨*type*⟩`}`+`\caption{...}`

Without hyperref loaded, both will give you identical results. But with hyperref loaded, and with `hypcap=true` requested, the hyperlink anchor will be placed different. For example:

```
\begin{minipage}{\linewidth}
  ...
  \captionof{figure}{A figure}
\end{minipage}
```

[10] These options are named after the hypcap package which they supersede.

will place the hyperlink anchor at the caption. (And if `hypcap=true` is set, you will get a warning about this.) But

```
\begin{minipage}{\linewidth}
  \captionsetup{type=figure}
  ...
  \caption{A figure}
\end{minipage}
```

will place the hyperlink anchor at the beginning of the `minipage`, since `\captionsetup{type=figure}` does not only set the caption type to "figure" but does place a hyperlink anchor, too.

`\caption[]{...}` vs. `\captionsetup{list=false}+\caption{...}`

Again, without hyperref loaded, both will give you identical results. But with hyperref loaded, the difference is in the nuances. So for example the optional argument of `\caption` will also be written to the `aux` file, and will be used by the `\nameref` command. So if you choose to use `\caption` with empty optional argument, `\nameref` will also give you an empty result. So it's better to use `\captionsetup{list=false}` if you don't want an entry in the List of Figures or List of Tables.

6.6 hypcap

The hypcap package[11] offers a solution to the problem that links to a float using hyperref may anchors to the caption rather than the beginning of the float. Since the caption package *v*3.1 already solves this problem for itself, the hypcap package is usually not needed.

If the hypcap package is loaded additionally to the hyperref package, it takes over the control of the hyperlink anchor placement from the caption package, overriding the options `hypcap=` and `hypcapspace=`.

So for a manual placement of hyperlink anchors `\captionsetup{type=`⟨*type*⟩`}` is not sufficient anymore, instead you need to use `\capstart` (provided by the hypcap package) for this.

Regarding the automatically placement the hypcap package offers good placement of hyperlink anchors for the floating environments `figure` and `table` only. In contrast the `hypcap=true` option of the caption package also offers good placements of hyperlink anchors for `floatingfigure`s (provided by the floatflt package), `FPfigure`s & `FPtable`s (provided by the fltpage package), `figwindow`s (provided by the picinpar package), `parpic`s (provided by the picins package), `SCfigure`s (provided by the sidecap package), `threeparttable`s (provided by the threeparttable package), and `wrapfigure`s (provided by the wrapfig package).

6.7 listings

The listings package[12] typesets programming code.

If you want to setup special options for the `lstlisting` environment you can use

```
\captionsetup[lstlisting]{⟨options⟩}   .
```

Please note that the listings package has its very own options for controlling the position and the skips of the caption: `captionpos=`, `abovecaptionskip=`, and `belowcaptionskip=`. (See listings documentation for details.) These listings options override the caption's ones, but can be again overwritten by `\captionsetup[lstlisting]{...}`, e.g.

```
\captionsetup[lstlisting]{skip=10pt}   .
```

6.8 longtable

The longtable package[13] offers an environment which behaves similar to the `tabular` environment, but the table itself can span multiple pages.

If you want to setup special options for the `longtable` environment you can use

`\captionsetup[longtable]{`⟨*options*⟩`}` .

These options will be executed additionally to the regular ones for `table`.

The `margin` and `width` settings usually override `\LTcapwidth`, so you get an equal look & feel of the captions in `table`s and `longtable`s. But if you set `\LTcapwidth` to a value different than its default = `4in`, the caption package will follow that. (But `\LTcapwidth` will be overwritten by `\captionsetup[longtable]{width=`⟨*value*⟩`}`, even if it is set to a value different than `4in`.)

Note: `\captionof` and `\ContinuedFloat` do *not* work for `longtable`s. Furthermore neither `\caption` nor `\captionlistentry` will increment the `table` counter here; it's incremented by the `longtable` environment instead. If you need a `longtable` which does not increment the `table` counter please use the `longtable*` environment (offered by the ltcaption package which is part of the caption package bundle and will be loaded automatically).

6.9 picinpar

Similar to the floatflt package the picinpar package[14] offers figures and tables which do not span the full width of a page and are float around by text. For a detailed discussion about the differences between these packages please take a look at The LaTeX Companion[1].

If you want to setup special options for the `figwindow` and `tabwindow` environments you can use

`\captionsetup[figwindow]{`⟨*options*⟩`}` *and*
`\captionsetup[tabwindow]{`⟨*options*⟩`}` .

These options will be executed additionally to the regular ones for `figure` or `table`.

Note: The `margin` resp. `width` setting will not be used for these figures and tables, unless you set it explicit with `\captionsetup[figwindow]{...}` or `\captionsetup[figtable]{...}`.

6.10 picins

Similar to the floatflt and picinpar package the picins package[15] offers figures and tables which do not span the full width of a page and are float around by text. For a detailed discussion about the differences between these packages please take a look at The LaTeX Companion[1].

If you want to setup special options for the `parpic` environment you can use

`\captionsetup[parpic]{`⟨*options*⟩`}` .

These options will be executed additionally to the regular ones for `figure` or `table`.

