The Implementation of the caption package*

Axel Sommerfeldt
caption@sommerfee.de

2008/08/24

Abstract

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

The kernel provides all the user commands and internal macros which are necessary for typesetting captions and setting parameters regarding these. While the standard LaTeX document classes provide an internal command called \@makecaption and no options to control its behavior (except the vertical skips above and below the caption itself), we provide similar commands called \caption@make and \caption@@make, but with a lot of options which can be selected with \captionsetup. Loading the kernel part do not change the output of a LaTeX document – it just provides functionality which can be used by LaTeX 2ε packages which typesets captions, for example the caption and subfig packages.

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 v3.1j, last revised 2008/08/24.

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

Contents

1 Kernel

1.1 Identification

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{caption3}[2008/08/24 v3.1j caption3 kernel (AR)]
```

1.2 Generic helpers

`\@nameundef` This is the opposite to `\@namedef` which is offered by the LaTeX kernel. We use it to remove the definition of some commands and keyval options after `\begin{document}` (to save TeX memory) and to remove caption options defined with `\captionsetup[`⟨*type*⟩`]`.

```
\providecommand*\@nameundef[1]{%
  \expandafter\let\csname #1\endcsname\@undefined}
```

`\l@addto@macro` The LaTeX 2ε kernel offers the internal helper macro `\g@addto@macro` which globally adds tokens to existing macros, like in `\AtBeginDocument`. This is the same but it works local, not global (using `\edef` instead of `\xdef`).

```
\providecommand\l@addto@macro[2]{%
  \begingroup
    \toks@\expandafter{#1#2}%
    \edef\@tempa{\endgroup\def\noexpand#1{\the\toks@}}%
  \@tempa}
```

`\bothIfFirst` `\bothIfFirst` tests if the first argument is not empty, `\bothIfSecond` tests if the
`\bothIfSecond` second argument is not empty. If yes both arguments get typeset, otherwise none of them.

```
\def\bothIfFirst#1#2{%
  \protected@edef\caption@tempa{#1}%
  \ifx\caption@tempa\@empty \else
    #1#2%
  \fi}
\def\bothIfSecond#1#2{%
  \protected@edef\caption@tempa{#2}%
  \ifx\caption@tempa\@empty \else
    #1#2%
  \fi}
```

`\caption@ifinlist` This helper macro checks if the first argument is in the comma separated list which is offered as second argument. So for example

```
\caption@ifinlist{frank}{axel,frank,olga,steven}{yes}{no}
```

would expand to `yes`.

```
\newcommand*\caption@ifinlist{%
  \@expandtwoargs\caption@@ifinlist}
\newcommand*\caption@@ifinlist[2]{%
  \begingroup
  \def\@tempa##1,#1,##2\@nil{%
    \endgroup
    \ifx\relax##2\relax
      \expandafter\@secondoftwo
    \else
      \expandafter\@firstoftwo
    \fi}%
  \@tempa,#2,#1,\@nil}%
```

`\caption@ifin@list` `\caption@ifin@list{⟨cmd⟩}{⟨list entry⟩}{⟨yes⟩}{⟨no⟩}`

```
\newcommand*\caption@ifin@list[2]{%
  \caption@ifempty@list#1%
    {\@secondoftwo}%
    {\@expandtwoargs\caption@@ifinlist{#2}{#1}}}
```

`\caption@g@addto@list` `\caption@g@addto@list{⟨cmd⟩}{⟨list entry⟩}`

```
\newcommand*\caption@g@addto@list[2]{%
  \caption@ifempty@list#1{\gdef#1{#2}}{\g@addto@macro#1{,#2}}}
```

`\caption@l@addto@list` `\caption@l@addto@list{⟨cmd⟩}{⟨list entry⟩}`

```
\newcommand*\caption@l@addto@list[2]{%
  \caption@ifempty@list#1{\def#1{#2}}{\l@addto@macro#1{,#2}}}
```

`caption@g@removefrom@list` `\caption@g@removefrom@list{⟨cmd⟩}{⟨list entry⟩}`

```
\newcommand*\caption@g@removefrom@list[2]{%
  \caption@l@removefrom@list#1{#2}%
  \global\let#1#1}
```

`caption@l@removefrom@list` `\caption@l@removefrom@list{⟨cmd⟩}{⟨list entry⟩}`

Caveat: ⟨*cmd*⟩ will be expanded during this process since `\@removeelement` is using `\edef` to build the new list!

```
\newcommand*\caption@l@removefrom@list[2]{%
  \caption@ifempty@list#1{}{\@expandtwoargs\@removeelement{#2}#1#1}}
```

`\caption@for@list` `\caption@for@list{⟨cmd⟩}{⟨code with #1⟩}`

```
\newcommand*\caption@for@list[2]{%
  \caption@ifempty@list#1{}{%
    \def\caption@tempb##1{#2}%
    \@for\caption@tempa:=#1\do{%
      \expandafter\caption@tempb\expandafter{\caption@tempa}}}}
```

`\caption@ifempty@list` `\caption@ifempty@list{⟨cmd⟩}{⟨true⟩}{⟨false⟩}`

```
\newcommand*\caption@ifempty@list[1]{%
  \ifx#1\@undefined
    \expandafter\@firstoftwo
  \else\ifx#1\relax
    \expandafter\expandafter\expandafter\@firstoftwo
  \else\ifx#1\@empty
    \expandafter\expandafter\expandafter\expandafter
      \expandafter\expandafter\expandafter\@firstoftwo
  \else
    \expandafter\expandafter\expandafter\expandafter
      \expandafter\expandafter\expandafter\@secondoftwo
  \fi\fi\fi}
```

`\caption@setbool` `\caption@set@bool` `\caption@ifbool` `\caption@undefbool`

For setting and testing boolean options we offer these three helper macros:

`\caption@setbool{⟨name⟩}{⟨value⟩}`
(with `value = false/true/no/yes/off/on/0/1`)
`\caption@ifbool{⟨name⟩}{⟨if-clause⟩}{⟨else-clause⟩}`
`\caption@undefbool{⟨name⟩}`

```
\newcommand*\caption@setbool[1]{%
  \expandafter\caption@set@bool\csname caption@if#1\endcsname}
\newcommand*\caption@set@bool[2]{%
  \caption@ifinlist{#2}{1,true,yes,on}{%
    \let#1\@firstoftwo
  }{\caption@ifinlist{#2}{0,false,no,off}{%
    \let#1\@secondoftwo
  }{%
    \caption@Error{Undefined boolean value `#2'}%
  }}}
\newcommand*\caption@ifbool[1]{\@nameuse{caption@if#1}}
\newcommand*\caption@undefbool[1]{\@nameundef{caption@if#1}}
```

`\caption@teststar` `\caption@teststar{`⟨*cmd*⟩`}{`⟨*star arg*⟩`}{`⟨*non-star arg*⟩`}`
`\caption@teststar@{`⟨*cmd*⟩`}{`⟨*star arg*⟩`}{`⟨*non-star arg*⟩`}`

