The Implementation of the caption package[*]

Axel Sommerfeldt
http://sourceforge.net/p/latex-caption/

2011/08/06

Abstract

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

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

User manuals

This document is describing the code implementation only. The user documentation can be found in

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

[*] This package has version number v3.1k, last revised 2009/10/09.

[1] Thanks a lot to Olga Lapko for this translation

Contents

1 Identification

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\def\caption@tempa$Id: #1 #2 #3-#4-#5 #6${%
  \def\caption@tempa{#3/#4/#5 }\def\caption@tempb{#2 }}
\caption@tempa $Id: caption.dtx 89 2013-05-02 07:05:20Z sommerfeldt $
\ProvidesPackage{caption}[\caption@tempa v3.3-\caption@tempb Customizing captions (A
```

2 Loading the kernel

```
\RequirePackage{caption3}[2013/05/01] % needs v1.6 or newer
```

3 Check against incompatible document classes

```
\caption@ifbool{documentclass}{}{%
  \caption@WarningNoLine{%
    Unsupported document class (or package) detected,\MessageBreak
    usage of the caption package is not recommended}%
  \caption@InfoNoLine{\string\@makecaption\space=\space\meaning\@makecaption}%
}
```

4 Check against incompatible packages

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

```
\caption@AtBeginDocument{%
  \@ifpackageloaded{ftcap}{\caption@DisablePositionOption{ftcap}}{}%
  \@ifpackageloaded{nonfloat}{\caption@DisablePositionOption{nonfloat}}{}%
  \@ifpackageloaded{topcapt}{\caption@DisablePositionOption{topcapt}}{}}
```

ption@DisablePositionOption

`\caption@DisablePositionOption{`⟨*package*⟩`}`
disables the 'position' option.

```
\newcommand*\caption@DisablePositionOption[1]{%
  \caption@InfoNoLine{%
    `#1' package detected; setting `position=b' for compatibility reasons}%
  \caption@setposition b%
```

```
  \DeclareCaptionOption{position}{%
    \caption@Error{Usage of the `position' option is incompatible\MessageBreak
      to the `#1' package}}}
```

```
\@onlypreamble\caption@DisablePositionOption
```

5 Declaration of options

5.1 Options for figure and table

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

```
\DeclareCaptionOption{tableposition}{%
  \captionsetup*[table]{position=#1}}
\@onlypreamble@key{caption}{tableposition}
```

```
\DeclareCaptionOption{figurename}{\caption@SetName{figure}{#1}}
\@onlypreamble@key{caption}{figurename}
\DeclareCaptionOption{tablename}{\caption@SetName{table}{#1}}
\@onlypreamble@key{caption}{tablename}
```

```
\DeclareCaptionOption{listfigurename}{\caption@SetName{listfigure}{#1}}
\@onlypreamble@key{caption}{listfigurename}
\DeclareCaptionOption{listtablename}{\caption@SetName{listtable}{#1}}
\@onlypreamble@key{caption}{listtablename}
```

`\caption@SetName`

`\caption@SetName{`⟨*float*⟩`}{`⟨*name*⟩`}`

```
\newcommand*\caption@SetName[2]{%
  \caption@NewFloat{\newfloat@setname{#1}{#2}}}
\@onlypreamble\caption@SetName
```

```
\DeclareCaptionOption{name}{\caption@setname\@captype{#1}}
```

`\caption@setname`

`\caption@setname{`⟨*float*⟩`}{`⟨*name*⟩`}`

```
\newcommand*\caption@setname[2]{%
  \@namedef{#1name}{#2}}
```

`caption@DeclareWithinOption`

`\caption@DeclareWithinOption{`⟨*float*⟩`}` declares the options ⟨*float*⟩`within` and ⟨*float*⟩`without`, e.g. `figurewithin` and `figurewithout`.

```
\newcommand*\caption@DeclareWithinOption[1]{%
  \DeclareCaptionOption{#1within}{\caption@Within{#1}{##1}}%
  \DeclareCaptionOptionNoValue{#1without}{\caption@Within{#1}{none}}}
\@onlypreamble\caption@DeclareWithinOption
```

```
\caption@DeclareWithinOption{figure}
\caption@DeclareWithinOption{table}
```

```
\DeclareCaptionOption{within}{%
  \caption@NewFloat{\newfloatsetup{within=#1}}}
\DeclareCaptionOptionNoValue{without}{%
  \caption@NewFloat{\newfloatsetup{without}}}
```

`\caption@Within`

`\caption@Within{`⟨*float*⟩`}{`⟨*value*⟩`}`

```
\newcommand*\caption@Within[2]{%
  \caption@NewFloat{\newfloat@setwithin{#1}{#2}}}
```

`\caption@NewFloat`

`\caption@NewFloat{`⟨*code*⟩`}` loads the newfloat package and executes the given code afterwards. Note: Since the newfloat package uses the crappy keyval package, too, we need to save & restore some macros here, otherwise this recursion won't work properly. *(TODO: Re-write* newfloat *so it uses* kvoptions *instead.)*

```
\newcommand*\caption@NewFloat[1]{%
  \let\KV@prefix@ORI\KV@prefix
  \let\@tempc@ORI\@tempc
  \caption@ifpackageloaded{newfloat}{#1}{}%
  \let\@tempc\@tempc@ORI
  \let\KV@prefix\KV@prefix@ORI}
```

5.2 Miscellaneous options

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

\caption@selectlistentry

\caption@selectlistentry{⟨heading *or* entry⟩}

```
\newcommand*\caption@selectlistentry[1]{%
  \caption@ifinlist{#1}{heading}{%
    \let\caption@iflistheading\@firstoftwo
  }{\caption@ifinlist{#1}{default,list-entry,entry}{%
    \let\caption@iflistheading\@secondoftwo
  }{%
    \caption@Error{Undefined list-entry selection `#1'}%
  }}}
```

```
\DeclareCaptionOption{list-entry}{\caption@selectlistentry{#1}}
\captionsetup{list-entry=default}
```

\caption@setparboxrestore

\caption@setparboxrestore{⟨partial *or* full⟩}

```
\newcommand*\caption@setparboxrestore[1]{%
  \caption@ifinlist{#1}{full}{%
    \caption@setfullparboxrestore
  }{\caption@ifinlist{#1}{default,light,partial}{%
    \let\caption@parboxrestore\@secondoftwo
  }{%
    \caption@Error{Undefined parboxrestore `#1'}%
  }}}
```

aption@setfullparboxrestore

\caption@setfullparboxrestore

This is an abbreviation for \caption@setparboxrestore{full}.

```
\newcommand*\caption@setfullparboxrestore{%
  \let\caption@parboxrestore\@firstoftwo}
```

```
\DeclareCaptionOption{parboxrestore}{\caption@setparboxrestore{#1}}
\captionsetup{parboxrestore=default}
```

```
\DeclareCaptionOption{@minipage}{%
  \caption@ifinlist{#1}{auto,default}%
    {\let\caption@if@minipage\@gobbletwo}%
    {\caption@set@bool\caption@if@minipage{#1}}}
\captionsetup{@minipage=default}
```

5.3 caption v1.x compatibility options

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

```
\DeclareCaptionOptionNoValue*{normal}{%
  \caption@setformat{plain}%
  \caption@setjustification{justified}}
\DeclareCaptionOptionNoValue*{isu}{%
  \caption@setformat{hang}%
  \caption@setjustification{justified}}
\DeclareCaptionOptionNoValue*{hang}{%
  \caption@setformat{hang}%
  \caption@setjustification{justified}}
\DeclareCaptionOptionNoValue*{center}{%
  \caption@setformat{plain}%
  \caption@setjustification{centering}}
\DeclareCaptionOptionNoValue*{centerlast}{%
  \caption@setformat{plain}%
  \caption@setjustification{centerlast}}
```

```
\DeclareCaptionOptionNoValue*{scriptsize}{\def\captionfont{\scriptsize}}
\DeclareCaptionOptionNoValue*{footnotesize}{\def\captionfont{\footnotesize}}
\DeclareCaptionOptionNoValue*{small}{\def\captionfont{\small}}
\DeclareCaptionOptionNoValue*{normalsize}{\def\captionfont{\normalsize}}
\DeclareCaptionOptionNoValue*{large}{\def\captionfont{\large}}
\DeclareCaptionOptionNoValue*{Large}{\def\captionfont{\Large}}
```

```
\DeclareCaptionOptionNoValue*{up}{\l@addto@macro\captionlabelfont\upshape}
\DeclareCaptionOptionNoValue*{it}{\l@addto@macro\captionlabelfont\itshape}
\DeclareCaptionOptionNoValue*{sl}{\l@addto@macro\captionlabelfont\slshape}
\DeclareCaptionOptionNoValue*{sc}{\l@addto@macro\captionlabelfont\scshape}
\DeclareCaptionOptionNoValue*{md}{\l@addto@macro\captionlabelfont\mdseries}
\DeclareCaptionOptionNoValue*{bf}{\l@addto@macro\captionlabelfont\bfseries}
\DeclareCaptionOptionNoValue*{rm}{\l@addto@macro\captionlabelfont\rmfamily}
\DeclareCaptionOptionNoValue*{sf}{\l@addto@macro\captionlabelfont\sffamily}
\DeclareCaptionOptionNoValue*{tt}{\l@addto@macro\captionlabelfont\ttfamily}
```

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

```
\caption@setbool{ruled}{0}
\DeclareCaptionOptionNoValue*{ruled}{\caption@setbool{ruled}{1}}
```

5.4 caption2 v2.x compatibility options

```
\DeclareCaptionOptionNoValue*{flushleft}{%
  \caption@setformat{plain}%
  \caption@setjustification{raggedright}}
\DeclareCaptionOptionNoValue*{flushright}{%
  \caption@setformat{plain}%
  \caption@setjustification{raggedleft}}
```

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

```
\DeclareCaptionOptionNoValue*{ignoreLTcapwidth}{%
```

```
  \caption@WarningNoLine{Obsolete option `ignoreLTcapwidth' ignored}}
```

5.5 Obsolete caption v3.0 options

```
\DeclareCaptionOption*{caption}{%
  \caption@setbool{temp}{#1}%
  \caption@ifbool{temp}{}{%
    \caption@Error{%
      The package option `caption=#1' is obsolete.\MessageBreak
      Please pass this option to the subfig package instead\MessageBreak
      and do *not* load the caption package anymore}}}
```

5.6 fltpage package support options

With these options is controlled where the list-of entry and `\ref` resp. `\pageref` or `\autoref` will link to. Defaults are `FPlist=caption` and `FPref=figure` which is inconsistent, but compatible to the usual behaviour of the fltpage package.

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

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

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

5.7 hyperref package support options

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

```
\DeclareCaptionOption{hypcap}[1]{\caption@setbool{hypcap}{#1}}
\DeclareCaptionOption{hypcapspace}{\def\caption@hypcapspace{#1}}
```

```
\captionsetup{hypcap=1,hypcapspace=.5\baselineskip}
```

6 AMS & SMF document classes support

```
\caption@ifamsclass{%
  \caption@InfoNoLine{AMS or SMF document class}%
```

```
  \setlength\belowcaptionskip{0pt}% set to 12pt by AMS class
```

```
}
```

7 KOMA-Script document classes support

```
\caption@ifkomaclass{%
  \caption@InfoNoLine{KOMA-Script document class}%
```

Here we emulate the caption related commands and take over the caption related settings from the KOMA-Script classes.

\@tablecaptionabovetrue
\@tablecaptionabovefalse

```
\g@addto@macro\@tablecaptionabovetrue{\captionsetup*[table]{position=t}}
\g@addto@macro\@tablecaptionabovefalse{\captionsetup*[table]{position=b}}
\if@tablecaptionabove
  \@tablecaptionabovetrue
\else
  \@tablecaptionabovefalse
\fi
```

\@figurecaptionabovetrue
\@figurecaptionabovefalse

```
\caption@ifundefined\@figurecaptionabovetrue{}{%
  \g@addto@macro\@figurecaptionabovetrue{\captionsetup*[figure]{position=t}}
  \g@addto@macro\@figurecaptionabovefalse{\captionsetup*[figure]{position=b}}
  \if@figurecaptionabove
    \@figurecaptionabovetrue
  \else
    \@figurecaptionabovefalse
  \fi
}%
```

\onelinecaptionstrue
\onelinecaptionsfalse

```
\g@addto@macro\onelinecaptionstrue{\let\caption@ifslc\@firstoftwo}
\g@addto@macro\onelinecaptionsfalse{\let\caption@ifslc\@secondoftwo}
\ifonelinecaptions
  \onelinecaptionstrue
\else
  \onelinecaptionsfalse
\fi
```

\@captionabovetrue
\@captionabovefalse

Please note that these are stronger than the `position` setting, therefore we override the options `figureposition` and `tableposition` to typeout a warning.

```
\g@addto@macro\@captionabovetrue{\let\caption@position\@firstoftwo}
\g@addto@macro\@captionabovefalse{\let\caption@position\@secondoftwo}
\DeclareCaptionOption{figureposition}{%
  \caption@WarningNoLine{Option ‘figureposition=#1’ has no effect\MessageBreak
  when used with a KOMA script document class}}
\DeclareCaptionOption{tableposition}{%
  \caption@WarningNoLine{Option ‘tableposition=#1’ has no effect\MessageBreak
  when used with a KOMA script document class}}
```

\setcapindent

```
\let\caption@KOMA@setcapindent\@setcapindent
\renewcommand*\@setcapindent[1]{%
  \caption@KOMA@setcapindent{#1}\caption@setcapindent}
\let\caption@KOMA@@setcapindent\@@setcapindent
\renewcommand*\@@setcapindent[1]{%
  \caption@KOMA@@setcapindent{#1}\caption@setcapindent}
```

```
\newcommand*\caption@setcapindent{%
  \captionsetup{indent=\ifdim\cap@indent<\z@\z@\else\cap@indent\fi}}
```

```
\caption@ifundefined\cap@indent{}{\caption@setcapindent}
```

\setcapwidth *Note:* The optional argument of \setcapwidth if not supported (yet), so we issue a warning if used. (Since this does not seem to have an negative effect when used by the captionbeside environment, we suppress the warning here.)