Furthermore `\piccaption[]{...}` produce no entry in the List of Figures, and `\piccaption*{...}` gives an unlabeled & unnumbered caption.

Note: The `margin` resp. `width` setting will not be used for these figures and tables. But you can override this by setting it explicit with `\captionsetup[parpic]{...}`.

If you want to have a `\piccaption` of another type as `figure`, please do *not* re-define `\@captype` as suggested by the picins documentation. Please use the `\piccaptiontype{`⟨*type*⟩`}` command which is offered by the caption package instead, e.g.:

```
\piccaptiontype{table}
\piccaption{An example table}
\parpic(50mm,10mm)[s]{...}
```

6.11 rotating

The rotating package[16] offers the floating environments `sidewaysfigure` and `sidewaystable` which are just like normal figures and tables but rotated by 90 resp. 270 degree. Furthermore a command `\rotcaption` is offered for rotated captions only.

The command `\rotcaption` will be extended by the caption package, so `\rotcaption*` and `\rotcaptionof` can be used analogous to `\caption*` and `\captionof`.

Furthermore `\DeclareCaptionType{`⟨*name*⟩`}` will also define an environment called `sideways`⟨*name*⟩ which behaves like `sidewaysfigure` and `sidewaystable`.

6.12 setspace

The setspace package[17] offers options and commands to change the spacing, e.g. `\usepackage[onehalfspacing]{setspace}` causes the document to be typeset in one-and-a-half spacing.

If the setspace package is used in conjunction with the caption package, the caption will be typeset with single spacing as default. This can be changed by specifying either `font=onehalfspacing`, `font=doublespacing`, or `font={stretch=`⟨*amount*⟩`}`. (See also section 2.3: *Fonts*)

6.13 sidecap

The sidecap package[18] offers the floating environments `SCfigure` and `SCtable` which are like usual figures and tables but the caption will be put *beside* the contents.

If you want to setup special options for the `SCfigure` and `SCtable` environments you can use

`\captionsetup[SCfigure]{`⟨*options*⟩`}` *and*
`\captionsetup[SCtable]{`⟨*options*⟩`}` .

These options will be executed additionally to the regular ones for `figure` or `table`.

Furthermore `\DeclareCaptionType{`⟨*name*⟩`}` will also define an environment called `SC`⟨*name*⟩ which behaves like `SCfigure` and `SCtable`.

Note: The sidecap package offers its own options for justification. If set, they will override the one specified with the caption option `justification=` for captions beside their contents.

Note: The `margin` resp. `width` setting will not be used for these figures and tables, unless you set it explicit with `\captionsetup[SCfigure]{...}` or `\captionsetup[SCtable]{...}`.

(Instead of using the sidecap package you can also use the more powerful and flexible floatrow package[8] for typesetting captions beside the contents.)

Undocumented features

The sidecap package *v*1.6 has some undocumented package options and commands which allow further customization of the side-captions:

`margincaption` The package option

`margincaption` (e.g. `\usepackage[margincaption]{sidecap}`)

causes all side-captions to be placed into the margin.

\sidecaptionvpos The command

\sidecaptionvpos{⟨*float type*⟩}{⟨*pos*⟩}

sets the vertical position of the side-caption. ⟨*pos*⟩ can be either 't' (for top alignment), 'b' (for bottom alignment), or 'c' (for center alignment). The default setting for table is 't', for figure and all other ones defined with \DeclareCaptionType it's 'b'.

6.14 subfigure

The subfigure package[19] provides support for small or 'sub' figures and tables within a single figure or table environment. This package is obsolete, new users should use subfig instead.

Since the subfigure package is obsolete it will only be supported in a way that old documents (which have used the caption package *v*1.*x* during development) should still compile as expected.

Please use the subfig or subcaption package instead which both supports the caption package *v*3.*x*.

(See the subfig package[20] documentation for details.)

6.15 supertabular and xtab

The supertabular[21] and xtab[24] packages offer an environment which can span multiple pages and is quite similar to the longtable environment provided by the longtable package[13]. For a detailed discussion about the differences between these powerful packages please take a look at The LaTeX Companion[1].

If you want to setup special options for the supertabular resp. xtabular environment you can use

\captionsetup[supertabular]{⟨*options*⟩} *resp.*
\captionsetup[xtabular]{⟨*options*⟩} .

These options will be executed additionally to the regular ones for table.

Note: \ContinuedFloat does not work for supertabulars and xtabulars.

6.16 threeparttable

The threeparttable package[22] provides a scheme for tables that have a structured note section after the table contents and the caption. This scheme provides an answer to the old problem of putting footnotes in tables—by making footnotes entirely unnecessary.

If you want to setup special options for the threeparttable and measuredfigure environments you can use

\captionsetup[threeparttable]{⟨*options*⟩} *and*
\captionsetup[measuredfigure]{⟨*options*⟩} .

These options will be executed additionally to the regular ones for table or figure.

Note: Since the caption will have the same width as the contents here, the margin resp. width setting will not be used for these figures and tables, at least unless you set it explicit with \captionsetup[threeparttable]{...} or \captionsetup[measuredfigure]{...}.

(Instead of using the threeparttable package you can also use the floatrow package[8] for this purpose.)