```
\newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}
\newcommand*\caption@teststar@[3]{%
  \@ifstar{#1{#2}}{\caption@ifatletter{#1{#2}}{#1{#3}}}}
\AtBeginDocument{\let\caption@teststar@\caption@teststar}
\newcommand*\caption@ifatletter{%
  \ifnum\the\catcode`\@=11
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi}
\AtBeginDocument{\let\caption@ifatletter\@secondoftwo}
```

`\caption@withoptargs` `\caption@withoptargs{`⟨*cmd*⟩`}`

```
\newcommand*\caption@withoptargs[1]{%
  \@ifstar
    {\def\caption@tempa{*}\caption@@withoptargs#1}%
    {\def\caption@tempa{}\caption@@withoptargs#1}}
\def\caption@@withoptargs#1{%
  \@ifnextchar[%]
    {\caption@@@withoptargs#1}%
    {\caption@@@@withoptargs#1}}
\def\caption@@@withoptargs#1[#2]{%
  \l@addto@macro\caption@tempa{[{#2}]}%
  \caption@@withoptargs#1}
\def\caption@@@@withoptargs#1{%
  \expandafter#1\expandafter{\caption@tempa}}
```

`\caption@CheckCommand` `\caption@CheckCommand{`⟨*macro*⟩`}{`⟨*definition of macro*⟩`}`
`\caption@IfCheckCommand` checks if a command already exists, with the same definition. It can be used more-than-once to check if one of multiple definitions will finally match. (It redefines itself later on to `\@gobbletwo` if the two commands match fine, making further checks harmless.)
`\caption@IfCheckCommand{`⟨*true*⟩`}{`⟨*false*⟩`}`
will execute the ⟨*true*⟩ code if one match was finally given, the ⟨*false*⟩ code otherwise. (It simply checks if `\caption@CheckCommand` is `\@gobbletwo` and restores the starting definition of `\caption@CheckCommand`.)

```
\newcommand\caption@DoCheckCommand[2]{%
  \begingroup
    \let\@tempa#1%
    #2%
    \ifx\@tempa#1%
      \endgroup
      \let\caption@CheckCommand\@gobbletwo
    \else
      \endgroup
    \fi}
\@onlypreamble\caption@DoCheckCommand

\let\caption@CheckCommand\caption@DoCheckCommand
\@onlypreamble\caption@CheckCommand

\newcommand*\caption@IfCheckCommand{%
  \ifx\caption@CheckCommand\@gobbletwo
    \let\caption@CheckCommand\caption@DoCheckCommand
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi}
\@onlypreamble\caption@IfCheckCommand
```

\caption@AtBeginDocument `\caption@AtBeginDocument*{`⟨*code*⟩`}`

Same as `\AtBeginDocument` but the execution of code will be surrounded by two `\PackageInfo`s. The starred variant causes the code to be executed after all code specified using the non-starred variant.

```
\let\caption@begindocumenthook\@empty
\let\caption@@begindocumenthook\@empty

\def\caption@AtBeginDocument{%
  \caption@teststar\g@addto@macro
    \caption@@begindocumenthook\caption@begindocumenthook}
\@onlypreamble\caption@AtBeginDocument

\AtBeginDocument{%
   \PackageInfo{caption}{Begin \noexpand\AtBeginDocument code\@gobble}%

   \def\caption@AtBeginDocument{%
     \@ifstar{\g@addto@macro\caption@@begindocumenthook}\@firstofone}%
   \caption@begindocumenthook
   \let\caption@begindocumenthook\@undefined

   \def\caption@AtBeginDocument{%
     \@ifstar\@firstofone\@firstofone}%
   \caption@@begindocumenthook
   \let\caption@@begindocumenthook\@undefined

   \PackageInfo{caption}{End \noexpand\AtBeginDocument code\@gobble}}
```

1.3 Errors

\caption@Warning `\caption@Error{`⟨*message*⟩`}`
\caption@WarningNoLine
\caption@Error
\caption@KV@err

```
\newcommand*\caption@Warning[1]{%
  \caption@WarningNoLine{#1\on@line}}
\newcommand*\caption@WarningNoLine[1]{%
```

```
  \PackageWarning{caption}{#1.^^J\caption@wh\@gobbletwo}}
\newcommand*\caption@Error[1]{%
  \PackageError{caption}{#1}\caption@eh}
\let\caption@KV@err\caption@Error
```

\caption@wh
\caption@eh

At the moment we only offer these two simple warning resp. error helpers.

```
\newcommand*\caption@wh{%
  See the caption package documentation for explanation.}
\newcommand*\caption@eh{%
  If you do not understand this error, please take a closer look\MessageBreak
  at the documentation of the 'caption' package, especially the\MessageBreak
  section about errors.\MessageBreak\@ehc}
```

1.4 Using the keyval package

We need the keyval package for option handling, so we load it here.

```
\RequirePackage{keyval}[1997/11/10]
```

\undefine@key

\undefine@key{⟨*family*⟩}{⟨*key*⟩}
This helper macro is the opposite of \define@key, it removes a keyval definition.

```
\providecommand*\undefine@key[2]{%
  \@nameundef{KV@#1@#2}\@nameundef{KV@#1@#2@default}}
```

\@onlypreamble@key

\onlypreamble@key{⟨*family*⟩}{⟨*key*⟩}
Analogous to \@onlypreamble from LaTeX 2ε.

```
\providecommand*\@preamble@keys{}
\providecommand*\@onlypreamble@key[2]{\@cons\@preamble@keys{{#1}{#2}}}
\@onlypreamble\@onlypreamble@key
\@onlypreamble\@preamble@keys

\providecommand*\@notprerr@key[1]{\KV@err{Can be used only in preamble}}

\caption@AtBeginDocument*{%
  \def\@elt#1#2{\expandafter\let\csname KV@#1@#2\endcsname\@notprerr@key}%
  \@preamble@keys
  \let\@elt\relax}
```

\DeclareCaptionOption

\DeclareCaptionOption{⟨*option*⟩}[⟨*default value*⟩]{⟨*code*⟩}
\DeclareCaptionOption*{⟨*option*⟩}[⟨*default value*⟩]{⟨*code*⟩}
We declare our options using these commands (instead of using \DeclareOption offered by LaTeX 2ε), so the keyval package is used. The starred form makes the option available during the lifetime of the current package only, so they can be used with \usepackage, but *not* with \captionsetup later on.