```
\expandafter\let\expandafter\caption@KOMA@setcapwidth
                \csname\string\setcapwidth\endcsname
\@namedef{\string\setcapwidth}[#1]#2{%
  \caption@KOMA@setcapwidth[#1]{#2}\caption@setcapwidth{#1}}
```

```
\newcommand*\caption@setcapwidth[1]{%
  \ifx\\#1\\\else
    \caption@ifundefined\cap@margin{}{%
      \def\@tempa{captionbeside}%
      \ifx\@tempa\@currenvir\else\caption@Warning{%
        Ignoring optional argument [#1] of \string\setcapwidth\MessageBreak}%
      \fi}%
  \fi
  \captionsetup{width=\cap@width}}
```

```
\def\caption@tempa{\hsize}%
\ifx\caption@tempa\cap@width \else
  \caption@setcapwidth{?}
\fi
```

\setcapmargin

```
\expandafter\let\expandafter\caption@KOMA@setcapmargin
                \csname\string\@setcapmargin\endcsname
\@namedef{\string\@setcapmargin}[#1]#2{%
  \caption@KOMA@setcapmargin[#1]{#2}\caption@setcapmargin}
```

```
\expandafter\let\expandafter\caption@KOMA@@setcapmargin
                \csname\string\@@setcapmargin\endcsname
\@namedef{\string\@@setcapmargin}[#1]#2{%
  \caption@KOMA@@setcapmargin[#1]{#2}\caption@setcapmargin}
```

```
\newcommand*\caption@setcapmargin{%
  \begingroup
    \let\onelinecaptionsfalse\relax
    \def\@twoside{0}%
    \def\if@twoside{\def\@twoside{1}\iffalse}%
    \cap@margin
    \def\@tempa{\endgroup}%
    \ifx\cap@left\hfill\else\ifx\cap@right\hfill\else
      \def\hspace##1##{\@firstofone}%
      \edef\@tempa{\endgroup
        \noexpand\captionsetup{%
          twoside=\@twoside,slc=0,%
```

```
            margin={\cap@left,\cap@right}}}%
      \fi\fi
      \@tempa}
  \ifx\cap@margin\relax \else
    \caption@setcapmargin
  \fi
}
```

8 Processing of options

```
\caption@SetupOptions{caption}{\caption@setkeys{#1}{#2}}
\caption@ProcessOptions*{caption}
```

9 \caption, \@caption, and \@makecaption

\ifcaption@caption
\ifcaption@subcaption
\ifcaption@ContinuedFloat

Since we support continued floats and sub-figures it's not an easy task to manage the figure (and table) counter. (Especially since previous versions of the caption package has proven that correcting a counter locally is not a good idea.)
These three switches hold the current status: \ifcaption@caption is set if the caption was already typeset (so the counter doesn't need to be incremented for sub-figures), \ifcaption@subcaption is set if there is already content which have incremented the counter (so the counter doesn't need to be incremented for the main caption, further sub-figures, and other stuff), and \ifcaption@ContinuedFloat is set if \ContinuedFloat was given.

```
\newif\ifcaption@caption
\newif\ifcaption@subcaption
\newif\ifcaption@ContinuedFloat
```

\caption@clrflag
\caption@setflag

Furthermore we introduce the macros \caption@clrflag and \caption@setflag for clearing resp. setting these flags.

```
\newcommand*\caption@clrflag[1]{%
  \caption@chgflag{#1}{false}}
\newcommand*\caption@setflag[1]{%
  \caption@chgflag{#1}{true}}
\newcommand*\caption@chgflag[2]{%
  \global\csname caption@#1#2\endcsname}
```

The floatrow package uses \FR@loc@ to keep changes (flags, counters, ...) local. Since it is not aware of our flags above we need to extend that macro (if defined).

```
\caption@AtBeginDocument{%
  \caption@ifundefined\FR@loc@{}{%
    \caption@InfoNoLine{floatrow package is loaded}%
    \g@addto@macro\FR@loc@{%
      \renewcommand*\caption@chgflag[2]{%
        \csname caption@#1#2\endcsname}}}}
```

\caption@caption

Here comes our definition of \caption and \caption*. Beside the support of the starred variant this code was adapted to the various packages we support. We are using \caption@dblarg instead of \@dblarg so \caption{} (with an empty arg.) will produce a list-of entry, but \caption[]{} won't.

```
\def\caption@caption{%
  \caption@iftype
    {\caption@checkgrouplevel\@empty\caption
     \caption@star
       {\caption@refstepcounter\@captype}%
       {\caption@dblarg{\@caption\@captype}}}%
    {\caption@Error{\noexpand\caption outside float}%
     \caption@gobble}}
```

\caption@star

A helper macro which processes the optional * after \caption.

```
\newcommand*\caption@star[2]{%
  \@ifstar{\caption@startrue#2[]}{#1#2}}
```

\caption@@caption

As above, our version has been adapted to the packages we support. Additionally our code is nested by \caption@beginex & \caption@end instead of \begingroup & \endgroup. Furthermore we use \caption@boxrestore instead of \@parboxrestore so this code also works correctly inside list-based environments like wide & addmargin. (This, and the fact that we use \linewidth instead of \hsize inside \@makecaption, solves LaTeX PR latex/2472.)

```
\long\def\caption@@caption#1[#2]#3{%

  \ifcaption@star \else
    \caption@prepareanchor{#1}{#2}%
    \memcaptioninfo{#1}{\csname the#1\endcsname}{#2}{#3}%
    \@nameuse{nag@hascaptiontrue}%
  \fi

  \par
  \caption@beginex{#1}{#2}{#3}%
    \caption@setfloatcapt{%
      \caption@boxrestore
      \if@minipage
        \@setminipage
      \fi
      \caption@normalsize
      \ifcaption@star
        \let\caption@makeanchor\@firstofone
      \fi
      \@makecaption{\csname fnum@#1\endcsname}%
                   {\ignorespaces\caption@makeanchor{#3}}\par
      \caption@if@minipage\@minipagetrue\@minipagefalse}%
  \caption@end}
```

memoir document class stuff:

```
\providecommand\M@gettitle[1]{}
\providecommand\memcaptioninfo[4]{}
```

\caption@prepareanchor

```
\newcommand*\caption@prepareanchor[2]{%
  \caption@makecurrent{#1}{#2}%
  \caption@ifhypcap\caption@@start\relax
  \M@gettitle{#2}}
```

\caption@makecaption

`\@makecaption{`⟨*label*⟩`}{`⟨*text*⟩`}`

We do basically the same as the original code (from the standard LaTeX document classes), but take care of the `position=` setting and use `\caption@@make` from the caption kernel to finally typeset the caption.

```
\long\def\caption@makecaption#1#2{%
  \caption@iftop
    {\vskip\belowcaptionskip}%
    {\caption@rule\vskip\abovecaptionskip}%
  \caption@@make{#1}{#2}%
  \caption@iftop
    {\vskip\abovecaptionskip\caption@rule}%
    {\vskip\belowcaptionskip}}
```

\caption@redefine

We only redefine `\caption` and `\@caption` if the current definitions are well known, so documents written in the old (caption package *v1.x*) days (where `\caption` & `\@caption` were not redefined by us) will still compile fine. For example the usage of the (now obsolete) captcont package, which brings its own definition of `\caption*`, was quite common these days.

```
\newcommand*\caption@redefine{}
\g@addto@macro\caption@redefine{%

  \caption@setbool{incompatible}{0}%

  \caption@CheckCommand\caption{%
    % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
    \def\caption{%
       \ifx\@captype\@undefined
         \@latex@error{\noexpand\caption outside float}\@ehd
         \expandafter\@gobble
       \else
         \refstepcounter\@captype
         \expandafter\@firstofone
       \fi
       {\@dblarg{\@caption\@captype}}%
    }}%

  \caption@CheckCommand\caption{%
    % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
    \def\caption{
      \ifx\@captype\@undefined
        \@latex@error{\noexpand\caption outside figure or table}\@ehd
        \expandafter\@gobble
      \else
        \refstepcounter\@captype
        \expandafter\@firstofone
      \fi
      {\@dblarg{\@caption\@captype}}%
    }}%
  \caption@CheckCommand\caption{%
    % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
    \long\def\caption{%
      \ifx\@captype\@undefined
        \@latex@error{\noexpand\caption outside float}\@ehd
        \expandafter\@gobble
      \else
        \refstepcounter\@captype
        \let\@tempf\@caption
        \expandafter\ifx\csname @float@c@\@captype\endcsname\relax\else
          \expandafter\expandafter\let
            \expandafter\@tempf\csname @float@c@\@captype\endcsname
        \fi
      \fi
      \@dblarg{\@tempf\@captype}}}%
  \caption@CheckCommand\caption{%
    % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
    % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
    % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
    \def\caption{%
      \ifx\@captype\@undefined
        \@latex@error{\noexpand\caption outside float}\@ehd
        \expandafter\@gobble
      \else
        \H@refstepcounter\@captype
        \@ifundefined{fst@\@captype}{%
          \let\Hy@tempa\@caption
        }{%
          \let\Hy@tempa\Hy@float@caption
        }%
        \expandafter\@firstofone
      \fi
      {\@dblarg{\Hy@tempa\@captype}}%
    }}%
  \caption@CheckCommand\caption{%
    % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
    \def\caption{%
      \ifx\@captype\@undefined
        \@latex@error{\noexpand\caption outside float}\@ehd
        \expandafter\@gobble
      \else
        \H@refstepcounter\@captype
        \let\Hy@tempa\@caption
        \@ifundefined{float@caption}{%
        }{%
```

```
        \expandafter\ifx\csname @float@c@\@captype\endcsname\float@caption
          \let\Hy@tempa\Hy@float@caption
        \fi
      }%
      \expandafter\@firstofone
    \fi
    {\@dblarg{\Hy@tempa\@captype}}%
  }}%
\caption@CheckCommand\caption{%
  % memhfixc.sty [2010/08/17 v1.15 nameref/hyperref package fixes for memoir clas
  % \let\m@moldhypcaption\caption
  \long\def\caption{\donemaincaptiontrue\m@moldhypcaption}}%
\caption@IfCheckCommand{}{%
  \caption@InfoNoLine{%
    Incompatible package detected (regarding \string\caption).\MessageBreak
    \string\caption\space=\space\meaning\caption}%
  \caption@setbool{incompatible}{1}}%
\caption@CheckCommand\@caption{%
  % ltfloat.dtx [2002/10/01 v1.1v LaTeX Kernel (Floats)]
  \long\def\@caption#1[#2]#3{%
    \par
    \addcontentsline{\csname ext@#1\endcsname}{#1}%
      {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
    \begingroup
      \@parboxrestore
      \if@minipage
        \@setminipage
      \fi
      \normalsize
      \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
    \endgroup}}%
\caption@CheckCommand\@caption{%
  % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
  \long\def\@caption#1[#2]#3{% second argument ignored
    \par\nobreak
    \begingroup
      \@parboxrestore
      \if@minipage
        \@setminipage
      \fi
      \beamer@makecaption{#1}{\ignorespaces #3}\par\nobreak
      \endgroup}}%
\caption@CheckCommand\@caption{%
  % memhfixc.sty [2010/08/17 v1.15 nameref/hyperref package fixes for memoir clas
  \long\def\@caption#1[#2]#3{%
    \MNR@old@caption{#1}[{#2}]{#3}%
    \def\@currentlabelname{#2}%
    \M@gettitle{#2}%
  }}%
\caption@CheckCommand\@caption{%
  % magyar.ldf [2005/03/30 v1.4j Magyar support from the babel system]
  \long\def\@caption#1[#2]#3{%
    \csname par\endcsname
    \addcontentsline{\csname ext@#1\endcsname}{#1}%
      {\protect\numberline{\csname the#1\endcsname.}{\ignorespaces #2}}%
    \begingroup
      \@parboxrestore
      \if@minipage
        \@setminipage
      \fi
      \normalsize
      \@makecaption{\csname fnum@#1\endcsname}%
          {\ignorespaces #3}\csname par\endcsname
    \endgroup}}%
%   \caption@CheckCommand\float@caption{%
%     % float.sty [2001/11/08 v1.3d Float enhancements (AL)]
%     \long\def\float@caption#1[#2]#3{%
%       \addcontentsline{\@nameuse{ext@#1}}{#1}%
%        {\protect\numberline{\@nameuse{the#1}}{\ignorespaces #2}}
%       \global\setbox\@floatcapt\vbox\bgroup\@parboxrestore
%         \normalsize\@fs@capt{\@nameuse{fnum@#1}}{\ignorespaces #3}%
%         \@ifnextchar[{\float@ccon}{\egroup}}%
%     \long\def\float@ccon[#1]{#1\par\egroup}}%
\caption@CheckCommand\@caption{%
  % hyperref.sty [2007/02/27 v6.75t Hypertext links for LaTeX]
  \long\def\@caption#1[#2]#3{%
    \hyper@makecurrent{\@captype}%
    \def\@currentlabelname{#2}%
    \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
      \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
    }%
    \begingroup
      \@parboxrestore
      \if@minipage
        \@setminipage
      \fi
      \normalsize
      \@makecaption{\csname fnum@#1\endcsname}{%
        \ignorespaces
        \ifHy@nesting
          \hyper@@anchor{\@currentHref}{#3}%
        \else
          \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
        \fi
      }%
      \par
    \endgroup
  }}%

\caption@CheckCommand\@caption{%
  % hyperref.sty [2007/04/09 v6.76a Hypertext links for LaTeX]
  % hyperref.sty [2007/06/12 v6.76h Hypertext links for LaTeX]
  % hyperref.sty [2007/08/05 v6.76j Hypertext links for LaTeX]
  \long\def\@caption#1[#2]#3{%
    \expandafter\ifx\csname if@capstart\expandafter\endcsname
                    \csname iftrue\endcsname
      \global\let\@currentHref\hc@currentHref
    \else
      \hyper@makecurrent{\@captype}%
    \fi
    \def\@currentlabelname{#2}%
    \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
      \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
    }%
    \begingroup
      \@parboxrestore
      \if@minipage
        \@setminipage
      \fi
      \normalsize
      \expandafter\ifx\csname if@capstart\expandafter\endcsname
                      \csname iftrue\endcsname
        \global\@capstartfalse
        \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
      \else
        \@makecaption{\csname fnum@#1\endcsname}{%
          \ignorespaces
          \ifHy@nesting
            \hyper@@anchor{\@currentHref}{#3}%
          \else
            \Hy@raisedlink{\hyper@@anchor{\@currentHref}{\relax}}#3%
          \fi
        }%
      \fi
      \par
    \endgroup
  }}%