6.17 wrapfig

Similar to the floatflt, the picinpar, and the picins package the wrapfig package[23] offers figures and tables which do not span the full width of a page and are float around by text. For a detailed discussion about the differences between these packages please take a look at The LaTeX Companion[1].

If you want to setup special options for the `wrapfigure` and `wraptable` environments you can use

`\captionsetup[wrapfigure]{`⟨*options*⟩`}` *and*
`\captionsetup[wraptable]{`⟨*options*⟩`}` .

These options will be executed additionally to the regular ones for `figure` or `table`.

Furthermore `\DeclareCaptionType{`⟨*name*⟩`}` will also define an environment called `wrap`⟨*name*⟩ which behaves like `wrapfigure` and `wraptable`.

Note: The `margin` resp. `width` setting will not be used for these figures and tables, unless you set it explicit with `\captionsetup[wrapfigure]{...}` or `\captionsetup[wraptable]{...}`.

Note: If the wrapfig package should be used in coorperation with the float package, it needs to be loaded *after* the float package. Furthermore `\newfloat` and `\restylefloat` commands *should* be placed after *both* packages have been loaded.

7 Further reading

I recommend the following documents for further reading:

- The TEX FAQ – Frequently asked questions about TEX and LATEX:

  http://faq.tug.org/

- A French FAQ can be found at

  http://www.grappa.univ-lille3.fr/FAQ-LaTeX/

- "What is a minimal working example?" from Christian Faulhammer and Ulrich Schwarz:

  http://www.minimalbeispiel.de/mini-en.html

- epslatex from Keith Reckdahl contains many tips around including graphics in LATEX 2ε documents. You will find this document in the directory

  ftp://tug.ctan.org/pub/tex/info/epslatex/

8 Thanks

I would like to thank Katja Melzner, Steven D. Cochran, Frank Mittelbach, Olga Lapko, David Carlisle, Carsten Heinz, Keith Reckdahl, and Markus Kohm. Thanks a lot for all your help, ideas, patience, spirit, and support!

Also I would like to thank Harald Harders, Peter Löffler, Peng Yu, Alexander Zimmermann, Matthias Pospiech, Jürgen Wieferink, Christoph Bartoschek, Uwe Stöhr, Ralf Stubner, Geoff Vallis, Florian Keiler, Jürgen Göbel, Uwe Siart, Sang-Heon Shim, Henrik Lundell, David Byers, William Asquith, Prof. Dr. Dirk Hoffmann, Frank Martini, Danie Els, Philipp Woock, and Fadi Semmo who all helped to make this package a better one.

A Alphabetical Reference

A.1 Options

Option	Short description	Section
aboveskip	sets the skip above caption	2.6
belowskip	sets the skip below caption	2.6
compatibility †	force (non-)compatibility	6
figurename †	sets the figure name	2.8
figureposition †	gives a hint about the figure caption position	2.6
figurewithin †	sets the figure “within” counter	2.8
font(+)	sets the font	2.3
format	sets the format	2.1
FPlist	To where the list entry of a FPfigure should link?	6.4
FPref	To where a \ref to a FPfigure should link?	6.4
hangindent	sets the hang indention	2.4
hypcap	selects ‘hypcap’ feature	6.5
hypcapspace	sets the distance between hyperlink and contents	6.5
indention	sets the indention	2.4
justification	sets the justification	2.2
labelfont(+)	sets the font of the caption label	2.3
labelformat	sets the format of the caption label	2.1
labelsep	sets the label separator	2.1
labelseparator	–same as labelsep–	2.1
list	switches the entries in the List on or off	2.7
listfigurename	sets the ‘List of Figure’ name	2.8
listformat	sets the ‘List of Figure/Table’ entry format	2.7
listtablename	sets the ‘List of Tables’ name	2.8
margin	sets the margin	2.4
margin*	sets the margin, but only if no width is set	2.4
maxmargin	sets the max. margin	2.4
minmargin	sets the min. margin	2.4
name	sets the name of the current environment	2.8
oneside	selects the one-side mode	2.4
options	executes the given option list	
parindent	sets the paragraph indention	2.4
parskip	sets the skip between paragraphs	2.4
position	gives a hint about the caption position	2.6
singlelinecheck	switches the single-line-check on or off	2.2
skip	sets the skip between content and caption	2.6
strut	switches the usage of \struts on or off	2.1
style	sets the caption style	2.5
subtype	sets the sub-caption type	– [11]
tablename †	sets the table name	2.8
tableposition †	gives a hint about the table caption position	2.6
tablewithin †	sets the table “within” counter	2.8
textfont(+)	sets the font of the caption text	2.3
textformat	sets the format of the caption text	2.1
twoside	selects the two-side mode	2.4
type	sets the caption type & places a hyperlink anchor	2.9
type*	sets the caption type only	2.9
width	sets a fixed caption width	2.4

Note: Obsolete options are not listed here. See section C.1: *caption v1.x* and section C.2: *caption2 v2.x* for a list of these options.