```
\newcommand*\DeclareCaptionOption{%
  \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
\@onlypreamble\DeclareCaptionOption

\newcommand*\caption@declareoption[2]{%
  #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
\@onlypreamble\caption@declareoption
```

clareCaptionOptionNoValue

\DeclareCaptionOptionNoValue{⟨*option*⟩}{⟨*code*⟩}
\DeclareCaptionOptionNoValue*{⟨*option*⟩}{⟨*code*⟩}
Same as \DeclareCaptionOption but issues an error if a value is given.

```
\newcommand*\DeclareCaptionOptionNoValue{%
  \caption@teststar\caption@declareoption@novalue\AtEndOfPackage\@gobble}
\@onlypreamble\DeclareCaptionOptionNoValue

\newcommand\caption@declareoption@novalue[3]{%
  \caption@declareoption{#1}{#2}[\KV@err]{%
    \caption@option@novalue{#2}{##1}{#3}}}
\@onlypreamble\caption@declareoption@novalue

\newcommand*\caption@option@novalue[2]{%
  \ifx\KV@err#2%
    \expandafter\@firstofone
  \else
    \KV@err{No value allowed for #1}%
    \expandafter\@gobble
  \fi}
```

`\ifcaptionsetup@star` If the starred form of `\captionsetup` is used, this will be set to `true`. (It will be reset to `false` at the end of `\caption@setkeys`.)

```
\newif\ifcaptionsetup@star
```

`\captionsetup` `\captionsetup[`⟨*type*⟩`]{`⟨*keyval-list of options*⟩`}`
`\captionsetup*[`⟨*type*⟩`]{`⟨*keyval-list of options*⟩`}`
If the optional argument 'type' is specified, we simply save or append the option list, otherwise we 'execute' it with `\setkeys`. (The non-starred variant issues a warning if ⟨*keyval-list of options*⟩ is not used later on.)
Note: The starred variant will be used inside packages automatically.

```
\newcommand*\captionsetup{%
  \caption@teststar@\@captionsetup\@gobble\@firstofone}

\newcommand*\@captionsetup[1]{%
  \captionsetup@startrue#1\captionsetup@starfalse
  \@ifnextchar[\caption@setup@options\caption@setup}

\newcommand*\caption@setup{\caption@setkeys{caption}}

\def\caption@setup@options[#1]#2{%
  \@bsphack
    \ifcaptionsetup@star\captionsetup@starfalse\else\caption@addtooptlist{#1}\fi
    \expandafter\caption@l@addto@list\csname caption@opt@#1\endcsname{#2}%
  \@esphack}
```

`\clearcaptionsetup` `\clearcaptionsetup[`⟨*option*⟩`]{`⟨*type*⟩`}`
`\clearcaptionsetup*[`⟨*option*⟩`]{`⟨*type*⟩`}`
This removes the saved option list associated with ⟨*type*⟩. If ⟨*option*⟩ is given, only this option will be removed from the list. (The starred variant does not issue warnings.)
Note: The starred variant will be used inside packages automatically.

```
\newcommand*\clearcaptionsetup{%
  \caption@teststar@\@clearcaptionsetup\@gobble\@firstofone}

\newcommand*\@clearcaptionsetup[1]{%
  \let\caption@tempa#1%
  \@testopt\@@clearcaptionsetup{}}

\def\@@clearcaptionsetup[#1]#2{%
  \@bsphack
```

```
    \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
      {\caption@tempa{\caption@Warning{Option list `#2' undefined}}}%
      {\ifx,#1,%
         \caption@clearsetup{#2}%
       \else
         \caption@@removefromsetup{#1}{#2}%
       \fi}%
  \@esphack}

\newcommand*\caption@clearsetup[1]{%
  \caption@removefromoptlist{#1}%
  \@nameundef{caption@opt@#1}}

\newcommand*\caption@removefromsetup{%
  \let\caption@tempa\@gobble
  \caption@@removefromsetup}

\newcommand*\caption@@removefromsetup[2]{%
  \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
  \expandafter\let\csname caption@opt@#2\endcsname\@undefined
  \def\@tempb##1=##2\@nil{##1}%
  \edef\@tempc{#1}%
  \@for\@tempa:=\@tempa\do{%
    \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%
    \ifx\@tempd\@tempc
      \let\caption@tempa\@gobble
    \else
      \expandafter\expandafter\expandafter\caption@l@addto@list
        \expandafter\csname caption@opt@#2\expandafter\endcsname
        \expandafter{\@tempa}%
    \fi}%
  \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
    {\caption@removefromoptlist{#2}}{}%
  \caption@tempa{\caption@Warning{%
    Option `#1' was not in list `#2'\MessageBreak}}}
```

`\showcaptionsetup` `\showcaptionsetup[`⟨*package*⟩`]{`⟨*type*⟩`}`

This comes for debugging issues: It shows the saved option list which is associated with ⟨*type*⟩.

```
\newcommand*\showcaptionsetup[2][\@firstofone]{%
  \@bsphack
    \GenericWarning{}{%
      #1 Caption Info: Option list on `#2'\MessageBreak
      #1 Caption Data: \@ifundefined{caption@opt@#2}{%
        -none-%
      }{%
        {\expandafter\expandafter\expandafter\strip@prefix
           \expandafter\meaning\csname caption@opt@#2\endcsname}%
      }}%
  \@esphack}

\DeclareCaptionOption{options}{\caption@setoptions{#1}}
```

`\caption@setoptions` `\caption@setoptions{`⟨*type or environment or...*⟩`}`

Caption options which have been saved with `\captionsetup[`⟨*type*⟩`]` can be executed by using this command. It simply executes the saved option list (and clears it afterwards), if there is any.

```
\newcommand*\caption@setoptions[1]{%
  \caption@Debug{options=#1}%
  \expandafter\let\expandafter\caption@opt\csname caption@opt@#1\endcsname
  \ifx\caption@opt\relax \else
    \caption@xsetup\caption@opt
    \caption@clearsetup{#1}%
  \fi}
\newcommand*\caption@xsetup[1]{\expandafter\caption@setup\expandafter{#1}}
```

`\caption@addtooptlist` `\caption@addtooptlist{`⟨*type*⟩`}`
`caption@removefromoptlist` `\caption@removefromoptlist{`⟨*type*⟩`}`
Adds or removes an ⟨*type*⟩ to the list of unused caption options. Note that the catcodes of ⟨*type*⟩ are sanitized here so removing ⟨*type*⟩ from the list do not fail when the float package is used (since `\float@getstyle` gives a result which tokens have catcode 12 = "other").