\caption@CheckCommand\@caption{%
  % hyperref.sty [2009/11/27 v6.79k Hypertext links for LaTeX]
  \long\def\@caption#1[#2]#3{%
    \expandafter\ifx\csname if@capstart\expandafter\endcsname
                    \csname iftrue\endcsname
      \global\let\@currentHref\hc@currentHref
    \else
      \hyper@makecurrent{\@captype}%
    \fi
    \def\@currentlabelname{#2}%
    \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
      \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
    }%
    \begingroup
      \@parboxrestore
      \if@minipage
        \@setminipage
      \fi
      \normalsize
      \expandafter\ifx\csname if@capstart\expandafter\endcsname
                      \csname iftrue\endcsname
        \global\@capstartfalse
        \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
      \else
        \@makecaption{\csname fnum@#1\endcsname}{%
          \ignorespaces
          \ifHy@nesting
            \expandafter\hyper@@anchor\expandafter{\@currentHref}{#3}%
          \else
            \Hy@raisedlink{%
              \expandafter\hyper@@anchor\expandafter{\@currentHref}{\relax}%
            }%
            #3%
          \fi
        }%
      \fi
      \par
    \endgroup
  }}%
\caption@CheckCommand\@caption{%
  % hyperref.sty [2009/12/09 v6.79m Hypertext links for LaTeX]
  % hyperref.sty [2009/12/28 v6.79z Hypertext links for LaTeX]
  \long\def\@caption#1[#2]#3{%
    \expandafter\ifx\csname if@capstart\expandafter\endcsname
                    \csname iftrue\endcsname
      \global\let\@currentHref\hc@currentHref
    \else
      \hyper@makecurrent{\@captype}%
    \fi
    \@ifundefined{NR@gettitle}{%
      \def\@currentlabelname{#2}%
    }{%
      \NR@gettitle{#2}%
    }%
    \par\addcontentsline{\csname ext@#1\endcsname}{#1}{%
      \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}%
    }%
    \begingroup
      \@parboxrestore
      \if@minipage
        \@setminipage
      \fi
      \normalsize
      \expandafter\ifx\csname if@capstart\expandafter\endcsname
                      \csname iftrue\endcsname
        \global\@capstartfalse
        \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces#3}%
      \else
        \@makecaption{\csname fnum@#1\endcsname}{%
          \ignorespaces
          \ifHy@nesting
            \expandafter\hyper@@anchor\expandafter{\@currentHref}{#3}%
          \else
            \Hy@raisedlink{%
              \expandafter\hyper@@anchor\expandafter{%
                \@currentHref
              }{\relax}%
            }%
            #3%
          \fi
        }%
      \fi
      \par
    \endgroup
  }}%
\caption@CheckCommand\@caption{%
  % nameref.sty [2006/12/27 v2.28 Cross-referencing by name of section]
  \long\def\@caption#1[#2]{%
    \def\@currentlabelname{#2}%
    \NR@@caption{#1}[{#2}]%
  }}%
\caption@CheckCommand\@caption{%
  % nameref.sty [2009/11/27 v2.32 Cross-referencing by name of section]
  \long\def\@caption#1[#2]{%
    \NR@gettitle{#2}%
    \NR@@caption{#1}[{#2}]%
  }}%
\caption@CheckCommand\@caption{%
  % subfigure.sty [2002/07/30 v2.1.4 subfigure package]
  \long\def\@caption#1[#2]#3{%
    \@ifundefined{if#1topcap}%
      {\subfig@oldcaption{#1}[{#2}]{#3}}%
      {\@nameuse{if#1topcap}%
         \@listsubcaptions{#1}%
         \subfig@oldcaption{#1}[{#2}]{#3}%
       \else
         \subfig@oldcaption{#1}[{#2}]{#3}%
         \@listsubcaptions{#1}%
       \fi}}}%
\caption@CheckCommand\@caption{%
  % subfig.sty [2005/06/28 ver: 1.3 subfig package]
  \def\@caption{\caption@}%
%   \long\def\caption@#1[#2]#3{%
%     \@ifundefined{caption@setfloattype}%
%       \caption@settype
%       \caption@setfloattype
%           \@captype
%     \sf@ifpositiontop{%
%       \@listsubcaptions{#1}%
%       \sf@old@caption{#1}[{#2}]{#3}%
%     }{%
%       \sf@old@caption{#1}[{#2}]{#3}%
%       \@listsubcaptions{#1}%
%     }}%
  }%
\caption@IfCheckCommand{}{%
  \caption@InfoNoLine{%
    Incompatible package detected (regarding \string\@caption).\MessageBreak
    \string\@caption\space=\space\meaning\@caption}%
  \caption@setbool{incompatible}{1}}%
```

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

```
\caption@ifundefined\caption@ifcompatibility
  {\let\caption@ifcompatibility\caption@ifincompatible
   \let\caption@tempa\caption@WarningNoLine}%
  {\let\caption@tempa\@gobble}% suppress warning
\caption@ifcompatibility{%
  \caption@tempa{%
    \noexpand\caption will not be redefined since it’s already\MessageBreak
    redefined by a document class or package which is\MessageBreak
    unknown to the caption package}%
  \renewcommand*\caption@redefine{}%
```

`\ContinuedFloat` is not supported in compatibility mode.

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

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

```
  \caption@AtBeginDocument*{%
    \let\caption@start\relax
    \caption@ifundefined\caption@ORI@capstart{}{%
      \caption@Debug{%
        Restore hypcap definition of \string\capstart\@gobble}%
      \let\capstart\caption@ORI@capstart}%
  }%
```

\caption@star

We redefine \caption@star here so it does not make any harm.

```
    \renewcommand*\caption@star[2]{#1#2}%
```

```
  }{%
```

```
    \caption@ifincompatible{%
      \caption@WarningNoLine{%
        Forced redefinition of \noexpand\caption since the\MessageBreak
        unsupported(!) package option ‘compatibility=false’\MessageBreak
        was given}%
    }{}%
```

\caption

\@caption

```
    \renewcommand*\caption@redefine{%
      \let\caption\caption@caption
      \let\@caption\caption@@caption}%
    \caption@redefine
```

```
  }%
```

```
  \caption@AtBeginDocument*{%
    \let\caption@ORI@capstart\@undefined}%
```

\@xfloat

We redefine \@xfloat so inside floating environments our type-specific options will be used, a hyperref anchor will be set etc.

```
  \let\caption@ORI@xfloat\@xfloat
  \def\@xfloat#1[#2]{%
    \caption@ORI@xfloat{#1}[#2]%
    \caption@settype{#1}}%
```

```
}
```

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

```
\caption@AtBeginDocument{\caption@redefine}
```

\@makecaption

```
\let\@makecaption\caption@makecaption
```

\phantomcaption

\phantomcaption
Use this one for figures with subcaptions but without main caption.

```
\newcommand\phantomcaption{%
  \caption@iftype
    {\caption@refstepcounter\@captype}%
    {\caption@Error{\noexpand\phantomcaption outside float}}}%
```

10 \captionof

```
\caption@AtBeginDocument{%
  \DeclareCaptionOption{type}{\setcaptiontype{#1}}%
  \DeclareCaptionOption{type*}{\setcaptiontype*{#1}}%
  \DeclareCaptionOptionNoValue{subtype}{\setcaptionsubtype\relax}%
  \DeclareCaptionOptionNoValue{subtype*}{\setcaptionsubtype*}%
}
```

Important Note: Like `\captionof` the option `type=` should only be used inside a group, box, or environment and does not check if the argument is a valid floating environment or not.

`\setcaptiontype` Like `\captionsetup{type=xxx}`, but also works if `\captionsetup` was redefined.

```
\newcommand\setcaptiontype{%
  \caption@boxrestore@mini
  \caption@settype}
```

`\setcaptionsubtype` Same, but sets the sub-type.

```
\newcommand\setcaptionsubtype{%
  \caption@iftype
    \caption@setsubtype
    {\caption@Error{\noexpand\setcaptionsubtype outside float}}}%
\newcommand\caption@setsubtype{%
  \@ifstar
    {\caption@@settype{sub}*{sub\@captype}}%
    {\caption@@settype{sub}{sub\@captype}}}%
```

`\caption@settype` `\caption@settype*{`⟨*type*⟩`}`
sets `\@captype` and executes the options associated with it (using `\caption@setoptions`). Furthermore we check `\currentgrouplevel` (if avail), redefine `\@currentlabel` so a `\label` before `\caption` will result in a hint instead of a wrong reference, and use the macro `\caption@(sub)typehook` (which will be used by our float package support).
The non-starred version sets a hyperref anchor additionally (if `hypcap=true` and the hypcap package is not loaded).

```
\newcommand*\caption@settype{%
  \caption@clrflag{caption}%
  \caption@clrflag{subcaption}%
  \caption@clrflag{ContinuedFloat}%
  \caption@set@type}
\newcommand*\caption@set@type{%
  \caption@@settype{}}
\newcommand*\caption@@settype[1]{%
  \caption@teststar{\caption@@@settype{#1}}\@firstoftwo\@secondoftwo}
```

```
\newcommand*\caption@@@settype[3]{%
% #1 = "" or "sub"
% #2 = \@firstoftwo in star form, \@secondoftwo otherwise
% #3 = <type>, e.g. "figure" or "table"
  \caption@Debug{#1type=#3}%
  \caption@checkgrouplevel{#1}{%
    \captionsetup{#1type#2*\@empty=...}#2{ or
                  \@backslashchar#1captionof}{}}%
```

```
  \edef\caption@tempa{#3}%
  \expandafter\ifx\csname @#1captype\endcsname\caption@tempa \else
    \ifcaptionsetup@star\else\@nameuse{caption@#1type@warning}\fi
  \fi
  \expandafter\let\csname @#1captype\endcsname\caption@tempa
```

```
  \@nameuse{caption@#1typehook}%
```

```
  \caption@setoptions{#3}%
  \ifx\caption@opt\relax
    \@nameundef{caption@#1type@warning}%
  \else
    \@namedef{caption@#1type@warning}{\caption@Warning{%
      The #1caption type was already set to
      `\csname @#1captype\endcsname'\MessageBreak}}%
  \fi
```

```
  #2{}{%
    \let\@currentlabel\caption@undefinedlabel
%   \let\@currentHlabel\@undefined
    \ifx\caption@x@label\@undefined
      \let\caption@x@label\label
      \let\label\caption@xlabel
    \fi
    \caption@start}}
```

\caption@typehook

Hook which will be used inside \caption@settype. It will be extended later on, e.g. by our float package support.

```
\newcommand*\caption@typehook{}
```

\caption@iftype

Since we often need to check if \@captype is defined (means: we are inside a floating environment) this helper macro was introduced.

```
\newcommand*\caption@iftype{%
  \caption@ifundefined\@captype\@secondoftwo\@firstoftwo}
```

\caption@checkgrouplevel

Checks if \captionsetup{type=...} or \caption is done inside a group or not – in the latter case a warning message will be issued. (needs ε-TeX)

```
\caption@ifeTeX{%
  \newcommand*\caption@checkgrouplevel[2]{%
    \@ifundefined{#1caption@grouplevel}{%
       \caption@ifundefined\caption@grouplevel{\let\caption@grouplevel\z@}{}%
       \ifnum\currentgrouplevel>\caption@grouplevel\relax
```

```
        \expandafter\edef\csname #1caption@grouplevel\endcsname{%
            \the\currentgrouplevel}%
       \else
         \caption@Warning{\string#2\MessageBreak outside box or environment}%
       \fi
    }{}}%
}{%
  \let\caption@checkgrouplevel\@gobbletwo
}
```

\caption@undefinedlabel

This label will be used for \currentlabel inside (floating) environments as default. (see above)

```
\newcommand*\caption@undefinedlabel{%
  \protect\caption@xref{\caption@labelname}{\on@line}}
```

```
\DeclareRobustCommand*\caption@xref[2]{%
  \caption@WarningNoLine{\noexpand\label without proper \string\caption#2}%
  \@setref\relax\@undefined{#1}}
```

```
\newcommand*\caption@labelname{??}
```

\caption@xlabel

The new code of \label inside floating environments. \label will be redefined using \caption@withoptargs, so #1 are the optional arguments (if any), and #2 is the mandatory argument here.

```
\newcommand*\caption@xlabel{%
  \caption@withoptargs\caption@@xlabel}
```

```
\newcommand*\caption@@xlabel[2]{%
  \caption@@@xlabel
  \def\caption@labelname{#2}%
  \caption@x@label#1{#2}}
```

```
\newcommand*\caption@@@xlabel{%
  \global\let\caption@@@xlabel\@empty
  \@bsphack
    \protected@write\@auxout{}%
      {\string\providecommand*\string\caption@xref[2]{%
        \string\@setref\string\relax\string\@undefined{\string##1}}}%
  \@esphack}
```

\captionof

\captionof{⟨*type*⟩}[⟨*lst_entry*⟩]{⟨*heading*⟩}
\captionof*[⟨*lst_entry*⟩]{⟨*heading*⟩}

Note: This will be defined with \AtBeginDocument so \usepackage{caption,capt-of} will still work. (Compatibility to *v1.x*)

```
\caption@AtBeginDocument{%
  \def\captionof{\caption@teststar\caption@of{\caption*}\caption}}
```

```
\newcommand*\caption@of[2]{\setcaptiontype*{#2}#1}
```

11 \captionlistentry

\captionlistentry

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

```
\newcommand*\captionlistentry{%
  \caption@teststar\@captionlistentry\@firstoftwo\@secondoftwo}
```

```
\newcommand*\@captionlistentry[1]{%
  \@testopt{\caption@listentry{#1}}\@captype}
```

```
\def\caption@listentry#1[#2]#3{%
  \@bsphack
    #1{\caption@gettitle{#3}}%
      {\caption@refstepcounter{#2}%
       \caption@makecurrent{#2}{#3}}%
    \caption@addcontentsline{#2}{#3}%
  \@esphack}
```

12 \captionbox

\captionbox

A \parbox with contents and caption, separated by an invisible \hrule.