A.2 Commands

Command	Section
\abovecaptionskip	2.6
\belowcaptionskip	2.6
\caption	3.1
\caption*	3.1
\captionlistentry	3.1
\captionof	3.1
\captionof*	3.1
\captionsetup	3.2
\captionsetup*	3.2
\centerfirst	2.2
\centerlast	2.2
\clearcaptionsetup	3.2
\clearcaptionsetup*	3.2
\ContinuedFloat	3.3
\DeclareCaptionFont	4
\DeclareCaptionFormat	4
\DeclareCaptionFormat*	4
\DeclareCaptionJustification	4
\DeclareCaptionLabelFormat	4
\DeclareCaptionLabelSeparator	4
\DeclareCaptionLabelSeparator*	4
\DeclareCaptionListFormat	4
\DeclareCaptionOption	4
\DeclareCaptionStyle	4
\DeclareCaptionSubType	– [12]
\DeclareCaptionTextFormat	4
\DeclareCaptionType	4
\showcaptionsetup	3.2

[†] These options are only available in the preamble of the document.
[11] The option `subtype` is explained in the subcaption package documentation.
[12] `\DeclareCaptionSubType` is explained in the subcaption package documentation.

A.3 Warnings

`\caption outside box or environment.`
– or –
`\captionsetup{type=...} outside box or environment.`
– or –
`\captionsetup{type*=...} or \captionof outside box or environment.`

You have placed a `\caption`, `\captionof`, or `\captionsetup{type=`⟨*type*⟩`}` command outside an box, group, or environment. You should not do this since it could cause some bad side-effects.

(See section 2.9: *Types* and section 3.1: *Typesetting captions*)

`\caption will not be redefined since it's already redefined by a document class or package which is unknown to the caption package.`

If the caption package detects that some (unknown) document class or package has enhanced `\caption`, it will not redefine `\caption`, too, since this would simply kill the enhancement. As a result some features, like `\caption*`, `\ContinuedFloat`, using the optional argument of `\captionsetup`, or the options `list=` and `hypcap=` will not be available.

If you don't care about the original enhancements but would like to use the full range of features of the caption package instead you can give the unsupported(!) option `compatibility=false` a try and keep your fingers crossed. (But you will get the next warning instead.)

(See section 5: *Document classes & Babel support* and section 6: *Package support*)

`Forced redefinition of \caption since the unsupported(!) package option 'compatibility=false' was given.`

Since you were so keen to specify the option `compatibility=false` the caption package will try to do its best to fulfill your wishes. But depending on the document class or other packages you use that can end in non-functional features or even errors. So keep your fingers crossed!

(See section 6: *Package support*)

`Hyperref support is turned off because hyperref has stopped early.`

If the hyperref package stops early during loading (because of what-ever reason), the hyperref support of the caption package will not be available. As a result you could get hyperref warnings and non-functional hyperlinks to figures or tables.

(See section 6.5: *hyperref*)

`Ignoring optional argument [`⟨*pos*⟩`] of \setcapwidth.`

The caption package tries to emulate the KOMA-Script commands regarding captions as best as it can. But the optional argument of the KOMA-Script command `\setcapwidth` is not (yet) working if you use this package, so if you try to use it anyway, you will get this warning.

(See section 5.4: *KOMA-Script: scrartcl, scrreprt, and scrbook*)

`Internal Warning:` ⟨*warning message*⟩`.`

You should never see this warning, either you use a package which redefines `figure` or `table` and which is unknown to the caption package, or this is a bug in the caption package. Please send me an e-mail reporting this issue.

`\label before \caption ignored`

Regarding `\label` the floating environments behave differently than its non-floating counterparts: The internal reference will not be generated at the beginning of the environment, but at `\caption` instead. So you have to place the `\label` command either just *after* or *inside* the caption text (mandatory argument of `\caption`).

`Option '⟨option⟩' was not in list '⟨option list⟩'.`

If you try to remove a specific option of an option list, for example with `\clearcaptionsetup[position]{table}`, and this option can not be found inside the option list, you will get this warning. If this is not because of a typo and you would like to suppress this warning, use `\clearcaptionsetup*` instead of `\clearcaptionsetup`.

(See section 3.2: *Setting options*)

`Option list '⟨option list⟩' undefined.`

If you try to remove a specific option of an option list, for example with `\clearcaptionsetup[format]{figure}`, and this option list is not defined (yet), you will get this warning. If this is not because of a typo and you would like to suppress this warning, use `\clearcaptionsetup*` instead of `\clearcaptionsetup`.

(See section 3.2: *Setting options*)

`'ragged2e' package not loaded, therefore substituting \raggedright for \RaggedRight.`

The caption option `justification=RaggedRight` is only full functional if you have the ragged2e package installed in your TEX system. If not, you will get this warning, and `justification=RaggedRight` will be treated as `justification=raggedright`.

(See section 2.2: *Justification*)

`Obsolete option 'ignoreLTcapwidth' ignored.`

The caption2 package option `ignoreLTcapwidth` will not be emulated by this version of the caption package, usually you can simply wipe it away.

(See section C.2: *caption2 v2.x* and section 6.8: *longtable*)

`'ragged2e' support has been changed. Rerun to get captions right.`

The ragged2e package will only be loaded by the caption package if it is actually needed. At least two LATEX runs are needed for that, so on the first run you could get this warning. Just compile your document again and this warning should go away.