```
\newcommand*\caption@addtooptlist[1]{%
  \@ifundefined{caption@opt@#1@lineno}{%
    \caption@dooptlist\caption@g@addto@list{#1}%
    \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
  }{}}
\newcommand*\caption@removefromoptlist[1]{%
  \caption@dooptlist\caption@g@removefrom@list{#1}%
  \global\expandafter\let\csname caption@opt@#1@lineno\endcsname\@undefined}
\newcommand*\caption@dooptlist[2]{%
  \begingroup
    \edef\@tempa{#2}\@onelevel@sanitize\@tempa
    \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
  \endgroup}
\AtEndDocument{%
  \caption@for@list\caption@optlist{%
    \caption@WarningNoLine{%
      Unused \string\captionsetup[#1]
      on input line \csname caption@opt@#1@lineno\endcsname}}}
```

`\caption@setkeys` `\caption@setkeys[`⟨*package*⟩`]{`⟨*family*⟩`}{`⟨*key-values*⟩`}`
This one simply calls `\setkeys{`⟨*family*⟩`}{`⟨*key-values*⟩`}` but lets the error messages not refer to the keyval package, but to the ⟨*package*⟩ package instead.

```
\newcommand*\caption@setkeys{\@dblarg\caption@@setkeys}
\long\def\caption@@setkeys[#1]#2#3{%
  \@bsphack
  \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
  \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
  \expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
  \let\KV@errx\KV@err
  \edef\caption@keydepth{\caption@keydepth i}%
  \caption@Debug{\protect\setkeys{#2}{#3}}%
  \setkeys{#2}{#3}%
```

```
  \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
  \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsnam
  \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsn
  \ifx\caption@keydepth\@empty \captionsetup@starfalse \fi
  \@esphack}
\let\caption@keydepth\@empty
```

`\caption@ExecuteOptions` `\caption@ExecuteOptions{`⟨*family*⟩`}{`⟨*key-values*⟩`}`

We execute our options using the keyval interface, so we use this one instead of `\ExecuteOptions` offered by LaTeX 2ε.

```
\newcommand*\caption@ExecuteOptions[2]{%
  \@expandtwoargs\caption@setkeys{#1}{#2}}%
\@onlypreamble\caption@ExecuteOptions
```

`\caption@ProcessOptions` `\caption@ProcessOptions*{`⟨*family*⟩`}`

We process our options using the keyval package, so we use this one instead of `\ProcessOptions` offered by LaTeX 2ε. The starred variant do not process the global options. (This code was taken from the hyperref package[9] *v6.74* and improved.)

```
\newcommand*\caption@ProcessOptions{%
  \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
\@onlypreamble\caption@ProcessOptions

\newcommand*\caption@@ProcessOptions[2]{%
  \let\@tempc\relax
  \let\caption@tempa\@empty
  #1{% \@firstofone -or- \@gobble
    \@for\CurrentOption:=\@classoptionslist\do{%
      \@ifundefined{KV@#2@\CurrentOption}{}{%
        \@ifundefined{KV@#2@\CurrentOption @default}{%
          \PackageInfo{#2}{Global option `\CurrentOption' ignored}%
        }{%
          \PackageInfo{#2}{Global option `\CurrentOption' processed}%
          \edef\caption@tempa{\caption@tempa,\CurrentOption,}%
          \@expandtwoargs\@removeelement\CurrentOption
            \@unusedoptionlist\@unusedoptionlist
        }%
      }%
    }%
    \let\CurrentOption\@empty
  }%
  \caption@ExecuteOptions{#2}{\caption@tempa\@ptionlist{\@currname.\@currext}}%
  \AtEndOfPackage{\let\@unprocessedoptions\relax}}
\@onlypreamble\caption@@ProcessOptions
```

1.5 Margin resp. width

`\captionmargin` `\captionwidth`

`\captionmargin` and `\captionwidth` contain the extra margin resp. the total width used for captions. Please never set these values in a direct way, they are just accessible in user documents to provide compatibility to *v1.x*.

Note that we can only set one value at a time, 'margin' *or* 'width'. If `\captionwidth` is not zero we will take this value afterwards, otherwise `\captionmargin` and `\captionmargin@`.

```
\newdimen\captionmargin
\newdimen\captionmargin@
\newdimen\captionwidth

\DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
\DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
\DeclareCaptionOption{width}{\setcaptionwidth{#1}}
\DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}
\DeclareCaptionOptionNoValue{oneside}{\caption@set@bool\caption@iftwoside0}

\DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
\DeclareCaptionOption{maxmargin}{\caption@setoptcmd\caption@maxmargin{#1}}
```

`\setcaptionmargin` `\setcaptionmargin{`⟨*amount*⟩`}`
`\setcaptionmargin*{`⟨*amount*⟩`}`
Please never use them in user documents, it's just there to provide compatibility to the caption2 package.

```
\newcommand*\setcaptionmargin{%
  \caption@teststar\caption@setmargin\@gobble\@firstofone}

\newcommand*\caption@setmargin[2]{%
  #1{\captionwidth\z@}%
  \caption@@setmargin#2,#2,\@nil}

\def\caption@@setmargin#1,#2,#3\@nil{%
  \setlength\captionmargin@{#2}%
  \setlength\captionmargin{#1}%
  \addtolength\captionmargin@{-\captionmargin}}
```

`\setcaptionwidth` `\setcaptionwidth{`⟨*amount*⟩`}`
Please never use this in user documents, it's just there to provide compatibility to the caption2 package.

```
\newcommand*\setcaptionwidth{%
  \captionmargin\z@
  \captionmargin@\z@
  \setlength\captionwidth}
```

`\caption@counter` This counter numbers the captions. At the moment it will be used inside `\caption@ifoddpage` only.

```
\newcommand*\caption@thecounter{0}

\newcommand*\caption@stepcounter{%
  \@tempcnta\caption@thecounter
  \advance\@tempcnta\@ne
  \xdef\caption@thecounter{\the\@tempcnta}}
```

`\caption@newlabel` This command is a modified version of `\newlabel` from LaTeX2e. It will be written to the `.aux` file to pass label information from one run to another. (We use it inside `\caption@ifoddpage` and `\caption@ragged`.)

```
\newcommand*\caption@newlabel{\@newl@bel{caption@r}}
```

`\caption@thepage` This command is a modified version of `\thepage` from LaTeX2e. It will be used inside `\caption@ifoddpage` only.