```
\newcommand*\captionbox{%
  \caption@withoptargs{\caption@ibox\@gobble}}
```

```
\newcommand\caption@ibox[3]{%
  \@testopt{\caption@iibox{#1}{#2}{#3}}{\wd\@tempboxa}}
```

```
\long\def\caption@iibox#1#2#3[#4]{%
  \@testopt{\caption@iiibox{#1}{#2}{#3}{#4}}\captionbox@hj@default}
```

```
\long\def\caption@iiibox#1#2#3#4[#5]#6{%
  \setbox\@tempboxa\hbox{#6}%
  \begingroup
  #1*% set \caption@position
  \caption@iftop{%
    \endgroup
    \parbox[t]{#4}{%
      #1\relax
      \caption@setposition t%
      \vbox{\caption#2{#3}}%
      \captionbox@hrule
      \csname caption@hj@#5\endcsname
      \unhbox\@tempboxa}%
  }{%
    \endgroup
    \parbox[b]{#4}{%
      #1\relax
      \caption@setposition b%
      \csname caption@hj@#5\endcsname
      \unhbox\@tempboxa
      \captionbox@hrule
      \vtop{\caption#2{#3}}}%
  }}
```

```
\newcommand*\captionbox@hj@default{c}
\newcommand*\captionbox@hrule{\hrule\@height\z@\relax}
```

```
\providecommand*\caption@hj@c{\centering}
\providecommand*\caption@hj@l{\raggedright}
\providecommand*\caption@hj@r{\raggedleft}
\providecommand*\caption@hj@s{}
```

13 \ContinuedFloat

\ContinuedFloat

`\ContinuedFloat`
`\ContinuedFloat*`
This mainly sets the appropriate flag, increments the continuation counter, and executes the given options. Furthermore we set `\caption@resetContinuedFloat` to `\@gobble` so the continuation counter will not be reset to zero inside `\caption@refstepcounter`.
When the hyperref package is used we have the problem that the usage of `\ContinuedFloat` will create duplicate hyper links – `\@currentHref` will be the same for the main float and the continued ones. So we have to make sure unique labels and references will be created each time. We do this by extending `\theHfigure` and `\theHtable`, so for continued floats the scheme

⟨*type*⟩`.`⟨*type #*⟩`\alph{`⟨*continued #*⟩`}`

will be used instead of

⟨*type*⟩`.`⟨*type #*⟩ .

(This implementation follows an idea from Steven Douglas Cochran.)
Note: This does not help if the hyperref package option `naturalnames=true` is set.

```
\def\ContinuedFloat{%
  \caption@iftype
    {\caption@ContinuedFloat\@captype}%
    {\caption@Error{\noexpand\ContinuedFloat outside float}}}
```

```
\newcommand*\caption@ContinuedFloat[1]{%
  \@ifstar
    {\caption@@refstepcounter\@captype
     \caption@@ContinuedFloat{#1}}%
    {\caption@Continued@Float{#1}}}
```

```
\newcommand*\caption@Continued@Float[1]{%
  \edef\caption@tempa{#1}%
  \ifx\caption@tempa\caption@CFtype
    \caption@restorecounters
    \caption@@ContinuedFloat{#1}%
  \else
    \caption@Error{Continued `#1' after `\caption@CFtype'}%
  \fi}
```

```
\newcommand*\caption@@ContinuedFloat[1]{%
  \caption@setflag{ContinuedFloat}%
  \stepcounter{ContinuedFloat}%
  \caption@@@ContinuedFloat{#1}}
```

```
\newcommand*\caption@@@ContinuedFloat[1]{%
  \caption@setoptions{ContinuedFloat}%
  \caption@setoptions{continued#1}%
  \expandafter\l@addto@macro\csname the#1\endcsname\theContinuedFloat
  \@ifundefined{theH#1}{}{%
    \expandafter\l@addto@macro\csname theH#1\endcsname{%
      \@alph\c@ContinuedFloat}}%
  \let\caption@@@ContinuedFloat\@gobble}
```

```
\newcommand*\caption@CFtype{??}
```

\theContinuedFloat

This one is preset to \@empty, so usually the continuation counter is not included in the caption label or references.

```
\newcounter{ContinuedFloat}
\let\theContinuedFloat\@empty
```

caption@resetContinuedFloat

\caption@resetContinuedFloat{⟨*type*⟩}
If a continuation counter is defined, we reset it. (This one will be called inside \@caption.)

```
\newcommand*\caption@resetContinuedFloat[1]{%
  \xdef\caption@CFtype{#1}%
  \@stpelt{ContinuedFloat}}
```

\donemaincaptionfalse

Since the memoir document class resets the sub-caption counter at \@float, right after \donemaincaptionfalse, we save all sub-caption counters at \donemaincaptionfalse so we can restore them using \caption@restorecounters inside \ContinuedFloat.

```
\caption@ifundefined\donemaincaptionfalse{}{%
  \g@addto@macro\donemaincaptionfalse\caption@savesubcounters}
```

14 Internal helpers

\caption@refstepcounter

Increments the float (i.e. figure or table) counter, resets the continuation counter, and redefines itself to \@gobble.

```
\newcommand*\caption@refstepcounter[1]{%
  \@ifundefined{c@#1}%
    {\caption@Error{No float type '#1' defined}}%
    {\caption@ref@stepcounter{#1}%
```

Set flag if this part of the figure (or table) contains a caption now. (If \caption@iftop is not set, \caption is at the bottom of this part, therefore ending this part, so we set the flag to false in this case.)

```
     \caption@fixposition
     \caption@iftop\caption@setflag\caption@clrflag{caption}%
```

This part does not contain content (like sub-figures) yet, so we set \ifcaption@subcaption to false.

```
     \caption@clrflag{subcaption}%
```

Support of the memoir document class.

```
     \@nameuse{donemaincaptiontrue}}}
\newcommand*\caption@ref@stepcounter{%
  \ifcaption@ContinuedFloat
    \let\caption@tempa\caption@@refcounter
    \caption@clrflag{ContinuedFloat}%
  \else
    \let\caption@tempa\caption@@refstepcounter
    \ifcaption@caption \else
      \ifcaption@subcaption
% Counter was already incremented by content, so we suppress \stepcounter{#1} here
        \let\caption@tempa\caption@@refcounter
      \fi
    \fi
  \fi
  \caption@tempa}
\newcommand*\caption@@refcounter[1]{%
  \let\caption@stepcounter@ORI\stepcounter
  \def\stepcounter##1{%
    \def\caption@tempa{#1}%
    \def\caption@tempb{##1}%
    \ifx\caption@tempa\caption@tempb \else
      \caption@stepcounter@ORI{##1}%
    \fi}%
   \caption@@@refstepcounter{#1}%
   \let\stepcounter\caption@stepcounter@ORI}
\newcommand*\caption@@refstepcounter[1]{%
  \caption@prepare@stepcounter{#1}{ref}%
  \caption@@@refstepcounter{#1}}
\newcommand*\caption@@stepcounter[1]{%
  \caption@prepare@stepcounter{#1}{}%
  \caption@@@stepcounter{#1}}
\newcommand*\caption@prepare@stepcounter[2]{%
  \caption@addsubcontentslines{#2stepcounter}%
  \caption@resetContinuedFloat{#1}}
\newcommand*\caption@@@refstepcounter{\refstepcounter}
\newcommand*\caption@@@stepcounter{\stepcounter}
```

\caption@dblarg A \relax was added compared to \@dblarg so \caption{} will be expanded to \caption[\relax]{} (and not to \caption[]{}). Furthermore support for option list-entry was added.

```
\@ifundefined{kernel@ifnextchar}{\let\kernel@ifnextchar\@ifnextchar}{}
```

```
\newcommand\caption@dblarg[1]{%
  \kernel@ifnextchar[{\caption@ydblarg{#1}}{\caption@xdblarg{#1}}}
```

```
\newcommand\caption@xdblarg[2]{%
  #1[{#2\relax}]{#2}}
```

```
\long\def\caption@ydblarg#1[#2]#3{%
  \caption@iflistheading{#1[{#3}]{#3}}{#1[{#2}]{#3}}}
```

`\caption@begin`

Our handling of `\caption` will always be surrounded by `\caption@begin` (or `\caption@beginex`) and `\caption@end`.
`\caption@begin{`⟨*type*⟩`}` performs these tasks:

1. Override the `position=` setting, if necessary. (for example if set to `auto` or used inside a `supertabular`)
2. Start a new group.
3. Define `\fnum@`⟨*type*⟩ if the caption label format is set to non-default.

```
\newcommand*\caption@begin[1]{%
  \caption@fixposition
  \begingroup
    \caption@setfnum{#1}}
```

`\caption@beginex`

`\caption@beginex{`⟨*type*⟩`}{`⟨*list entry*⟩`}{`⟨*heading*⟩`}`
performs the same tasks as `\caption@begin` and additionally:

4. Set `\lst@@caption`, so `\fnum@lstlisting` will include a numbering.
5. Make an entry in the list-of-whatever.
6. Set `\caption@ifempty` according argument ⟨*heading*⟩.

```
\newcommand\caption@beginex[3]{%
  \caption@begin{#1}%
  \let\lst@@caption\relax
  \caption@addcontentsline{#1}{#2}%
  \caption@ifempty{#3}{}}
```

`\caption@end`

`\caption@end` closes the group.

```
\newcommand*\caption@end{%
  \endgroup}
```

`\caption@setfnum`

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

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

\caption@boxrestore

The original code (from `latex/base/ltboxes.dtx`):

```
\def\@parboxrestore{\@arrayparboxrestore\let\\\@normalcr}
\def\@arrayparboxrestore{%
  \let\if@nobreak\iffalse
  \let\if@noskipsec\iffalse
  \let\par\@@par
  \let\-\@dischyph
  \let\’\@acci\let\‘\@accii\let\=\@acciii
  \parindent\z@ \parskip\z@skip
  \everypar{}%
  \linewidth\hsize
  \@totalleftmargin\z@
  \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
  \parfillskip\@flushglue \lineskip\normallineskip
  \baselineskip\normalbaselineskip
  \sloppy}
```

This one will be used by `\@caption` instead of `\@parboxrestore`.

```
\newcommand*\caption@boxrestore{%
  \caption@parboxrestore{\@parboxrestore}{%
    \let\if@nobreak\iffalse
    \let\if@noskipsec\iffalse
    \let\par\@@par
%    \let\-\@dischyph
%    \let\’\@acci\let\‘\@accii\let\=\@acciii
    \parindent\z@ \parskip\z@skip
    \everypar{}%
%    \linewidth\hsize
%    \@totalleftmargin\z@
    \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
    \parfillskip\@flushglue \lineskip\normallineskip
    \baselineskip\normalbaselineskip
    \sloppy
    \let\\\@normalcr
  }}
```

\caption@boxrestore@mini

Resets `\par` so the very first `\par` in `\@caption` behaves quite the same as in floating environments. Will be used by `\setcaptiontype`.

```
\newcommand\caption@boxrestore@mini{%
  \let\par\@@par
  \parindent\z@ \parskip\z@skip
  \sloppy}
```

\caption@normalsize

This one will be used by `\@caption` instead of `\normalsize`.
Its code is equivalent to

```
    \caption@font{normal}%
```

but executes faster (since the starred form of \caption@font does not use \setkeys internally).

```
\newcommand*\caption@normalsize{%
  \caption@font*{\KV@caption@fnt@normal\@unused}}
```

\caption@setfloatcapt

Needed for support of the float package, where the caption will not be typeset directly, but catched in a \vbox called \@floatcapt instead.

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

\caption@makecurrent

This one is needed for support of the nameref and hyperref package.

```
\newcommand*\caption@makecurrent[1]{\caption@gettitle}
```

\caption@makeanchor
\caption@start
\caption@@start
\caption@freezeHref
\caption@defrostHref

All these are needed for support of the hyperref package.

```
\let\caption@makeanchor\@firstofone
\let\caption@start\relax
\let\caption@@start\relax
\let\caption@freezeHref\relax
\let\caption@defrostHref\relax
```

\caption@gettitle

This one is needed for support of the nameref package.

```
\newcommand\caption@gettitle[1]{%
  \caption@ifundefined\NR@gettitle
    {\def\@currentlabelname{#1}}%
    {\NR@gettitle{#1}}}
```

15 Support for sub-captions

\caption@DeclareSubType

\caption@DeclareSub initializes the usage of \caption in sub-floats.

```
\def\caption@DeclareSubType sub#1\@nil{%
  \caption@Debug{Initializing subtype for `#1'\@gobble}%
  \@namedef{caption@beginsub#1}{\caption@beginsubfloat{#1}}}
\@onlypreamble\caption@DeclareSubType
```

Initialize the sub-captions defined with \DeclareCaptionSubType…

```
\caption@For*{subtypelist}{\caption@DeclareSubType sub#1\@nil}
```

Initialize the sub-captions defined with \newsubfloat[**?**]…

```
\caption@AtBeginDocument*{%
  \caption@ifundefined\sf@counterlist{}{%
    \@for\sf@temp:=\sf@counterlist\do{%
      \expandafter\caption@DeclareSubType\sf@temp\@nil}}}
```

\caption@subtypehook

Hook, will be used inside \caption@setsubtype.
(Note: If we are inside an subfloatrow environment we have to keep the \@makecaption code of the floatrow package intact.)

```
\newcommand*\caption@subtypehook{%
  \ifx\caption\caption@subcaption \else
    \caption@warmup
    \ifcaption@caption \else
% no \caption in this part of the (floating) environment yet
      \let\caption@add@contentsline\caption@addsubcontentsline
      \let\caption@addsubcontentslines\@gobble
      \ifcaption@subcaption \else
% no \subcaption in this part of the (floating) environment yet
        \ifcaption@ContinuedFloat
          \caption@clrflag{ContinuedFloat}%
        \else
          \caption@@stepcounter\@captype
        \fi
        \caption@setflag{subcaption}%
      \fi
    \fi
```

```
    \c@ContinuedFloat=0\relax
    \let\caption@setfloatcapt\@firstofone
```

```
    \caption@setbox{none}%
    \caption@clearmargin
    \caption@iflist{}{\let\caption@setlist\@gobble}%
    \caption@setoptions{sub}%
    \caption@setoptions{subfloat}% for subfig-package compatibility
```

```
% redefine \setcaptiontype
    \def\caption@settype{\caption@withoptargs\caption@sub@settype}%
    \def\caption@sub@settype##1##2{%
      \def\caption@tempa{##2}%
      \ifx\caption@tempa\@captype
%%%     \caption@setsubtype##1\relax
      \else
        \caption@Error{##2 inside \@subcaptype}%
      \fi}%
```

```
% redefine \caption
    \let\caption\caption@subcaption
    \let\phantomcaption\caption@subphantom
```

```
% restore \@makecaption
    \if@subfloatrow
      \caption@Debug{Keeping \string\@makecaption}%
    \else
      \let\@makecaption\caption@makecaption
    \fi
  \fi}%
```

`\if@subfloatrow` This macro tests if we are inside an `subfloatrow` or `subfloatrow*` environment.

```
\caption@AtBeginDocument{%
  \caption@ifundefined\@subfloatrowtrue
   {\newif\if@subfloatrow
    \caption@ifundefined\subfloatrow{}%
    {\caption@Debug{Patching subfloatrow environment}%
     \g@addto@macro\capsubrowsettings{\@subfloatrowtrue}%
     \g@addto@macro\killfloatstyle{%
       \ifx\c@FRobj\c@FRsobj\@subfloatrowtrue\fi}}}%
   {\caption@Debug{\string\if@subfloatrow is already defined}}}%
```

\caption@subcaption

Makes a sub-caption.