(See section 2.2: *Justification*)

`Reference on page ⟨page no.⟩ undefined.`

If you use a `twoside` page layout, the caption package needs to track the page numbers to get the margins right. At least two LATEX runs are needed for that, on the first run you could get this warning. Just compile your document again and this warning should go away.

(See section 2.4: *Margins and further paragraph options*)

`The caption type was already set to '⟨type⟩'.`

This warning informs you about mixed caption options. For example if you use a `\captionsetup{type=table}` or `\captionof{table}{...}` inside a `figure` environment, this would result in using both option sets for that specific caption, the one for `figure` (specified with `\captionsetup[figure]{...}`) and the one for `table` (specified with `\captionsetup[table]{...}`) as well.

(You can suppress this warning by using the starred form `\captionsetup*{type=...}`.)

(See section 3.2: *Setting options*)

`The option 'hypcap=true' will be ignored for this particular \caption.`

The caption package hasn't found a proper hyperlink anchor for this particular caption, so it decides to ignore the setting `hypcap=true` (which is set by default). As a result a link to this caption (e.g. in the List of Figures, or set by `\ref` or `\autoref`) will link you to the caption of the figure or table, not to the figure or table itself.

This can happen if you use `\captionof` inside a non-floating environment, but also if you use some package which redefines `figure` or `table`, and which is unknown to the caption package.

If this is ok for you but you want to suppress this warning, simply place a `\caption-setup{hypcap=false}` just before the `\caption` or `\captionof` command which is causing the warning. If this is not ok for you, you can set a hyperlink anchor with `\caption-setup{type=`⟨*float type*⟩`}` for yourself.

(See section 6.5: *hyperref*)

`Unsupported document class (or package) detected, usage of the caption package is not recommended.`

Either the document class you use is unknown to the caption package, or you have included a package in your document which redefines `\@makecaption` (which is responsible for typesetting the caption internally) as well. This means the caption package will either change the design of captions in an unwanted way, or it even refuses to work correctly at all.

(See section 5: *Document classes & Babel support* and section 6: *Package support*)

`Unused \captionsetup[`⟨*type*⟩`].`

You have specified options with `\captionsetup[`⟨*type*⟩`]` which are not used later on. This can be because of a typo in ⟨*type*⟩, or because you use a package which redefines `figure` or `table` and which is unknown to the caption package, or simply because you don't have any usage of the environment ⟨*type*⟩ after this line. (If you want to suppress this warning you can use `\captionsetup*` instead of `\captionsetup`.)

(See section 3.2: *Setting options*)

`Usage of the ⟨package⟩ package together with the caption package is strongly not recommended. Instead of loading the ⟨package⟩ package you should use the caption package option 'tableposition=top'.`

The package ⟨*package*⟩ is dealing with caption skips as well. Please decide which one do you actually want to use, the caption package or the other one, using both can lead to wrong skips above or below the caption.

(See section 2.6: *Skips*)

A.4 Errors

`Argument of \@caption has an extra }.`
– or –
`Paragraph ended before \@caption was complete.`

If you want to typeset something special (like a tabular) as caption, you need to give an optional argument to `\caption` resp. `\captionof` for the List of Figures resp. List of Tables, too, even if you don't use such list.

(See section 3.1: *Typesetting captions*, section 2.7: *Lists*, and section 6.5: *hyperref*)

`\caption outside float.`

You have placed a `\caption` command outside a floating environment, or a `longtable` or `wrapfigure`. If this is what you want to, please use either `\captionsetup{type=`⟨*type*⟩`}` + `\caption` or `\captionof`.

(See section 3.1: *Typesetting captions*)

`\ContinuedFloat outside float.`

You have placed a `\ContinuedFloat` command outside a floating environment. If this is what you want to, please use the combination `\captionsetup{type=`⟨*type*⟩`}` + `\ContinuedFloat`.

Please note that `\ContinuedFloat` inside a `longtable` is not working, but maybe using the `longtable*` environment, which typesets a `longtable` without incrementing the `table` counter, if sufficient for you.

(See section 3.3: *Continued floats* and section 6.8: *longtable*)

`Continued '`⟨*type*⟩`' after '`⟨*type*⟩`'.`

Continued figures or tables are not allowed to be interrupted by a floating environment (or `longtable`) of another type, e.g. a table between a figure and a continued figure.

(See section 3.3: *Continued floats*)

```
For a successful cooperation of the 'wrapfig' package
with the 'float' package you should load the 'wrapfig'
package *(right) after* the 'float' package.
```

Please take care of the load order of packages if you use the wrapfig package in cooperation with the float package.

(See section 6.17: *wrapfig*)

```
For a successful cooperation of the 'wrapfig' package
with the 'float' package you should use at least
'wrapfig' version 3.6.
```

Please take care of the version of the wrapfig package if you use the wrapfig package in cooperation with the float package.

(See section 6.17: *wrapfig*)

`For a successful cooperation we need at least version '`⟨*date*⟩`' of package `⟨*package*⟩`, but only version '`⟨*old-date*⟩`' is available.`

The caption package does not work with such an outdated package, please update it to a more recent version, at least to the one requested.