```
\newcommand*\caption@thepage{\the\c@page}
```

\caption@label This command is a modified version of \label from LaTeX2e. It will be used inside \caption@ifoddpage and \FP@helpNote.

```
\newcommand*\caption@label[1]{%
  \caption@@label
  \protected@write\@auxout{\let\caption@thepage\relax}%
          {\string\caption@newlabel{#1}{\caption@thepage}}}
\newcommand*\caption@@label{%
  \global\let\caption@@label\relax
  \protected@write\@auxout{}%
    {\string\providecommand*\string\caption@newlabel[2]{}}}
```

\caption@pageref This command is a modified version of \pageref from LaTeX2e. It will be used inside \caption@ifoddpage and \FP@helpNote.

```
\newcommand*\caption@pageref[1]{%
  \expandafter\ifx\csname caption@r@#1\endcsname\relax
    \G@refundefinedtrue % => 'There are undefined references.'
    \caption@Warning{Reference on page \thepage \space undefined}%
  \else
    \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
  \fi}
```

\caption@ifoddpage At the moment this macro uses an own label...ref mechanism, but an alternative implementation method would be using the refcount package[24] and \ifodd\getpagerefnumber{...}. *Note:* This macro re-defines itself so the .aux file will only be used once per group.

```
\newcommand*\caption@ifoddpage{%
  \caption@iftwoside{%
    \caption@label\caption@thecounter
    \caption@pageref\caption@thecounter
    \ifodd\caption@thepage
      \let\caption@ifoddpage\@firstoftwo
    \else
      \let\caption@ifoddpage\@secondoftwo
    \fi
  }{\let\caption@ifoddpage\@firstoftwo}%
  \caption@ifoddpage}
```

\caption@setoptcmd \caption@setoptcmd{⟨*cmd*⟩}{⟨*off -or- value*⟩}

```
\newcommand*\caption@setoptcmd[2]{%
  \caption@ifinlist{#2}{0,false,no,off}{\let#1\@undefined}{\def#1{#2}}}
```

1.6 Indentions

\caption@indent These are the indentions we support.
\caption@parindent
\caption@hangindent

```
\newdimen\caption@indent
\newdimen\caption@parindent
\newdimen\caption@hangindent

\DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
       \setlength\caption@indent{#1}}
\DeclareCaptionOption{indention}[\leftmargini]{%
       \setlength\caption@indent{#1}}
\DeclareCaptionOption{parindent}{%
```

```
        \setlength\caption@parindent{#1}}
\DeclareCaptionOption{hangindent}{%
        \setlength\caption@hangindent{#1}}
\DeclareCaptionOption{parskip}{%
        \l@addto@macro\caption@@par{\setlength\parskip{#1}}}
```

There is an option clash between the KOMA-Script document classes and the caption kernel, both define the options `parindent` and `parskip` but with different meaning. Furthermore the ones defined by the caption kernel take a value as parameter but the KOMA-Script ones do not. So we need special versions of the options `parindent` and `parskip` here which determine if a value is given (and therefore should be treated as our option) or not (and therefore should be ignored by us).[2]

```
\@ifundefined{scr@caption}{}{%
  \let\caption@KV@parindent\KV@caption@parindent
  \DeclareCaptionOption{parindent}[]{%
    \ifx,#1,%
      \caption@Debug{Option 'parindent' ignored}%
    \else
      \caption@KV@parindent{#1}%
    \fi}%
  \let\caption@KV@parskip\KV@caption@parskip
  \DeclareCaptionOption{parskip}[]{%
    \ifx,#1,%
      \caption@Debug{Option 'parskip' ignored}%
    \else
      \caption@KV@parskip{#1}%
    \fi}%
}
```

1.7 Styles

`\DeclareCaptionStyle` `\DeclareCaptionStyle{`⟨*name*⟩`}[`⟨*single-line-list-of-KV*⟩`]{`⟨*list-of-KV*⟩`}`

```
\newcommand*\DeclareCaptionStyle[1]{%
  \@testopt{\caption@declarestyle{#1}}{}}
\@onlypreamble\DeclareCaptionStyle
\def\caption@declarestyle#1[#2]#3{%
  \global\@namedef{caption@sls@#1}{#2}%
  \global\@namedef{caption@sty@#1}{#3}}
\@onlypreamble\caption@declarestyle
\DeclareCaptionOption{style}{\caption@setstyle{#1}}
\DeclareCaptionOption{style*}{\caption@setstyle*{#1}}
\DeclareCaptionOption{singlelinecheck}[1]{\caption@set@bool\caption@ifslc{#1}}
\DeclareCaptionOption{slc}[1]{\KV@caption@singlelinecheck{#1}}
```

`\caption@setstyle` `\caption@setstyle{`⟨*name*⟩`}`
`\caption@setstyle*{`⟨*name*⟩`}`

Selecting a caption style means saving the additional ⟨*single-line-list-of-KV*⟩ (this will be done by `\caption@sls`), resetting the caption options to the base ones (this will be

[2] This problem was completely solved due a change of `\caption@ProcessOptions` in the caption package *v3.0j*, but we still need this workaround since these options would otherwise still collide with the current version 1.3 of the subfig package (Sigh!)

done using `\caption@resetstyle`) and executing the ⟨*list-of-KV*⟩ options (this will be done using `\caption@setup`).
The starred version will give no error message if the given style is not defined.

```
\newcommand*\caption@setstyle{%
  \caption@teststar\caption@@setstyle\@gobble\@firstofone}
\newcommand*\caption@@setstyle[2]{%
  \@ifundefined{caption@sty@#2}%
    {#1{\caption@Error{Undefined style `#2'}}}%
    {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
     \ifx\caption@setstyle@flag\@undefined
       \let\caption@setstyle@flag\relax
       \caption@resetstyle
       \caption@xsetup\caption@sty
       \let\caption@setstyle@flag\@undefined
     \else
       \caption@xsetup\caption@sty
     \fi
     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
       \expandafter{\caption@sls}}}
```

`\caption@resetstyle` This resets (nearly) all caption options to the base ones. *Note that this does not touch the skips and the positioning!*

```
\newcommand*\caption@resetstyle{%
  \caption@setup{%
    format=plain,labelformat=default,labelsep=colon,textformat=simple,%
    justification=justified,font=,size=,labelfont=,textfont=,%
    margin=0pt,minmargin=0,maxmargin=0,%
    indent=0pt,parindent=0pt,hangindent=0pt,%
    slc,rule,strut}%
  \caption@clearsetup{singleline}}
```

Currently there are two pre-defined styles, called 'base' & 'default'. The first one is a perfect match to the behavior of `\@makecaption` offered by the standard LaTeX document classes (and was called 'default' in the caption package *v3.0*), the second one matches the document class actually used.