```
\newcommand*\caption@subcaption{%
  \caption@checkgrouplevel{sub}\subcaption
  \caption@star
    {\caption@@@refstepcounter\@subcaptype}%
    {\caption@dblarg{\@caption\@subcaptype}}}
```

\caption@subphantom

Same as \phantomcaption, but for sub-captions.

```
\newcommand*\caption@subphantom{%
  \caption@checkgrouplevel{sub}\phantomsubcaption
  \caption@@@refstepcounter\@subcaptype}
```

ption@clearsubcontentslines

Clear pending sub-caption list entries.

```
\newcommand*\caption@clearsubcontentslines{%
  \global\let\caption@subcontentslines\@empty}
```

```
\caption@clearsubcontentslines
```

\caption@addsubcontentsline

Add a pending sub-caption list entry.

```
\newcommand*\caption@addsubcontentsline[4]{%
  \caption@Debug{\string\caption@addsubcontentsline{#1}{#2}}%
  \begingroup
  \let\label\caption@gobble
  \let\index\caption@gobble
  \let\glossary\caption@gobble
  \protected@edef\@tempa{\endgroup
    \noexpand\g@addto@macro\noexpand\caption@subcontentslines{%
      \noexpand\@namedef{the#2}{\csname the#2\endcsname}%
      \ifx\@currentHref\@undefined \else
        \noexpand\def\noexpand\@currentHref{\@currentHref}%
      \fi
      \noexpand\caption@@@addcontentsline{#1}{#2}{#3}{#4}}}%
  \@tempa}
```

\flushsubcaptionlistentries

Writes pending sub-caption list entries.

```
\newcommand*\flushsubcaptionlistentries{%
  \caption@addsubcontentslines{user}}
```

```
\renewcommand*\caption@addsubcontentslines[1]{%
  \caption@Debug{\string\flushsubcaptionlistentries (#1)}%
  \begingroup
    \caption@subcontentslines
  \endgroup
  \caption@clearsubcontentslines}
```

We need to patch \chapter otherwise a vertical gap will be inserted into the list prior pending sub-caption list entries. We do this \AtBeginDocument so packages like newfloat will not complain about an unknown document class.

```
\AtBeginDocument{\caption@ifundefined\chapter{}{%
  \let\caption@chapter@ORI\chapter
  \def\chapter{%
    \caption@addsubcontentslines{chapter}\caption@chapter@ORI}}}
```

Same for \appendix.

```
\AtBeginDocument{\caption@ifundefined\appendix{}{%
  \let\caption@appendix@ORI\appendix
  \def\appendix{%
    \caption@addsubcontentslines{appendix}\caption@appendix@ORI}}}
```

Flush the list of pending sub-caption list entries at the end of the document.

```
\AtEndDocument{%
  \caption@addsubcontentslines{AtEndDocument}}
```

16 Document class & Babel package support

16.1 The $\mathcal{AMS}$ & SMF classes

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

16.2 The beamer class

```
\@ifclassloaded{beamer}{%
  \caption@InfoNoLine{beamer document class}%
```

\figure
\table

We redefine figure & table so our type-specific options will be used etc.

```
  \expandafter\let\expandafter\caption@ORI@figure
    \csname\string\figure\endcsname
  \@namedef{\string\figure}[#1]{%
    \caption@ORI@figure[#1]%
    \caption@settype{figure}}

  \expandafter\let\expandafter\caption@ORI@table
    \csname\string\table\endcsname
  \@namedef{\string\table}[#1]{%
    \caption@ORI@table[#1]%
    \caption@settype{table}}

}{}
```

16.3 The KOMA-Script classes

KOMA-Script contains the code \AtBeginDocument{\let\scr@caption\caption} so we need to update \scr@caption here, too.

```
\caption@ifundefined\scr@caption{}{%
  \caption@AtBeginDocument{\let\scr@caption\caption}}
```

16.4 The frenchb Babel option

Suppress “Package frenchb.ldf Warning: The definition of \@makecaption has been changed, frenchb will NOT customize it.” (but only if we emulate this customization)

```
\@nameuse{caption@frenchb}\@nameundef{caption@frenchb}
```

16.5 The frenchle/pro package

```
\caption@AtBeginDocument{\caption@ifundefined\frenchTeXmods{}{%
  \caption@InfoNoLine{frenchle/pro package is loaded}%
```

```
  \let\captionfont@ORI\captionfont
  \let\captionlabelfont@ORI\captionlabelfont
  \let\@makecaption@ORI\@makecaption
```

If \GOfrench is defined as \relax all the re-definitions regarding captions have already been done, so we can do our patches immediately. Otherwise we must add our stuff to \GOfrench.

```
  \caption@ifundefined\GOfrench
    {\let\caption@tempa\@firstofone}%
    {\def\caption@tempa{\g@addto@macro\GOfrench}}%
  \caption@tempa{%
```

```
    \let\captionfont\captionfont@ORI
    \let\captionfont@ORI\@undefined
    \let\captionlabelfont\captionlabelfont@ORI
    \let\captionlabelfont@ORI\@undefined
    \let\@makecaption\@makecaption@ORI
    \let\@makecaption@ORI\@undefined
```

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

```
    \let\@cnORI\caption
```

\@tablescaption The frenchle/pro package sets \caption to \@tablescaption at \begin{table} for special treatment of footnotes. Therefore we have to patch \@tablescaption so \caption* will work inside the table environment.

```
    \let\caption@tcORI\@tablescaption
    \def\@tablescaption{\caption@star\relax\caption@tcORI}%
```

\f@ffrench \f@tfrench \f@ffrench and \f@tfrench reflect \fnum@figure and \fnum@table when used in French mode. These contain additional code which typesets the caption separator \captionseparator instead of the usual colon. Because this breaks with our \@makecaption code we have to remove this additional code here.

```
    \let\@eatDP\@undefined
    \let\caption@tempa\@empty
    \ifx\f@ffrench\fnum@figure
      \l@addto@macro\caption@tempa{\let\fnum@figure\f@ffrench}%
    \fi
    \ifx\f@tfrench\fnum@table
      \l@addto@macro\caption@tempa{\let\fnum@table\f@tfrench}%
    \fi
    \def\f@ffrench{\ifx\listoffigures\relax\else\figurename~\thefigure\fi}%
    \def\f@tfrench{\ifx\listoftables\relax\else\tablename~\thetable\fi}%
    \caption@tempa

  }%
}}
```

16.6 The hungarian and magyar Babel option

```
\def\caption@tempa#1{%
  \@ifundefined{extras#1}\caption@AtBeginDocument\@firstofone{%
    \@ifundefined{extras#1}{}{%
      \caption@InfoNoLine{#1 babel option is loaded}%
      \expandafter\addto\csname extras#1\endcsname{%
         % reverse changes made by magyar.ldf
         \let\@makecaption\caption@makecaption
         \babel@save\@makecaption
         \caption@redefine
         \babel@save\@caption}%
    }}}
\caption@tempa{hungarian}%
\caption@tempa{magyar}%
```

17 Package support

\caption@IfPackageLoaded

\caption@IfPackageLoaded{⟨*package*⟩}[⟨*version*⟩]{⟨*true*⟩}{⟨*false*⟩}
Some kind of combination of \@ifpackageloaded and \@ifpackagelater. If the ⟨*package*⟩ is not loaded yet, the check will be (re-)done \AtBeginDocument, so the ⟨*package*⟩ could be loaded later on, too.

```
\newcommand\caption@IfPackageLoaded[1]{%
  \@testopt{\caption@@IfPackageLoaded{#1}}{}}
\@onlypreamble\caption@IfPackageLoaded

\long\def\caption@@IfPackageLoaded#1[#2]#3#4{%
  \@ifpackageloaded{#1}\@firstofone{%
    \caption@Debug{#1 package is not loaded (yet)\@gobble}%
    \caption@AtBeginDocument}{%
      \caption@If@Package@Loaded{#1}[#2]{#3}{#4}}}
\@onlypreamble\caption@@IfPackageLoaded

\long\def\caption@If@Package@Loaded#1[#2]{%
  \@ifpackageloaded{#1}{%
    \caption@InfoNoLine{#1 package is loaded}%
    \@ifpackagelater{#1}{#2}\@firstoftwo{%
      \caption@Error{%
        For a successful cooperation we need at least version\MessageBreak
          `#2' of package #1,\MessageBreak
        but only version\MessageBreak
          `\csname ver@#1.\@pkgextension\endcsname'\MessageBreak
        is available}%
      \@secondoftwo}%
  }{\@secondoftwo}}
\@onlypreamble\caption@If@Package@Loaded
```

\caption@clearmargin

This macro will be used by some package support stuff where the usual margin setting is not welcome, e.g. in the sidecap package.

```
\newcommand*\caption@clearmargin{%
  \setcaptionmargin\z@
  \let\caption@minmargin\@undefined}
```

```
\caption@setbool{needfreeze}{0}
\caption@AtBeginDocument*{%
  \caption@ifneedfreeze{%
```

\caption@freezetype

\caption@freezetype{⟨*type*⟩}

Used by the fltpage & sidecap package support.

```
    \newcommand*\caption@freezetype[1]{%
      \caption@settype*{#1}%
      \captionsetup*[sub]{hypcap=true}% Note: This is just a (q&d) workaround!
      \caption@freeze}%
```

\caption@freeze

\caption@freeze

Used by the bicaption package.

```
    \newcommand*\caption@freeze{%
      \let\caption@frozen@ContinuedFloat\ContinuedFloat
      \def\ContinuedFloat{%
        \caption@@freeze{\caption@@@ContinuedFloat\@captype}%
        \caption@frozen@ContinuedFloat}%
```

```
      \let\caption@frozen@setup\caption@setup
      \def\caption@setup##1{%
        \caption@@freeze{\caption@setup{##1}}%
        \caption@frozen@setup{##1}}%
```

```
      \let\caption@frozen@caption\caption
      \def\caption{%
        \def\caption{%
          \caption@Error{%
            Only one \noexpand\caption can be placed in this environment}%
          \caption@gobble}%
        \@ifstar
        {\caption@SC@caption*}%
        {\let\@currentlabel\caption@SClabel
         \caption@withoptargs\caption@SC@caption}}%
      \long\def\caption@SC@caption##1##2{%
        \caption@@freeze{\caption##1{##2}}%
        \ignorespaces}%
```

```
      \let\caption@frozen@label\label
      \def\label{%
```

```
    \caption@withoptargs\caption@SC@label}%
   \def\caption@SC@label##1##2{%
    \ifx\@currentlabel\caption@SClabel
     \@bsphack
      \caption@freeze@label{##1}{##2}%
     \@esphack
    \else
     \caption@frozen@label##1{##2}%
    \fi}%
   \def\caption@SClabel{\caption@undefinedlabel}%
   \def\caption@freeze@label##1##2{%
    \caption@@freeze{\label##1{##2}}}%

   \global\let\caption@frozen@content\@empty
   \long\def\caption@@freeze{%
    \g@addto@macro\caption@frozen@content}%

   \def\caption@warmup{%
    \let\ContinuedFloat\caption@frozen@ContinuedFloat
    \let\caption@setup\caption@frozen@setup
    \let\caption\caption@frozen@caption
    \let\label\caption@frozen@label}}%
```

\caption@prepare@defrost Prevent resetting the caption flags

```
  \newcommand*\caption@prepare@defrost{%
   \let\caption@settype\caption@set@type}
```

\caption@defrost `\caption@defrost`

```
  \newcommand*\caption@defrost{%
   \ifx\caption@frozen@caption\@undefined
    \caption@frozen@content
   \else
    \caption@Error{Internal Error:\MessageBreak
     \noexpand\caption@defrost in same group as \string\caption@freeze}%
   \fi}%

 }{}%
 \caption@undefbool{needfreeze}}
```

\caption@warmup `\caption@warmup`

```
\let\caption@warmup\relax
```

17.1 The float package

The float package usually do not use the LaTeX kernel command `\@caption` to typeset the caption but `\float@caption` instead. (`\@caption` will only be used if the float is re-styled with `\restylefloat*`.)
The main two things `\float@caption` is doing different are:

- The caption will be typeset inside a `\savebox` called `\@floatcapt` so it can be placed above or below the float contents afterwards.

- \@makecaption will not be used to finally typeset the caption. Instead \@fs@capt will be used which definition is part of the float style. (Note that \@fs@capt will not typeset any vertical space above or below the caption; instead this space will be typeset by the float style code itself.)

```
\caption@IfPackageLoaded{float}[2001/11/08 v1.3d]{%
 \@ifpackageloaded{floatrow}{%
  \caption@If@Package@Loaded{floatrow}[2007/08/24 v0.2a]{}{}%
 }{%
```

\@float@setevery

\@float@setevery{⟨*float type*⟩} is provided by the float package; it's called every time a floating environment defined with \newfloat or \restylefloat begins. We use this hook to do some adaptations and to setup the proper caption style (if defined) and additional settings declared with \captionsetup[⟨*float style*⟩].

```
 \let\caption@ORI@float@setevery\@float@setevery
 \def\@float@setevery#1{%
  \float@ifcaption{#1}{%
```

First of all we set the caption position to it's proper value by converting \@fs@iftopcapt (which is part of a float style and controls where the caption will be typeset, above or below the float contents) to our position= setting. Since the spacing above and below the caption will be done by the float style and *not* by us this sounds quite useless. But in fact it isn't, since some packages based on the caption package (like the subfig package) could have an interest for this information and therefore use the \caption@iftop macro we provide in our kernel. Furthermore we need this information for ourself in \captionof which uses \@makecaption to finally typeset the caption with skips.

```
   \caption@setposition{\@fs@iftopcapt t\else b\fi}%
```

Afterward we redefine \caption@setfloatcapt (which will be used inside \@caption) so the caption will be set inside the box \@floatcapt, without extra vertical space.

```
   \renewcommand\caption@setfloatcapt[1]{%
    \let\@makecaption\caption@@make
    \global\setbox\@floatcapt\vbox{%
     \color@begingroup ##1\color@endgroup}}%
```

To allow different caption styles for different float styles we also determine the current float style (e.g. 'ruled') and select a caption style (and additional settings) with the same name, if defined.

```
   \float@getstyle\float@style{#1}%
   \caption@setstyle*\float@style
   \caption@setoptions\float@style
```

```
  }{}%
  \caption@freezeHref % will be defrosted in \float@makebox
  \caption@ORI@float@setevery{#1}}%
```

\float@makebox

Redefine \float@makebox (only if we are not operating in compatibility mode).

```
 \caption@AtBeginDocument{\caption@ifcompatibility{}{%
```

If it was redefined by the hyperref package, we need to build on the original definition (and not on the redefined one).