`Internal Error:` ⟨*error message*⟩`.`

You should never see this error. Please send me an e-mail reporting this issue.

`No float type '`⟨*type*⟩`' defined.`

The ⟨*type*⟩ you have specified in `\captionsetup{type=`⟨*type*⟩`}`, `\captionof{`⟨*type*⟩`}`, or `\DeclareCaptionSubType` is not defined. ⟨*type*⟩ should be either ‘`figure`’ or ‘`table`’, or any other floating environment defined with `\DeclareCaptionType`, `\newfloat` offered by the float package[6], or `\DeclareNewFloatType` offered by the floatrow package[8].

`Not allowed in longtable* environment.`

The usage of `\caption` is not allowed inside the `longtable*` environment. Please use either `\caption*` for a caption without label or use the regular `longtable` environment instead.

`Not available in compatibility mode.`

The feature required is not supported in compatibility mode. 'compatibility mode' means that the caption package has detected either an incompatible document class or an incompatible package which also extents the `\caption` command. Leaving the original extension intact, some features of the caption package are not supported.

(See section 6: *Package support*)

`Only one \caption can be placed in this environment.`

Inside the environments offered by the fltpage and sidecap package only *one* caption can be placed. (This is due implementation design.)

`Option clash for package caption.`
– but sometimes also –
`Missing \begin{document}.`

The caption package has already been loaded by some other LaTeX package, so you can't do that again specifying different options. A candidate causing this could be the subfig package; if this is the case, please load the caption package *before* the subfig package or specify the option `caption=false` while loading the subfig package.

(See the subfig package[20] documentation)

`Paragraph ended before \caption@makecurrent was complete.`
– or –
`Paragraph ended before \caption@prepareanchor was complete.`

If you want to typeset multi-paragraph captions, you need to give an optional argument to `\caption` resp. `\captionof` for the List of Figures resp. List of Tables, too, even if you don't use such list.

(See section 3.1: *Typesetting captions*, section 2.7: *Lists*, and section 6.5: *hyperref*)

`Something's wrong--perhaps a missing \caption`
`in the last figure or table.`

It seems that you have used a `\subcaption` command (or a different one which has typeset a sub-caption) without a corresponding `\caption` command. This is not supported.

`The option 'labelsep=`⟨*name*⟩`' does not work`
`with 'format=hang'.`
– or –
`The option 'labelsep=`⟨*name*⟩`' does not work`
`with \setcaphanging (which is set by default).`

A caption label separator which contains a `\\` command (like `labelsep=newline`) can not be combined with a hanging caption format (like `format=hang`). Please select either another caption label separator (e.g. `labelsep=colon`) or another caption format (e.g. `format=plain`).

(See section 2.1: *Formatting* resp. section 5.4: *KOMA-Script: scrartcl, scrreprt, and scrbook*)

`The package option 'caption=false' is obsolete.`
`Please pass this option to the subfig package instead`
`and do *not* load the caption package anymore.`

You have specified the option `caption=false`. This used to be a workaround for not using the whole caption package (leaving the caption stuff offered by the document class or other packages intact), but keeping the subfig package working. This mechanism is obsolete and not offered anymore, please pass this option to the subfig package instead and do not load the caption package anymore.

(See the subfig package[20] documentation)

`Undefined boolean value '⟨value⟩'.`

You tried to set a boolean option (like `singlelinecheck=` or `hypcap=`) with an improper value. Only `false`, `no`, `off`, `0` or `true`, `yes`, `on`, `1` is allowed here.

`Undefined format '⟨name⟩'.`

You tried to set a caption format which does not exists. Maybe a typo!?

(See section 2.1: *Formatting*)

`Undefined label format '⟨name⟩'.`

You tried to set a caption label format which does not exists. Maybe a typo!?

(See section 2.1: *Formatting*)

`Undefined label separator '⟨name⟩'.`

You tried to set a caption label separator which does not exists. Maybe a typo!?

(See section 2.1: *Formatting*)

`Undefined list format '⟨name⟩'.`

You tried to set a caption list-of format which does not exists. Maybe a typo!?

(See section 2.7: *Lists*)

`Undefined position '⟨name⟩'.`

You tried to set a caption position with an improper value. Maybe a typo!?

(See section 2.6: *Skips*)

`Undefined style '⟨name⟩'.`

You tried to set a caption style which does not exists. Maybe a typo!?

(See section 2.5: *Styles*)

`Usage of the 'position' option is incompatible`
`to the '⟨package⟩' package.`

The given package is dealing with caption skips as well. Please decide which one do you actually want to use: The `position=` option of the caption package or the mechanism of the other one; using both this way would lead to wrong skips above or below the caption and is therefore not supported.

(See section 2.6: *Skips*)

`Undefined text format '⟨name⟩'.`

You tried to set a caption text format which does not exists. Maybe a typo!?

(See section 2.1: *Formatting*)

```
You can't use both, the (obsolete) caption2 *and*
the (current) caption package.
```

This error message says it all, you simply can't do that. Please use only the caption package.