```
\DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
\DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
  format=default,labelsep=default,textformat=default,%
  justification=default,font=default,labelfont=default,textfont=default}
```

1.8 Formats

`\DeclareCaptionFormat` `\DeclareCaptionFormat{`⟨*name*⟩`}{`⟨*code with #1, #2, and #3*⟩`}`
`\DeclareCaptionFormat*{`⟨*name*⟩`}{`⟨*code with #1, #2, and #3*⟩`}`
The starred form causes the code being typeset in vertical (instead of horizontal) mode, but does not support the `indention=` option.

```
\newcommand*\DeclareCaptionFormat{%
  \caption@teststar\caption@declareformat\@gobble\@firstofone}
\@onlypreamble\DeclareCaptionFormat
```

```
\newcommand*\caption@declareformat[2]{%
  \@dblarg{\caption@@declareformat#1{#2}}}
\@onlypreamble\caption@declareformat

\long\def\caption@@declareformat#1#2[#3]#4{%
  \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
  \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
  \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
\@onlypreamble\caption@@declareformat

\DeclareCaptionOption{format}{\caption@setformat{#1}}
```

`\caption@setformat` `\caption@setformat{`⟨*name*⟩`}`

Selecting a caption format simply means saving the code (in `\caption@fmt`) and if the code should be used in horizontal or vertical mode (`\caption@ifh`).

```
\newcommand*\caption@setformat[1]{%
  \@ifundefined{caption@fmt@#1}%
    {\caption@Error{Undefined format `#1'}}%
    {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
     \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
     \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}
```

`clareCaptionDefaultFormat`

```
\newcommand*\DeclareCaptionDefaultFormat[1]{%
  \expandafter\def\expandafter\caption@fmt@default\expandafter
    {\csname caption@fmt@#1\endcsname}%
  \expandafter\def\expandafter\caption@slfmt@default\expandafter
    {\csname caption@slfmt@#1\endcsname}%
  \expandafter\def\expandafter\caption@ifh@default\expandafter
    {\csname caption@ifh@#1\endcsname}}
\@onlypreamble\DeclareCaptionDefaultFormat
```

There are two pre-defined formats, called ‘plain’ and ‘hang’.

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

\DeclareCaptionFormat{hang}[#1#2#3\par]{%
  \caption@ifin@list\caption@lsepcrlist\caption@lsepname
    {\caption@Error{%
       The option `labelsep=\caption@lsepname' does not work\MessageBreak
       with `format=hang'}}%
    {\@hangfrom{#1#2}%
     \advance\caption@parindent\hangindent
     \advance\caption@hangindent\hangindent
     \caption@@par#3\par}}
```

‘default’ usually maps to ‘plain’.

```
\DeclareCaptionDefaultFormat{plain}
```

1.9 Label formats

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

```
\newcommand*\DeclareCaptionLabelFormat[2]{%
  \global\@namedef{caption@lfmt@#1}##1##2{#2}}
\@onlypreamble\DeclareCaptionLabelFormat
```

```
480 \DeclareCaptionOption{labelformat}{\caption@setlabelformat{#1}}
```

\caption@setlabelformat

\caption@setlabelformat{⟨*name*⟩}

Selecting a caption label format simply means saving the code (in \caption@lfmt).

```
481 \newcommand*\caption@setlabelformat[1]{%
482   \@ifundefined{caption@lfmt@#1}%
483     {\caption@Error{Undefined label format `#1'}}%
484     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}
```

There are four pre-defined label formats, called 'empty', 'simple', 'parens', and 'brace'.

```
485 \DeclareCaptionLabelFormat{empty}{}
486 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
487 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace}(#2)}
488 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace}#2)}
```

'default' usually maps to 'simple'.

```
489 \def\caption@lfmt@default{\caption@lfmt@simple}
```

1.10 Label separators

lareCaptionLabelSeparator

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

The starred form causes the label separator to be typeset *without* using \captionlabelfont.

```
490 \newcommand\DeclareCaptionLabelSeparator{%
491   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
492 \@onlypreamble\DeclareCaptionLabelSeparator

493 \newcommand\caption@declarelabelseparator[3]{%
494   \global\expandafter\let\csname caption@iflf@#2\endcsname#1%
495   \global\long\@namedef{caption@lsep@#2}{#3}%
496   \caption@@declarelabelseparator{#2}#3\\\@nil}
497 \@onlypreamble\caption@declarelabelseparator

498 \long\def\caption@@declarelabelseparator#1#2\\#3\@nil{%
499   \def\@tempa{#3}\ifx\@tempa\@empty \else
500     \caption@g@addto@list\caption@lsepcrlist{#1}%
501   \fi}
502 \@onlypreamble\caption@@declarelabelseparator

503 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
504 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

caption@setlabelseparator

\caption@setlabelseparator{⟨*name*⟩}

Selecting a caption label separator simply means saving the code (in \caption@lsep).

```
505 \newcommand*\caption@setlabelseparator[1]{%
506   \@ifundefined{caption@lsep@#1}%
507     {\caption@Error{Undefined label separator `#1'}}%
508     {\edef\caption@lsepname{#1}%
509      \expandafter\let\expandafter\caption@iflf\csname caption@iflf@#1\endcsname
510      \expandafter\let\expandafter\caption@lsep\csname caption@lsep@#1\endcsname}}
```

There are seven pre-defined label separators, called 'none', 'colon', 'period', 'space', 'quad', 'newline', and 'endash'.

```
511 \DeclareCaptionLabelSeparator{none}{}
```

```
\DeclareCaptionLabelSeparator{colon}{: }
\DeclareCaptionLabelSeparator{period}{. }
\DeclareCaptionLabelSeparator{space}{ }
\DeclareCaptionLabelSeparator*{quad}{\quad}
\DeclareCaptionLabelSeparator*{newline}{\\}
\DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}
```

'default' usually maps to 'colon'.

```
\def\caption@lsep@default{\caption@lsep@colon}
\def\caption@iflf@default{\caption@iflf@colon}
```

1.11 Text formats

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

```
\newcommand*\DeclareCaptionTextFormat[2]{%
  \global\long\@namedef{caption@tfmt@#1}##1{#2}}
\@onlypreamble\DeclareCaptionTextFormat

\DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
\DeclareCaptionOption{strut}[1]{\caption@set@bool\caption@ifstrut{#1}}
```

\caption@settextformat `\caption@settextformat{`⟨*name*⟩`}`

Selecting a caption text format simply means saving the code (in `\caption@tfmt`).