```
    \caption@ifundefined\HyOrg@float@makebox
      {\let\caption@ORI@float@makebox\float@makebox}%
      {\let\caption@ORI@float@makebox\HyOrg@float@makebox}%
    \renewcommand\float@makebox[1]{%
      \caption@ORI@float@makebox{#1\relax \caption@defrostHref}}%
  }}%
```

`\caption@typehook` LaTeX and almost every other packages use `\`⟨*type*⟩`name` to provide a macro for the type resp. environment name – for example the command `\figurename` will usually contain the name of the floating environment `figure`:

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

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

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

```
  \g@addto@macro\caption@typehook{%
    \expandafter\ifx\csname #1name\endcsname\relax
      \expandafter\let\csname #1name\expandafter\endcsname
                      \csname fname@#1\endcsname
    \fi}%
```

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

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

`\float@getstyle` `\float@getstyle{`⟨*cmd*⟩`}{`⟨*type*⟩`}`

Determining the float style is not so easy because the only hint provided by the float package is the macro `\fst@`⟨*float type*⟩ which points to the macro which represents the float style. So for example after

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

`\fst@Program` will be defined as

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

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

TODO: We need to convert the catcodes here.

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

\float@ifcaption

\float@ifcaption{⟨*type*⟩}{⟨*if-clause*⟩}{⟨*else-clause*⟩}

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

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

```
}}{%
  \providecommand*\float@ifcaption[1]{\@secondoftwo}%
% \clearcaptionsetup{boxed}% used by the floatrow package?
}
```

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

```
\captionsetup[boxed]{skip=2pt} % do not issue a warning when not used
```

To emulate the ‘ruled’ definition of \@fs@capt we provide a caption style ‘ruled’ with appropriate options. But if the package option ruled was specified, we setup some caption parameters to emulate the behavior of the caption package *v1.x* option ruled instead, i.e., the current caption settings will be used, but without margin and without ‘single-line-check’.

```
\caption@ifbool{ruled}{%
  \captionsetup[ruled]{margin=0pt,minmargin=0,slc=0}%
}{%
  \DeclareCaptionStyle{ruled}{labelfont=bf,labelsep=space,strut=0}%
}
\caption@undefbool{ruled}
```

17.2 The floatflt package

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

\floatingfigure

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

```
  \let\caption@ORI@floatingfigure\floatingfigure
  \def\floatingfigure{%
```

```
    \caption@floatflt{figure}%
    \caption@ORI@floatingfigure}%
```

`\floatingtable` Same with `\floatingtable`…

```
  \let\caption@ORI@floatingtable\floatingtable
  \def\floatingtable{%
    \caption@floatflt{table}%
%   \caption@setautoposition b%
    \caption@ORI@floatingtable}%
```

`\caption@floatflt` Here we do two things:

1. We use `\caption@setoptions{floating`⟨*type*⟩`}` so `\captionsetup[floating`⟨*type*⟩`]{…}` is supported.
2. `\linewidth` must be set correctly. Usually this is done by `\@parboxrestore` inside `\@caption`, but since we use `\@caption@boxrestore` we have to map this to `\@parboxrestore` instead.

```
  \newcommand*\caption@floatflt[1]{%
    \caption@settype{#1}%
    \caption@clearmargin
    \caption@setfullparboxrestore
    \caption@setoptions{floating#1}}%
```

```
}{}
```

17.3 The fltpage package

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

`\FP@positionLabel` Original code:

```
\newcommand{\FP@positionLabel}{%
  FP\@captype-\number\value{FP@\@captype C}-pos}
```

```
  \renewcommand\FP@positionLabel{%
    FP\FP@captype-\number\value{FP@\FP@captype C}-pos}%
```

`\FP@helpNote` Original code:

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

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

\FP@floatBegin Original code:

```
\newcommand{\FP@floatBegin}[1]{%
  \gdef\@captype{#1}%
  \global\let\FP@savedCaptionCommand\caption%
  \global\let\FP@savedLabelCommand\label%
  \ifthenelse{\equal{\@captype}{figure}}
     {\global\let\old@Fnum\fnum@figure}%
     {\global\let\old@Fnum\fnum@table}%
  \let\FP@LabelText\@empty%
  \let\FP@CaptionText\@empty%
  \let\FP@optionalCaptionText\@empty%
  \renewcommand\label[1]{\gdef\FP@LabelText{##1}}%
  \renewcommand\caption[2][]{%
    \gdef\FP@optionalCaptionText{##1}\gdef\FP@CaptionText{##2}}%
  \begin{lrbox}{\FP@floatCorpusBOX}%
}%
```

```
\renewcommand*\FP@floatBegin[1]{%
  \def\FP@captype{#1}%
  \begin{lrbox}{\FP@floatCorpusBOX}%
  \minipage\hsize % changes from LR mode to vertical mode
  \caption@freezetype{#1}%
  \ignorespaces}%
```

\FP@floatEnd Original code:

```
\newcommand{\FP@floatEnd}{%
  \end{lrbox}%
  \global\setbox\FP@floatCorpusBOX=\box\FP@floatCorpusBOX
  \stepcounter{FP@\@captype C}%
  \FP@savedLabelCommand{\FP@positionLabel}%
  \FP@helpNote{\@captype}{\FP@positionLabel}%
  \FP@float
    {\FP@positionLabel}% location label test
    {\begin{\@captype}[p!]
       \usebox{\FP@floatCorpusBOX}%
       \refstepcounter{\@captype}%
       \ifthenelse{\equal{\FP@LabelText}{\@empty}}
         {}{\FP@savedLabelCommand{\expandafter\protect\FP@LabelText}}%
     \end{\@captype}}
    {\addtocounter{\@captype}{-1}}
    {\begin{\@captype}[b!]%
       \ifthenelse{\equal{\FP@guide}{\@empty}}%
         {}{\ifthenelse{\equal{\@captype}{figure}}%
             {\renewcommand{\fnum@figure}{\old@Fnum\ {\FP@guide}}}%
             {\renewcommand{\fnum@table}{\old@Fnum\ {\FP@guide}}}}%
       \setlength{\abovecaptionskip}{2pt plus2pt minus 1pt} % length above caption
       \setlength{\belowcaptionskip}{2pt plus2pt minus 1pt} % length above caption
       \FP@separatorCaption%
       \ifthenelse{\equal{\FP@optionalCaptionText}{\@empty}}%
         {\FP@savedCaptionCommand{\expandafter\protect\FP@CaptionText}}%
         {\FP@savedCaptionCommand[\expandafter\protect\FP@optionalCaptionText]%
                                 {\expandafter\protect\FP@CaptionText}}%
     \end{\@captype}}%
}%
```

```
  \renewcommand*\FP@floatEnd{%
    \endminipage
    \end{lrbox}%
    \stepcounter{FP@\FP@captype C}%
    \caption@label\FP@positionLabel
    \FP@helpNote\FP@captype\FP@positionLabel
    \FP@float
      {\FP@positionLabel}% location label test
      {\caption@prepare@defrost
       \begin\FP@captype[p!]%
         \usebox\FP@floatCorpusBOX
       \end\FP@captype}%
      {\@ifundefined{theH\FP@captype}{}{%
         \expandafter\l@addto@macro\csname theH\FP@captype\endcsname{.FP}}}%
      {\caption@prepare@defrost
       \begin\FP@captype[b!]%
         \let\FP@savedSetfnumCommand\caption@setfnum
         \def\caption@setfnum##1{%
           \FP@savedSetfnumCommand{##1}%
           \ifx\FP@guide\@empty \else
             \expandafter\l@addto@macro\csname fnum@##1\endcsname{\ {\FP@guide}}%
           \fi}%
         \setlength\abovecaptionskip{2pt plus 2pt minus 1pt}% length above caption
         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below caption
         \caption@setoptions{FP\@captype}%
         \FP@separatorCaption
         \caption@defrost
       \end\FP@captype}%
  }%
}{%
  \let\caption@ifFPlistcap\@undefined
  \let\caption@ifFPrefcap\@undefined
}
```

17.4 The hyperref package

```
\caption@IfPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
  % Test if hyperref has stopped early
  \caption@ifundefined\IfHyperBoolean{%
    \caption@set@bool\caption@ifhyp@stoppedearly0%
    \caption@ifundefined\H@refstepcounter
      {\caption@set@bool\caption@ifhyp@stoppedearly1}{%
    \caption@ifundefined\hyper@makecurrent
      {\caption@set@bool\caption@ifhyp@stoppedearly1}{%
    \caption@ifundefined\measuring@true
      {\caption@set@bool\caption@ifhyp@stoppedearly1}{}}}%
  }{%
    \def\caption@ifhyp@stoppedearly{\IfHyperBoolean{stoppedearly}}%
  }%
  \caption@ifhyp@stoppedearly{% hyperref has stopped early
    \caption@InfoNoLine{%
      Hyperref support is turned off\MessageBreak
      because hyperref has stopped early}%
  }{%
    \g@addto@macro\caption@prepareslc{\measuring@true}%
```

\caption@@@refstepcounter

We redefine \caption@@refstepcounter so \H@refstepcounter will be used instead of \refstepcounter inside \caption & \captionlistentry.

```
  \renewcommand*\caption@@@refstepcounter{\H@refstepcounter}%
```

\caption@makecurrent

We redefine \caption@makecurrent so a hyperref label will be defined inside \@caption.
Note: Will be redefined by \caption@start.

```
  \renewcommand*\caption@makecurrent[2]{%
    \caption@makecurrentHref{#1}%
    \caption@Debug{hyperref current=\@currentHref}%
    \caption@gettitle{#2}}%
```

```
  \newcommand*\caption@makecurrentHref{\hyper@makecurrent}%
```

\caption@makeanchor

We redefine \caption@makeanchor so a hyperref anchor will be set inside \@caption.
Note: Will be redefined by \caption@start.

```
  \renewcommand\caption@makeanchor[1]{%
    \caption@Debug{hyperref anchor: \@currentHref}%
    % If we cannot have nesting, the anchor is empty.
    \ifHy@nesting
      \expandafter\hyper@@anchor\expandafter{\@currentHref}{#1}%
    \else
      \Hy@raisedlink{%
        \expandafter\hyper@@anchor\expandafter{\@currentHref}{\relax}%
      }#1%
    \fi}%
```

```
  \g@addto@macro\caption@prepareslc{\let\caption@makeanchor\@firstofone}%
```

The hypcap option

\if@capstart

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

```
  \newif\if@capstart
```

\caption@start

While the hypcap package defines a macro called `\capstart` our variant is called `\caption@start` and is controlled by the option `hypcap=false`/`true`.

```
  \def\caption@start{\caption@ifhypcap\caption@start@\relax}%
  \def\caption@start@{%
```

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

```
    \caption@makestart\@captype
    \caption@startanchor\@currentHref
```

Prevent `\@caption` from generating a new hyperref label, use the label we save in `\hc@currentHref` instead. (We also support the `@capstart` flag from the hypcap package.)

```
    \global\@capstarttrue
    \let\hc@currentHref\@currentHref
    \def\caption@makecurrentHref##1{%
      \global\@capstartfalse
      \global\let\@currentHref\hc@currentHref}%
```

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

```
    \let\caption@makeanchor\@firstofone
  }%
```

\caption@makestart

`\caption@makestart{`⟨*type*⟩`}` defines a hyperref anchor inside `\caption@start`. Since we offer `\ContinuedFloat` the float counter can change between ‘now’ and `\caption`, i.e., we simply don’t know the figure or table counter yet and therefore we are not able to generate the ‘right’ hyperref label. Two different solutions of this problem came into my mind:

1. I could use the aux file for this purpose.

   -or-

2. I set `hypertexnames=false` locally. Furthermore I use `#1.caption.`⟨*counter*⟩ (instead of `#1.`⟨*counter*⟩) as naming scheme for `\@currentHref` to avoid conflicts with other hyper links which are generated with `hypertexnames=true`.

The first idea has the advantage that the ‘right’ anchor name will be generated, but one needs an additional LaTeX run if figures or tables will be inserted or removed.
The second idea has the advantage that it’s very easy to implement, but has some side-effects, e.g. the anchor names don’t follow the figure or table label names anymore.
Since I’m lazy I implemented the second idea, maybe I will revise this later on.

```
  \newcommand*\caption@makestart[1]{%
    \begingroup
      \Hy@hypertexnamesfalse
%     \gdef\@currentHlabel{}%
      \hyper@makecurrent{#1.caption}%
    \endgroup
    \caption@Debug{hypcap start=\@currentHref}}%
```

\caption@startanchor

`\caption@startanchor{`⟨*Href*⟩`}` sets a hyperref anchor inside `\caption@start`. This code was taken from the hypcap package[**?**] and adapted.

Note: Since `\hyper@@anchor{`⟨*Href*⟩`}{\relax}` can cause a change from vertical mode to horizontal mode (design flaw in hyperref package!?), and since the workaround `\let\leavevmode\relax` which can be found in the hypcap package is not always sufficient (for example with "Direct pdfmark support" and `breaklinks=true`), we use `\caption@anchor` instead of `\hyper@@anchor` here.

```
\newcommand*\caption@startanchor[1]{%
  \ifvmode\begingroup
    \caption@Debug{hypcap anchor: #1 (vertical mode)}%
    \@tempdima\prevdepth
    \nointerlineskip
    \vspace*{-\caption@hypcapspace}%
    \caption@anchor{#1}%
    \vspace*{\caption@hypcapspace}%
    \prevdepth\@tempdima
  \endgroup\else
    \caption@Debug{hypcap anchor: #1 (horizontal mode)}%
    \caption@anchor{#1}%
  \fi}%
```

\caption@anchor

`\caption@anchor{`⟨*Href*⟩`}` sets a hyperref anchor.

```
\newcommand*\caption@anchor[1]{%
  \ifmeasuring@ \else
    \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
  \fi}%
```

Note: Since `\Hy@raisedlink` change `\@tempdima` we surrounded it by `\ifvmode`, suppressing "`LaTeX Warning:  Float too large for page by 1.0pt`" in `sideways` floats. (This is not necessary since hyperref *v6.77*.)

```
\ifx\HyperRaiseLinkLength\@tempdima
  \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
\else
  \let\caption@raisedlink\Hy@raisedlink
\fi
```

\caption@@start

Will be used by `\caption@freezeHref`. Apart from that we issue a warning if we expect a saved hyperref label coming from `\caption@start`, but there isn't any.

```
\def\caption@@start{%
  \caption@ifundefined\hc@currentHref{%
    \caption@Warning{%
      The option 'hypcap=true' will be ignored for this\MessageBreak
      particular \string\caption}}{}}%
```

\caption@freezeHref

Suppress `\caption@start` from generating a hyperref label and setting a hyperref anchor. Instead if `\@caption` generates a hyperref label, it will be stored in `\caption@currentHref`. Furthermore we need to redefine `\caption@setfloatcapt` so no hyperref anchor will be placed in `\@caption`.