(See section C.2: *caption2 v2.x*)

B Version history

The version 1.0 was written in 1994 and was offering a handful of options to customize the look & feel of the captions. Furthermore this version already supported the rotating and subfigure packages. Version 1.1 introduced the `centerlast` option; version 1.2 added the support of the float package. Version 1,3 offered a better support of the subfigure package, while version 1.4 came with the new option `nooneline`.

The caption2 package 2.0 was an experimental side-version of the regular caption package. It was made public as beta test version without proper documentation in 1995 because of the strong demand for new features and adaptations to additional packages like the longtable package. Furthermore it offered a revised support of the subfigure package. (A version 2.1 was offered as bugfix-release in 2002.)

In 2003 I finally found some time, so a new regular release 3.0 of the caption package could be build in cooperation with Frank Mittelbach and Steven Cochran. It was released in December 2003 and superseded the neglected caption2 package. Main parts were re-written and it provided a complete re-work of the user interface. Furthermore it supported the hyperref, hypcap, listings, sidecap, and supertabular packages additionally.

While all the previous versions were designed for usage with the standard LaTeX document classes article, report, and book, the current version 3.1 released in 2007 also supports the $\mathcal{AMS}$, KOMA-Script, NTG, and SMF document classes, and the beamer class as well. Furthermore it was adapted to the frenchb Babel option, the frenchle & frenchpro packages, and the floatflt, fltpage, picinpar, picins, setspace, threeparttable, and wrapfig packages. New options and commands were introduced as well, among other things `font+`, `figurewithin` & `tablewithin`, `list` & `listformat`, `maxmargin` & `minmargin`, `\captionlistentry`, `\DeclareCaptionListFormat`, and `\DeclareCaptionType`. Further benefits are the new compatibility check (see section 6: *Package support*), the new "hypcap" feature (see section 6.5: *hyperref*), and the subcaption feature (see subcaption package documentation).

C Compatibility to older versions

C.1 caption v1.*x*

This version of the caption package still supports the old options and commands provided by the version 1.0 to 1.4 of this package. So there shouldn't occur any problems compiling old documents, but please don't mix old options and commands with the new ones. This isn't supported and can cause ugly side effects.

Here comes a short oversight of the obsolete options and how they have been replaced within this version of the caption package:

caption *v*1.*x*	caption *v*3.*x*
`normal`	`format=plain`
`hang`	`format=hang`
`isu`	`format=hang`
`center`	`justification=centering`
`centerlast`	`justification=centerlast`
`nooneline`	`singlelinecheck=off`
`scriptsize`	`font=scriptsize`
`footnotesize`	`font=footnotesize`
`small`	`font=small`
`normalsize`	`font=normalsize`
`large`	`font=large`
`Large`	`font=Large`
`up`	`labelfont=up`
`it`	`labelfont=it`
`sl`	`labelfont=sl`
`sc`	`labelfont=sc`
`md`	`labelfont=md`
`bf`	`labelfont=bf`
`rm`	`labelfont=rm`
`sf`	`labelfont=sf`
`tt`	`labelfont=tt`

Beside the options for setting up the desired font there were also the commands `\captionsize` resp. `\captionfont` and `\captionlabelfont` who could be redefined with `\renewcommand` and allowed an alternate and more flexible way to change the font used for captions. This mechanism was replaced by the commands

```
\DeclareCaptionFont{...}{...}     and
\captionsetup{font=...,labelfont=...}     .
```

(See section 4: *Own enhancements*)

Setting the margin for captions was done in *v*1.*x* with

```
\setlength{\captionmargin}{...}   .
```

This was replaced by

```
\captionsetup{margin=...}     .
```

(See section 2.4: *Margins and further paragraph options*)

For example the old-style code

```
\usepackage[hang,bf]{caption}
\renewcommand\captionfont{\small\sffamily}
\setlength\captionmargin{10pt}
```

will still work fine, but should be written today as

```
\usepackage[format=hang,labelfont=bf,font={small,sf},
            margin=10pt]{caption}
```

or

```
\usepackage{caption}
\captionsetup{format=hang,labelfont=bf,font={small,sf},
              margin=10pt}    .
```

The quite exotic option `ruled` which allowed a partial usage of the caption settings for `ruled` floats defined with the float package will be emulated by this version of the caption package, too. But using this option is not recommended anymore since this version of the caption package offers a more flexible way for changing the captions of these floating environments:

```
\DeclareCaptionStyle{ruled}{...}
```

resp.

```
\captionsetup[ruled]{...}    .
```

(See section 4: *Own enhancements*, section 3.2: *Setting options*, and section 6.2: *float*)

C.2 caption2 v2.*x*

Although they do very similar stuff, the packages caption and its experimental and now obsolete variant caption2 have a very different implementation design. Therefore a full compatibility could not be offered. For that reason you will still find a file called `caption2.sty` in this package distribution, so old documents using the caption2 package will still compile fine.

Newly created documents should use the actual version of the caption package instead. In most cases it's sufficient to replace the command

```
\usepackage[...]{caption2}
```

by

```
\usepackage[...]{caption}    .
```

But some options and commands will not be emulated, so you can get error messages afterwards. This section will hopefully help you removing these errors. If you have problems migrating from caption2 to caption please don't hesitate to send me an e-mail asking for help.