```
\newcommand*\caption@settextformat[1]{%
  \@ifundefined{caption@tfmt@#1}%
    {\caption@Error{Undefined text format '#1'}}%
    {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}
```

There are two pre-defined text formats, called 'simple' and 'period'.

```
\DeclareCaptionTextFormat{simple}{#1}
\DeclareCaptionTextFormat{period}{#1.}
```

'default' usually maps to 'simple'.

```
\def\caption@tfmt@default{\caption@tfmt@simple}
```

1.12 Fonts

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

```
\newcommand*\DeclareCaptionFont[2]{%
  \define@key{caption@fnt}{#1}[]{\l@addto@macro\caption@fnt{#2}}}
\@onlypreamble\DeclareCaptionFont
```

\DeclareCaptionDefaultFont `\DeclareCaptionDefaultFont{`⟨*name*⟩`}{`⟨*code*⟩`}`

```
\newcommand*\DeclareCaptionDefaultFont[2]{%
  \global\@namedef{caption#1@default}{#2}}
\@onlypreamble\DeclareCaptionDefaultFont

\DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
\DeclareCaptionOption{font+}{\caption@addtofont{font}{#1}}
\DeclareCaptionDefaultFont{font}{}

\DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
\DeclareCaptionOption{labelfont+}{\caption@addtofont{labelfont}{#1}}
\DeclareCaptionDefaultFont{labelfont}{}
```

```
\DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
\DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
\DeclareCaptionDefaultFont{textfont}{}
```

`\caption@setfont` `\caption@setfont{`⟨*name*⟩`}{`⟨*keyval-list of names*⟩`}`

Selecting a caption font means saving all the code snippets in `\caption`⟨*name*⟩.

```
\newcommand*\caption@setfont[1]{%
  \expandafter\let\csname caption#1\endcsname\@empty
  \caption@addtofont{#1}}
```

`\caption@addtofont` `\caption@addtofont{`⟨*name*⟩`}{`⟨*keyval-list of names*⟩`}`

Like `\caption@setfont`, but adds the code snippets to `\caption`⟨*name*⟩.
Because we use `\setkeys` recursive here we need to do this inside an extra group.

```
\newcommand*\caption@addtofont[2]{%
  \begingroup
    \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
    \define@key{caption@fnt}{default}[]{%
      \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
    \caption@setkeys[caption]{caption@fnt}{#2}%
    \global\let\caption@tempa\caption@fnt
  \endgroup
  \expandafter\let\csname caption#1\endcsname\caption@tempa}
```

`\caption@font` `\caption@font{`⟨*keyval-list of names*⟩`}`
`\caption@font*{`⟨*keyval-code*⟩`}`

Sets the given font, e.g. `\caption@font{small,it}` is equivalent to `\small\itshape`.

```
\newcommand*\caption@font{%
  \caption@teststar\caption@@font\@firstofone
           {\caption@setkeys[caption]{caption@fnt}}}
\newcommand*\caption@@font[2]{%
  \begingroup
  \def\caption@fnt{\endgroup}%
  #1{#2}%
  \caption@fnt}
```

These are the pre-defined font code snippets.

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

```
\DeclareCaptionFont{normalfont}{\normalfont}
\DeclareCaptionFont{up}{\upshape}
\DeclareCaptionFont{it}{\itshape}
\DeclareCaptionFont{sl}{\slshape}
\DeclareCaptionFont{sc}{\scshape}
\DeclareCaptionFont{md}{\mdseries}
\DeclareCaptionFont{bf}{\bfseries}
\DeclareCaptionFont{rm}{\rmfamily}
\DeclareCaptionFont{sf}{\sffamily}
\DeclareCaptionFont{tt}{\ttfamily}
```

```
\DeclareCaptionFont{scriptsize}{\scriptsize}
\DeclareCaptionFont{footnotesize}{\footnotesize}
\DeclareCaptionFont{small}{\small}
\DeclareCaptionFont{normalsize}{\normalsize}
```

```
\DeclareCaptionFont{large}{\large}
\DeclareCaptionFont{Large}{\Large}

\DeclareCaptionFont{singlespacing}{\setstretch\setspace@singlespace}% normally 1
\DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
\DeclareCaptionFont{doublespacing}{\doublespacing}
\DeclareCaptionFont{stretch}{\setstretch{#1}}
\caption@AtBeginDocument{\providecommand*\setstretch[1]{}}

%\DeclareCaptionFont{normal}{%
%  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
\DeclareCaptionFont{normal}{%
  \caption@font*{%
    \KV@caption@fnt@normalcolor\@unused
    \KV@caption@fnt@normalfont\@unused
    \KV@caption@fnt@normalsize\@unused
    \KV@caption@fnt@singlespacing\@unused}}
```

The old versions *v1.x* of the caption package offered this command to setup the font size used for captions. We still do so old documents will work fine.

```
\DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
\DeclareCaptionDefaultFont{size}{}
```

1.13 Justifications

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

```
\newcommand*\DeclareCaptionJustification[2]{%
  \global\@namedef{caption@hj@#1}{#2}% for compatibility to v3.0
  \DeclareCaptionFont{#1}{#2}}
\@onlypreamble\DeclareCaptionJustification
```

ptionDefaultJustification `\DeclareCaptionDefaultJustification{`⟨*code*⟩`}`

```
\newcommand*\DeclareCaptionDefaultJustification[1]{%
  \global\@namedef{caption@hj@default}{#1}% for compatibility to v3.0
  \DeclareCaptionDefaultFont{@hj}{#1}}
\@onlypreamble\DeclareCaptionDefaultJustification

\DeclareCaptionOption{justification}{\caption@setjustification{#1}}
\DeclareCaptionDefaultJustification{}
```

\caption@setjustification `\caption@setjustification{`⟨*name*⟩`}`

Selecting a caption justification simply means saving the code (in `\caption@hj`).

```
\newcommand*\caption@setjustification{\caption@setfont{@hj}}
```

These are the pre-defined justification code snippets.