```
    \def\caption@freezeHref{%
      \let\caption@ORI@start\caption@start
      \def\caption@start{\let\caption@start\caption@ORI@start}%
%     \let\caption@ORI@@start\caption@@start
%     \l@addto@macro\caption@subtypehook{%
%       \let\caption@@start\caption@ORI@@start}%
      \global\let\caption@currentHref\@undefined
      \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
      \let\caption@ORI@setfloatcapt\caption@setfloatcapt
      \renewcommand*\caption@setfloatcapt{%
        \ifx\caption@currentHref\@undefined \else
          \let\caption@makeanchor\@firstofone
        \fi
        \caption@ORI@setfloatcapt}}%
```

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

```
    \def\caption@defrostHref{%
      \ifx\caption@currentHref\@undefined \else
        \caption@startanchor\caption@currentHref
        \global\let\caption@currentHref\@undefined
      \fi}%
  }}{}
```

17.5 The hypcap package

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

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

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

```
    \let\caption@ORI@capstart\capstart % save for compatibility mode
    \caption@ifundefined\capstarttrue % check for v1.10 of hypcap package
      {\def\capstart{\caption@start@}}%
      {\def\capstart{\ifcapstart\caption@start@\fi}}%
    \let\caption@start\relax
    \let\caption@@start\relax
```

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

```
    \caption@set@bool\caption@ifhypcap 1%
    \renewcommand*\caption@hypcapspace{\hypcapspace}%
  \fi}{}
```

17.6 The listings package

```
\caption@IfPackageLoaded{listings}[2004/02/13 v1.2]{%
```

\lst@MakeCaption To support the listings package we need to redefine \lst@MakeCaption so the original stuff is nested with \caption@begin and \caption@end etc.

Note: This macro is always called twice (with ‘t’ resp. ‘b’ as parameter), therefore we need an extra group here.

```
  \let\caption@ORI@lst@MakeCaption\lst@MakeCaption
  \def\lst@MakeCaption#1{% #1 is ‘t’ or ‘b’
    \begingroup
```

Workaround for bug in listings package: If \hsize seems not to be set correctly, we set it to \linewidth.

```
    \ifdim\hsize>\linewidth
      \hsize\linewidth
    \fi
```

First of all, we set position=#1 and if it was set to ‘top’, we swap the skips so the default behavior of the listings package will not be changed. (Note that the listings package has set its own \abovecaptionskip & \belowcaptionskip values prior to calling \lst@MakeCaption.)

```
      \caption@setposition{#1}%
      \caption@iftop{%
        \@tempdima\belowcaptionskip
        \belowcaptionskip\abovecaptionskip
        \abovecaptionskip\@tempdima}{}%
```

Workaround for issue with wrong skips (should be examined further)

```
      \caption@setup{rule=0}%
```

Afterwards we set the local ‘lstlisting’ options.

```
      \caption@setoptions{lstlisting}%
```

If the position= is now set to auto, we take over the captionpos= setting from the listings package.

```
      \caption@setautoposition{#1}%
```

At the end we do similar stuff as in our \@caption code.

```
      \caption@begin{lstlisting}%
        \caption@ORI@lst@MakeCaption{#1}%
      \caption@end
    \endgroup}%
```

\lst@makecaption
\lst@maketitle Wrapper macros for typesetting the caption= resp. title= value.

```
  \def\lst@makecaption{\caption@starfalse\@makecaption}%
  \def\lst@maketitle{\caption@startrue\@makecaption\@empty}%
```

\ext@lstlisting Since the listings package do not define \ext@lstlisting but we needed it when \captionof{lstlisting} will be done by the end user, we define it here.

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

17.7 The longtable package

\LTcaptype

\LTcaptype is preset to table.

```
\providecommand*\LTcaptype{table}
```

```
\caption@IfPackageLoaded{longtable}[1995/05/24 v3.14]{%
```

```
  \RequirePackage{ltcaption}[2007/09/01]%
  \let\LT@@makecaption\@undefined
```

\LT@array

We redefine \LT@array here to get \captionsetup{⟨*options*⟩} working inside longtables.

Note: Since the hyperref package patches \LT@array as well and since this only works with the original definition of \LT@array, we have to do this after the hyperref package, i.e. \AtBeginDocument.

```
  \caption@AtBeginDocument{%
    \let\caption@ORI@LT@array\LT@array
    \renewcommand*\LT@array{%
```

\captionsetup for longtable:

```
      \global\let\caption@opt@@longtable\@undefined
      \def\captionsetup{%
        \noalign\bgroup
        \@ifstar\@captionsetup\@captionsetup}% gobble *
      \def\@captionsetup##1{\LT@captionsetup{##1}\egroup}%
      \def\LT@captionsetup##1{%
        \captionsetup@startrue\caption@setup@options[@longtable]{##1}%
        \global\let\caption@opt@@longtable\caption@opt@@longtable}%
```

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

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

\captionlistentry for longtable:

```
      \def\captionlistentry{%
        \noalign\bgroup
        \@ifstar{\egroup\LT@captionlistentry}% gobble *
                {\egroup\LT@captionlistentry}}%
      \def\LT@captionlistentry##1{%
        \caption@listentry\@firstoftwo[\LTcaptype]{##1}}%
```

\ContinuedFloat for longtable:

(Commented out, since it's not deeply tested and quite useless anyway)

Note: hyperref versions < v6.76j uses 2× \hyper@makecurrent

```
%      \caption@ifhypcap{%
%        \let\caption@ORI@hyper@makecurrent\hyper@makecurrent
%        \def\hyper@makecurrent##1{%
%          \let\hyper@makecurrent\caption@ORI@hyper@makecurrent
%          \caption@makestart{##1}%
%%          \let\Hy@LT@currentHlabel\@currentHlabel
%          \let\Hy@LT@currentHref\@currentHref
%         \def\hyper@makecurrent####1{%
%%          \let\@currentHlabel\Hy@LT@currentHlabel
%           \let\@currentHref\Hy@LT@currentHref}}%
%       \let\caption@ORI@ContinuedFloat\ContinuedFloat
%       \def\ContinuedFloat{\noalign{%
%         \gdef\caption@setContinuedFloat{%
%           \let\caption@resetContinuedFloat\@gobble}%
%         \def\caption@setoptions####1{%
%           \g@addto@macro\caption@setContinuedFloat{%
%             \caption@setoptions{####1}}}%
%         \let\@captype\LTcaptype
%         \caption@ORI@ContinuedFloat}}%
%     }{%
%       \def\ContinuedFloat{\noalign{%
%         \caption@Error{%
%           \noexpand\ContinuedFloat inside longtables\MessageBreak
%           is only available with ‘hypcap=true’}}}%
%     }%
%     \global\let\caption@setContinuedFloat\@empty
      \def\ContinuedFloat{\noalign{%
        \caption@Error{\noexpand\ContinuedFloat outside float}}}%
      \caption@ORI@LT@array}}%
```

`\LT@c@ption`

The original implementation:

```
\def\LT@c@ption#1[#2]#3{%
  \LT@makecaption#1\fnum@table{#3}%
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else
     {\let\\\space
     \addcontentsline{lot}{table}{\protect\numberline{\thetable}{#2}}}%
  \fi}
```

Our implementation uses `\LTcaptype` instead of `{table}`:

```
  \long\def\LT@c@ption#1[#2]#3{%
    \LT@makecaption#1{\csname fnum@\LTcaptype\endcsname}{#3}%
    \LT@captionlistentry{#2}}%
```

`\LT@makecaption`

`\LT@makecaption{`⟨*cmd*⟩`}{`⟨*label*⟩`}{`⟨*text*⟩`}`

The original definition:

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

Our definition:

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

If `\LTcapwidth` is not set to its default value `4in` we assume that it shall overwrite our own setting. (But `\captionsetup[longtable]{width=...}` will overwrite `\LTcapwidth`.)

```
    \caption@settype*\LTcaptype
    \ifdim\LTcapwidth=4in \else
      \setcaptionwidth\LTcapwidth
    \fi
    \caption@setoptions{longtable}%
%   \caption@setContinuedFloat
    \caption@setoptions{@longtable}%
```

`position=auto` is a bad idea for longtables, but we do our very best. This works quite well for captions inside the longtable contents, but not for captions inside the longtable (end)foot.
Note: This should be 'top' if unclear!

```
    \caption@setautoposition{\ifcase\LT@rows t\else b\fi}%
```

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

```
    \caption@startrue#1\caption@starfalse
```

```
    \caption@prepare@stepcounter\LTcaptype{LT}%
```

```
    \caption@begin\LTcaptype
      \caption@normalsize
```

The following skip has the purpose to correct the height of the `\parbox[t]`. Usually it's the height of the very first line, but because of our extra skips (`\abovecaptionskip` and `\belowcaptionskip`) it's always `0pt`.
(A different idea would be typesetting the first skip outside the longtable column with `\noalign{\vskip...}`, but this means we have to move `\caption@begin` to some other place because it does not work in tabular mode. And at the moment I have no idea on how to do this in an elegant way...)

```
      \vskip-\ht\strutbox
```

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

```
      \caption@iftop{\vskip\belowcaptionskip}{\vskip\abovecaptionskip}%
      \caption@@make{#2}{#3}\endgraf
      \caption@iftop{\vskip\abovecaptionskip}{\vskip\belowcaptionskip}%
    \caption@end}}%
```

```
}{}
```

17.8 The picinpar package

```
\caption@IfPackageLoaded{picinpar}{%
```

\figwindow \tabwindow The picinpar package comes with its own caption code (`\wincaption`, `\@wincaption`, `\@makewincaption`, ...) so we redefine `\figwindow` & `\tabwindow` to use `\caption` instead.

```
  \long\def\figwindow[#1,#2,#3,#4] {%
    \caption@window{figure}%
    \caption@setoptions{figwindow}%
    \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
  \long\def\tabwindow[#1,#2,#3,#4] {%
    \caption@window{table}%
    \caption@setoptions{tabwindow}%
    \begin{window}[#1,#2,{#3},\caption@wincaption{#4}] }%
```

\caption@window Beside calling `\caption@settype` we redefine `\caption@boxrestore` (as in floatflt & picins package support) and `\@makecaption` (as in float package support) here.

```
  \newcommand*\caption@window[1]{%
    \let\@makecaption\caption@@make
    \caption@setautoposition b%
    \caption@settype{#1}%
    \caption@clearmargin
    \caption@setfullparboxrestore}%
```

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

```
  \newcommand\caption@wincaption[1]{%
```

This will be done twice for every `figwindow` & `tabwindow` caption – on the first run `\picwd` is `0pt`, on the second run `\picwd` is `\hsize`.

```
    \ifdim\picwd=\z@
      \let\caption@makecurrent\@gobbletwo
      \let\caption@@start\relax
      \caption@prepareslc
    \else
      \caption@ContinuedFloattrue
    \fi
```

The argument `#1` could contain simply the caption text (e.g. `A figure caption`), but it could also contain an optional argument, the ⟨*lst_entry*⟩ (e.g. `[An entry to the LOF]{A figure caption}`). Therefore we have to test if `#1` begins with `[` or not; furthermore we support a starred variant – as in `\caption*` – so we test for `*`, too.

```
    \edef\@tempa{\expandafter\noexpand\@car#1\@nil}%
    \if\@tempa*%
      \let\@tempa\@firstofone
    \else\if\@tempa[%]
      \let\@tempa\@firstofone
    \else
      \let\@tempa\@empty
    \fi\fi
    \expandafter\caption\@tempa{#1}}%
```

```
}{}
```

17.9 The picins package

\piccaptiontype

\piccaptiontype{⟨*type*⟩}
We offer this macro for changing the ⟨*type*⟩ of the caption, so the user doesn't have to redefine \@captype, as proposed in the picins documentation.
Note: We define this macro here so it can be used in the preamble of the document, even when the caption package was loaded prior to the picins package.

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

```
\caption@IfPackageLoaded{picins}{%
```

Initial set \@piccaptype and undefine \@captype which was set to figure by the picins package.

```
  \caption@ifundefined\@piccaptype{%
    \caption@iftype{%
      \let\@piccaptype\@captype
    }{%
      \def\@piccaptype{figure}%
    }%
  }{}%
  \let\@captype\@undefined
```

\piccaption

The original code:

```
\def\piccaption{\@ifnextchar [{\@piccaption}{\@piccaption[]}}
```

Our code uses \caption@star so \piccaption* works, and \caption@dblarg so \piccaption{} works correctly.

```
  \def\piccaption{\caption@star\relax{\caption@dblarg\@piccaption}}%
```

\make@piccaption

The original code:

```
\def\make@piccaption{%
 [...]
 \setbox\@TEXT=\vbox{\hsize\hsiz@\caption[\sh@rtf@rm]{\capti@nt@xt}}%
}
```

In our code we have to correct several things:

1. \@captype must be defined, since we have removed the global definition.
2. We use \caption@setoptions{parpic} so \captionsetup[parpic]{...} is supported.

3. `\linewidth` must be set correctly. Usually this is done by `\@parboxrestore` inside `\@caption`, but since we use `\@caption@boxrestore` we have to map this to `\@parboxrestore` instead.

4. The two arguments of `\caption` (`\sh@rtf@rm` & `\capti@nt@xt`) should be expanded on first level so `\caption[]{...}` and `\caption[...]{}` work correctly.

```
\let\caption@ORI@make@piccaption\make@piccaption
\def\make@piccaption{%
  \let\caption@ORI\caption

  \long\def\caption[##1]##2{%
    \caption@freezeHref % will be defrosted in \ivparpic
    \caption@settype\@piccaptype
%    \ifnum\c@piccaptionpos>2\relax
    \caption@clearmargin
%    \else
%      \captionwidth\z@ % do not use "width=" setting
%    \fi
    \caption@setfullparboxrestore
    \caption@setoptions{parpic}%
    \caption@setautoposition b%

    \expandafter\expandafter\expandafter\caption@ORI
      \expandafter\expandafter\expandafter[%
      \expandafter\expandafter\expandafter{%
      \expandafter##1\expandafter}\expandafter]\expandafter{##2}}%
```

```
-or-  \begingroup
        \toks0\expandafter{##1} \toks2\expandafter{##2}
        \edef\x{\endgroup
          \noexpand\caption@ORI[{\the\toks0}]{\the\toks2}}
        \x

-or-  \edef\x{%
        \noexpand\caption@ORI[{\unexpanded\expandafter{##1}}]%
                              {\unexpanded\expandafter{##2}}}
      \-
```

```
  \caption@ORI@make@piccaption
  \let\caption\caption@ORI}%
```

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

```
\let\caption@ORI@ivparpic\ivparpic
\def\ivparpic(#1,#2)(#3,#4)[#5][#6]#7{%
  \let\caption@ORI@noindent\noindent
  \def\noindent{%
    \caption@defrostHref
    \let\noindent\caption@ORI@noindent
    \noindent}%
  \caption@ORI@ivparpic(#1,#2)(#3,#4)[#5][#6]{#7}%
  \let\noindent\caption@ORI@noindent}%
```

```
}{%
  \let\piccaptiontype\@undefined
}
```

17.10 The rotating package

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

\rotcaption Make \rotcaption* work.

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

\rotcaptionof Make \rotcaptionof(*) work.