In addition to the obsolete options shown in the last section these ones will be emulated, too:

caption2 *v2.x*	caption *v3.x*
`flushleft`	`justification=raggedright`
`flushright`	`justification=raggedleft`
`oneline`	`singlelinecheck=on`

Setting the margin for captions was done in *v2.x* with

`\setcaptionmargin{...}` resp. `\setcaptionwidth{...}` .

This was replaced by

`\captionsetup{margin=...}` resp. `\captionsetup{width=...}` .

(See section 2.4: *Margins and further paragraph options*)

Setting an indention was done in *v2.x* with

```
\captionstyle{indent}
\setlength\captionindent{...}  .
```

This is now done with

```
\captionsetup{format=plain,indention=...}
```

.

The so-called single-line-check was controlled by the commands `\onelinecaptionsfalse` (for switching the check off) and `\onelinecaptionstrue` (for switching the check on). This was replaced by `\captionsetup{singlelinecheck=off}` resp. `\captionsetup{singlelinecheck=on}` . (See section 2.2: *Justification*)

The commands

```
\captionlabeldelim, \captionlabelsep, \captionstyle,
\defcaptionstyle, \newcaptionstyle, and \renewcaptionstyle
```

do not have a simple replacement and therefore will not be emulated by this version of the caption package. (So using them will cause error messages.) Rewriting such code is not always easy and straight-ahead, but by conscientious reading of this manual you should be able to find appropriate options and commands instead.

The *v*2.*x* option `ignoreLTcapwidth` do not have a replacement, too. But in most cases you can simply drop using that option because in this version of the caption package the value of `\LTcapwidth` will be ignored anyway (unless you set it to a different value than the default one which is `4in`). (See section 6.8: *longtable*)

C.3 caption v3.0

The caption package *v*3.0 did not support any document classes other than the standard LaTeX ones: article, report, and book. Therefore the default settings used to be fixed, but now the caption package supports more classes, therefore they are now set in dependence on the document class used.

For example: While in *v*3.0 the default caption `justification` used to be always `justified`, it's now still `justified` when using one of the standard document classes, but `raggedright` will be used as default format when used with the beamer document class.

An easy way to select the 'old' defaults is using the option `style=base` when loading the caption package (or later on using `\captionsetup`).

Another new feature is the automatic check for compatibility; if an incompatibility will be found, a warning message will be issued and if a strong incompatibility was found some features of the caption package will be disabled. Please note that only this check was added to the caption package, so if you get such warning message, the previous versions of the caption package were incompatible as well, but did not issued such warning, they were "only" having side-effects or causing problems. So these warnings only say that some bad side effects or problems could happen, but not that they actually will happen. Anyway, if you use the caption package in such circumstances, you should use it with care.

Note: The caption package *v*3.0 offered the option `caption=false`, which used to be a workaround for not using the whole caption package (leaving the caption stuff offered by the document class or other packages intact), but keeping the subfig package[20] working. This mechanism is obsolete and not offered anymore, please pass this option to the subfig package instead and do not load the caption package anymore.

References

[1] Frank Mittelbach and Michel Goossens:
The LaTeX Companion (2nd. Ed.),
Addison-Wesley, 2004.

[2] Till Tantau:
User Guide to the Beamer Class, Version 3.07,
March 11, 2007

[3] Markus Kohm & Jens-Uwe-Morawski:
KOMA-Script – a versatile LaTeX 2ε bundle,
2007-01-09

[4] Victor Eijkhout:
An introduction to the Dutch LaTeX document classes,
3 September 1989

[5] Rogério Brito:
Algorithms,
June 2, 2006

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

[7] Mats Dahlgren:
Welcome to the floatflt package,
1998/06/05

[8] Olga Lapko:
The floatrow package documentation,
2007/12/24

[9] Sebastian Gross:
Welcome to the beta test of fltpage package!,
1998/11/13

[10] Sebastian Rahtz & Heiko Oberdiek:
Hypertext marks in LaTeX,
November 12, 2007

[11] Heiko Oberdiek:
The hypcap package – Adjusting anchors of captions,
2007/04/09

[12] Carsten Heinz & Brooks Moses:
The Listings Package,
2007/02/22

[13] David Carlisle:
The longtable package,
2004/02/01

[14] Friedhelm Sowa:
Pictures in Paragraphs,
July 13, 1993

[15] Joachim Bleser and Edmund Lang:
PicIns-Benutzerhandbuch Version 3.0,
September 1992

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

[17] Erica M. S. Harris & Geoffrey Tobin:
LaTeX Document Package "setspace",
1 December 2000

[18] Rolf Niepraschk & Hubert Gäßlein:
The sidecap package,
2003/06/06

[19] Steven D. Cochran:
The subfigure package,
2002/07/02

[20] Steven D. Cochran:
The subfig package,
2005/07/05

[21] Johannes Braams and Theo Jurriens:
The supertabular environment,
2002/07/19

[22] Donald Arseneau:
Three part tables: title, tabular environment, notes,
2003/06/13

[23] Donald Arseneau:
WRAPFIG.STY ver 3.6,
2003/01/31

[24] Peter Wilson:
The xtab package,
2004/05/24