```
  \def\rotcaptionof{%
    \caption@teststar\caption@of{\rotcaption*}\rotcaption}%
```

\@makerotcaption Original (bugfixed) code:

```
\long\def\@makerotcaption#1#2{%
  \setbox\@tempboxa\hbox{#1: #2}%
  \ifdim \wd\@tempboxa > .8\vsize
    \rotatebox{90}{%
    \begin{minipage}{.8\textheight}#1: #2\end{minipage}%
    }%\par   % <== \par removed (AR)
  \else%
    \rotatebox{90}{\box\@tempboxa}%
  \fi
  \nobreak\hspace{12pt}% <== \nobreak added (AR)
}
```

Our version emulates this behavior, but if width= is set, the rotated caption is always typeset as \parbox. (Note that margin= is not supported here.)

```
  \long\def\@makerotcaption#1#2{%
    \rotatebox{90}{%
      \ifdim\captionwidth=\z@
        \setcaptionwidth{.8\vsize}%
        \l@addto@macro\caption@singleline{%
          \caption@setup{parbox=none}}%
      \fi
      \let\caption@calcmargin\relax
      \caption@@make{#1}{#2}}%
    \nobreak\hspace{12pt}}%

}{}
```

17.11 The sidecap package

```
\caption@IfPackageLoaded{sidecap}[2003/06/06 v1.6f]{%
  \caption@setbool{needfreeze}{1}%
```

`\SC@zfloat` This macro will be called at the start of the environment, here is a good opportunity to do some adaptations to `\caption` and `\captionsetup`.

```
\let\caption@ORI@SC@zfloat\SC@zfloat
\def\SC@zfloat#1#2#3[#4]{%
```

First we use the original definition, but restore `\caption` and `\label` so `\caption@freeze` and `\caption@warmup` will work correctly.

```
  \caption@ORI@SC@zfloat{#1}{#2}{#3}[#4]%
  \SC@RestoreCommands
```

Since the sidecap package uses our `\caption` code outside the environment the regular `\captionsetup` will not work. So we need a special version here which saves the given argument list which will be executed later on. Furthermore we need to make `\caption*` work.

```
  \caption@freezetype{#2}%
```

The sidecap package uses `\ifx\label\SC@label` to test if it is just inside a `SCfigure` or not. So we redefine `\SC@label` here so this test will still work.

```
  \let\SC@label\label}%
```

```
\providecommand*\SC@RestoreCommands{%
  \let\caption=\SC@orig@caption \let\label=\SC@orig@label}%
```

`\endSC@FLOAT` This macro will be called at the end of the environment, here we need to setup our stuff before the sidecap package actually typesets its caption.

```
\let\caption@ORI@endSC@FLOAT\endSC@FLOAT
\def\endSC@FLOAT{%
```

Before we can typeset the caption we need to set the margin to zero because any extra margin would only be disturbing here.
(We don't need to take care about the caption position because the sidecap package set both `\abovecaptionskip` and `\belowcaptionskip` to a skip of zero anyway.)
Furthermore `\SC@justify` will override the caption justification, if set. The usage of `\SC@justify` differs from version to version of the sidecap package:

Version 1.4: `\SC@justify` is not defined
Version 1.5: `\SC@justify` is `\relax` when not set
Version 1.6: `\SC@justify` is `\@empty` when not set

```
  \def\caption@setSC@justify{%
    \caption@clearmargin
      \ifx\SC@justify\@empty \else
        \let\caption@hj\SC@justify
        \let\SC@justify\@empty
      \fi}%
```

Make the original definition of `\endSC@FLOAT` to use our caption stuff instead of its own.
Note: At this point the sidecap definition of `\caption` is valid, not the regular one!

```
  \let\caption\SC@orig@caption
  \def\SC@orig@caption[##1]##2{%
    \caption@setSC@justify
```

```
%%%   \caption@setoptions{SC}%
      \caption@setoptions{SC\@captype}%
      \caption@defrost}%
```

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

```
    \caption@setSC@justify % for compatibility mode
    \caption@prepare@defrost
    \caption@ORI@endSC@FLOAT}%
```

```
}{}
```

17.12 The subfigure package

```
\caption@IfPackageLoaded{subfigure}[2002/01/23 v2.1]{%
```

\sf@ifpositiontop

If the subfigure package is loaded, we map \sf@ifpositiontop to \iffiguretopcap resp. \iftabletopcap, so the subfigure *v2.1* options figbotcap etc. will still work.

```
  \def\sf@ifpositiontop{%
    \ifx\@captype\@undefined
      \expandafter\@gobbletwo
    \else\ifx\@captype\relax
      \expandafter\expandafter\expandafter\@gobbletwo
    \else
      \expandafter\expandafter\expandafter\sf@if@position@top
    \fi\fi}
```

```
  \def\sf@if@position@top{%
    \@ifundefined{if\@captype topcap}%
      {\@gobbletwo}%
      {\@nameuse{if\@captype topcap}%
         \expandafter\@firstoftwo
       \else
         \expandafter\@secondoftwo
       \fi}}
```

```
}{}
```

17.13 The supertabular and xtab packages

```
\caption@IfPackageLoaded{supertabular}[2002/07/19 v4.1e]{%
```

\tablecaption

Make \topcaption* and \bottomcaption* work.

```
  \renewcommand*\tablecaption{%
    \caption@star
      {\refstepcounter{table}}%
      {\caption@dblarg{\@xtablecaption}}}%
```

\@xtablecaption

Make \nameref and \autoref work.

```
  \let\caption@ORI@xtablecaption\@xtablecaption
  \long\def\@xtablecaption[#1]#2{%
    \caption@gettitle{#2}%
    \caption@ORI@xtablecaption[#1]{#2}}%
```

\ST@caption The original code:

```
\long\def\ST@caption#1[#2]#3{\par%
  \addcontentsline{\csname ext@#1\endcsname}{#1}%
                  {\protect\numberline{%
                      \csname the#1\endcsname}{\ignorespaces #2}}
  \begingroup
    \@parboxrestore
    \normalsize
    \if@topcaption \vskip -10\p@ \fi
    \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
    \if@topcaption \vskip 10\p@ \fi
  \endgroup}
```

```
  \long\def\ST@caption#1[#2]#3{\par%
    \caption@settype*{#1}%
    \caption@setoptions{supertabular}%
```

The position= setting will be overwritten by the supertabular package: If \topcaption was used, the position will be top automatically, bottom otherwise.

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

```
    \caption@beginex{#1}{#2}{#3}%
      \caption@boxrestore
      \caption@normalsize
      \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
    \caption@end}%
```

```
}{}
```

```
\caption@IfPackageLoaded{xtab}[2000/04/09 v2.3]{%
```

\tablecaption Make \topcaption* and \bottomcaption* work.

```
  \renewcommand*\tablecaption{%
    \caption@star
      {\refstepcounter{table}}%
      {\caption@dblarg{\@xtablecaption}}}%
```

\@xtablecaption Make \nameref and \autoref work.

```
  \let\caption@ORI@xtablecaption\@xtablecaption
  \long\def\@xtablecaption[#1]#2{%
    \caption@gettitle{#2}%
    \caption@ORI@xtablecaption[#1]{#2}}%
```

\ST@caption The original code:

```
\long\def\ST@caption#1[#2]#3{\par%
  \@initisotab
  \addcontentsline{\csname ext@#1\endcsname}{#1}%
                  {\protect\numberline{%
                    \csname the#1\endcsname}{\ignorespaces #2}}%
  \begingroup
    \@parboxrestore
    \normalsize
  %%  \if@topcaption \vskip -10\p@ \fi
    \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
  %%  \if@topcaption \vskip 10\p@ \fi
  \endgroup
  \global\advance\ST@pageleft -\PWSTcapht
  \ST@trace\tw@{Added caption. Space left for xtabular: \the\ST@pageleft}}
```

```
\long\def\ST@caption#1[#2]#3{\par%
  \caption@settype*{#1}%
  \caption@setoptions{xtabular}%
  \def\caption@fixposition{%
    \caption@setposition{\if@topcaption t\else b\fi}}%
  \@initisotab
  \caption@beginex{#1}{#2}{#3}%
    \caption@boxrestore
    \caption@normalsize
    \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
  \caption@end
  \global\advance\ST@pageleft -\PWSTcapht
  \ST@trace\tw@{Added caption. Space left for xtabular: \the\ST@pageleft}}%
}{}
```

17.14 The threeparttable package

```
\caption@IfPackageLoaded{threeparttable}[2003/06/13 v3.0]{%
```

\threeparttable Unfortunately \@captype is not set when \TPT@common will be used, so we have to redefine \threeparttable and \measuredfigure instead.

```
  \let\caption@ORI@threeparttable\threeparttable
  \renewcommand*\threeparttable{%
    \caption@settype{table}%
    \caption@setposition a% ?
    \caption@clearmargin
    \caption@setoptions{threeparttable}%
    \caption@ORI@threeparttable}%
```

\measuredfigure Same here…

```
  \let\caption@ORI@measuredfigure\measuredfigure
  \renewcommand*\measuredfigure{%
    \caption@settype{figure}%
    \caption@setposition a% ?
    \caption@clearmargin
    \caption@setoptions{measuredfigure}%
    \caption@ORI@measuredfigure}%
```

\TPT@caption The original code:

```
\def\TPT@caption#1[#2]#3{\gdef\TPT@docapt
 {\par\global\let\TPT@docapt\@undefined \TPT@LA@caption{#1}[{#2}]%
   {\strut\ignorespaces#3\ifhmode\unskip\@finalstrut\strutbox\fi}}%
 \ifx\TPT@hsize\@empty \let\label\TPT@gatherlabel \abovecaptionskip\z@skip
 \else \TPT@docapt \fi \ignorespaces}
```

```
  \def\TPT@caption#1[#2]#3{%
    \gdef\TPT@docapt{%
      \global\let\TPT@docapt\@undefined
      \caption@setautoposition\caption@TPT@position
      \TPT@LA@caption{#1}[{#2}]{#3}}%
    \ifx\TPT@hsize\@empty
      \let\label\TPT@gatherlabel % Bug: does not work for measuredfigures
      \gdef\caption@TPT@position{t}%
      \g@addto@macro\TPT@docapt\caption@TPT@eatvskip
    \else
      \def\caption@TPT@position{b}%
      \TPT@docapt
    \fi
    \ignorespaces}%
  %\newcommand*\caption@TPT@eatvskip{\vskip-.2\baselineskip}%
  \def\caption@TPT@eatvskip#1\vskip{#1\@tempdima=}%
}{}
```

17.15 The wrapfig package

```
\caption@IfPackageLoaded{wrapfig}[2003/01/31 v3.6]{%
```

\wrapfloat First of all we make the wrapfig package independent from the package load order regarding the float package. Since the usage of \@float@setevery is missing in the code of the wrapfig package (it should be in the redefinition of \float@restyle, right after \@nameuse{fst@#1}), we don't use it here, too, especially since \wrapfloat will usually not be used when used with re-styled floats.

```
  \renewcommand*\wrapfloat[1]{%
    \def\@captype{#1}%
    \@ifundefined{fst@#1}{}{%
      \@nameuse{fst@#1}%
%      \@float@setevery{#1}%
      \def\WF@floatstyhook{\let\@currbox\WF@box
         \global\setbox\WF@box\float@makebox{\wd\WF@box}}}%
    \@ifnextchar[\WF@wr{\WF@wr[]}}
```

\WF@rapt Original code:

```
\def\WF@rapt[#1]#2{% final two args: #1 = overhang,  #2 = width,
  \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
  \global\setbox\WF@box\vtop\bgroup \setlength\hsize{#2}%
  \ifdim\hsize>\z@ \@parboxrestore \else
  \setbox\z@\hbox\bgroup \let\wf@@caption\caption \let\caption\wf@caption
  \ignorespaces \fi}
```

Our code has \WF@captionstyhook in addition:

```
\def\WF@rapt[#1]#2{% final two args: #1 = overhang,  #2 = width,
  \gdef\WF@ovh{#1}% hold overhang for later, when \width is known
  \global\setbox\WF@box\vtop\bgroup \setlength\hsize{#2}%
  \expandafter\WF@captionstyhook\expandafter{\@captype}% <= new
  \ifdim\hsize>\z@ \@parboxrestore \else
  \setbox\z@\hbox\bgroup \let\wf@@caption\caption \let\caption\wf@caption
  \ignorespaces \fi}%
```

\WF@captionstyhook We place our hyperref anchor here, apply the ‘wrap’ options etc. Since the usage of \@float@setevery is missing in the wrapfig package we will catch it up here for making the necessary adaptions to the float package.

```
\def\WF@captionstyhook#1{%
  \let\@captype\@undefined
  \@ifundefined{fst@#1}{}{\@float@setevery{#1}}%
  \caption@settype{#1}%
  \caption@clearmargin
%%%    \caption@setoptions{wrap}%
  \caption@setoptions{wrap#1}}%

}{}
```

References

[1] Till Tantau:
The beamer class, User Guide for version 3.25,
December 26, 2012

[2] Markus Kohm & Jens-Uwe-Morawski:
KOMA-Script – a versatile LaTeX 2ε bundle,
2012-07-22

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

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

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

[6] Olga Lapko:
The floatrow package documentation,
2009/08/02

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

[8] Sebastian Rahtz & Heiko Oberdiek:
Hypertext marks in LaTeX: a manual for hyperref,
November 2012

[9] Heiko Oberdiek:
The hypcap package – Adjusting anchors of captions,
2011/02/16

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

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

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

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

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

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

[16] Steven D. Cochran:
The subfigure package,
2005/03/15

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

[18] Johannes Braams and Theo Jurriens:
The supertabular environment,
2004/02/20

[19] Donald Arseneau:
Three part tables: title, tabular environment, notes,
March 5, 2010

[20] Donald Arseneau:
WRAPFIG.STY ver 3.6,
Jan 31, 2003

[21] Peter Wilson:
The xtab package,
2011/07/31