```
\DeclareCaptionJustification{justified}{}
\DeclareCaptionJustification{centering}{\centering}
\DeclareCaptionJustification{centerfirst}{\centerfirst}
\DeclareCaptionJustification{centerlast}{\centerlast}
\DeclareCaptionJustification{raggedleft}{\raggedleft}
\DeclareCaptionJustification{raggedright}{\raggedright}
```

\centerfirst Please blame Frank Mittelbach for the code of `\centerfirst :-)`

```
\providecommand\centerfirst{%
```

```
  \let\\\@centercr
  \edef\caption@normaladjust{%
    \leftskip\the\leftskip
    \rightskip\the\rightskip
    \parfillskip\the\parfillskip\relax}%
  \leftskip\z@\@plus -1fil%
  \rightskip\z@\@plus 1fil%
  \parfillskip\z@skip
  \noindent\hskip\z@\@plus 2fil%
  \@setpar{\@@par\@restorepar\caption@normaladjust}}
```

`\centerlast` This is based on code from Anne Brüggemann-Klein[23]

```
\providecommand\centerlast{%
  \let\\\@centercr
  \leftskip\z@\@plus 1fil%
  \rightskip\z@\@plus -1fil%
  \parfillskip\z@\@plus 2fil\relax}
```

1.13.1 The ragged2e package

We also support the upper-case commands offered by the ragged2e package. Note that these just map to their lower-case variants if the ragged2e package is not available.

```
\DeclareCaptionJustification{Centering}{%
  \caption@ragged\Centering\centering}
\DeclareCaptionJustification{RaggedLeft}{%
  \caption@ragged\RaggedLeft\raggedleft}
\DeclareCaptionJustification{RaggedRight}{%
  \caption@ragged\RaggedRight\raggedright}
```

`\caption@ragged` `\caption@ragged` will be basically defined as

```
\AtBeginDocument{\IfFileExists{ragged2e.sty}%
  {\RequirePackage{ragged2e}\let\caption@ragged\@firstoftwo}%
  {\let\caption@ragged\@secondoftwo}}
```

but with an additional warning if the ragged2e package is not loaded (yet). (This warning will be type out only one time per option, that's why we need the `caption\string#1` stuff.) Furthermore we load the ragged2e package, if needed and available.

```
\newcommand*\caption@ragged{%
  \caption@Debug{We need ragged2e}%
  \protected@write\@auxout{}{\string\caption@newlabel{ragged2e}{}}%
  \global\let\caption@ragged\caption@@ragged
  \caption@ragged}
\caption@AtBeginDocument{%
  \@ifundefined{caption@r@ragged2e}{%
    \newcommand*\caption@@ragged{%
      \caption@Warning{%
        'ragged2e' support has been changed.\MessageBreak
        Rerun to get captions right}%
      \global\let\caption@ragged\@secondoftwo % suppress further warnings
      \caption@ragged}%
  }{%
    \caption@Debug{We load ragged2e}%
```

```
    \IfFileExists{ragged2e.sty}{%
      \RequirePackage{ragged2e}%
      \let\caption@@ragged\@firstoftwo
    }{%
      \newcommand*\caption@@ragged[2]{%
        \@ifundefined{caption\string#1}{%
          \caption@Warning{%
            'ragged2e' package not loaded, therefore\MessageBreak
            substituting \string#2 for \string#1\MessageBreak}%
          \global\@namedef{caption\string#1}}{}%
        #2}%
    }%
  }}
```

1.14 Vertical spaces before and after captions

\abovecaptionskip
\belowcaptionskip

Usually these skips are defined within the document class, but some document classes don't do so.

```
\@ifundefined{abovecaptionskip}{%
  \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}}{}
\@ifundefined{belowcaptionskip}{%
  \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}}{}

\DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
\DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
\DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}
```

\caption@rule

\caption@rule
Draws an invisible rule to adjust the "skip" setting.

```
\newcommand*\caption@rule{\caption@ifrule\caption@hrule\relax}
\newcommand*\caption@hrule{\hrule\@height\z@}

\DeclareCaptionOption{rule}[1]{\caption@set@bool\caption@ifrule{#1}}
```

1.15 Positioning

These macros handle the right position of the caption. Note that the position is actually *not* controlled by the caption3 kernel options, but by the user (or a specific package like the float package) instead. The user can put the \caption command wherever he likes! So this stuff is only to give us a *hint* where to put the right skips, the user usually has to take care for himself that this hint actually matches the right position.

```
\DeclareCaptionOption{position}{\caption@setposition{#1}}
```

\caption@setposition

\caption@setposition{⟨*position*⟩}
Selecting the caption position means that we put \caption@position to the right value. *Please do* ***not*** *use the internal macro* \caption@position *in your own package or document, but use the wrapper macro* \caption@iftop *instead.*

```
\newcommand*\caption@setposition[1]{%
  \caption@ifinlist{#1}{d,default}{%
    \let\caption@position\caption@defaultpos
  }{\caption@ifinlist{#1}{t,top,above}{%
```

```
    \let\caption@position\@firstoftwo
  }{\caption@ifinlist{#1}{b,bottom,below}{%
    \let\caption@position\@secondoftwo
  }{\caption@ifinlist{#1}{a,auto}{%
    \let\caption@position\@undefined
  }{%
    \caption@Error{Undefined position `#1'}%
  }}}}}
```

`\caption@defaultpos` The default 'position' is 'auto', this means that the caption package will try to guess the current position of the caption. (But in many cases, for example in `longtables`, this is doomed to fail!)
The setting 'bottom' correspondents to the `\@makecaption` implementation in the standard LaTeX document classes, but 'auto' should give better results in most cases.

```
%\caption@setdefaultpos{a}% default = auto
\let\caption@defaultpos\@undefined
```

`\caption@iftop` `\caption@iftop{`⟨*true-code*⟩`}{`⟨*false-code*⟩`}`
(If the `position=` is set to `auto` we assume a `bottom` position here.)

```
\newcommand*\caption@iftop{%
  \ifx\caption@position\@undefined
    \let\caption@position\@secondoftwo
%   = \caption@setposition b%
  \fi
  \caption@position}
```

`\caption@fixposition` `\caption@fixposition`
This macro checks if the 'position' is set to 'auto'. If yes, `\caption@autoposition` will be called to set `\caption@position` to a proper value we can actually use.

```
\newcommand*\caption@fixposition{%
  \ifx\caption@position\@undefined
    \caption@autoposition
  \fi}
```

`\caption@autoposition` `\caption@autoposition`
We guess the current position of the caption by checking `\prevdepth`.
A different solution would be setting the `\spacefactor` to something not much less than 1000 (for example 994) in `\caption@start` and checking this value here by `\ifnum\spacefactor=994`. (It's implemented in the threeparttable package[20] this way.)
Another idea would be checking `\@ifminipage`, but since some packages typeset the caption within a simple `\vbox` this does not seem to be a good one.

```
\newcommand*\caption@autoposition{%
  \ifvmode
    \edef\caption@tempa{\the\prevdepth}%
    \caption@Debug{\protect\prevdepth=\caption@tempa}%
    \ifdim\prevdepth>-\p@
      \let\caption@position\@secondoftwo
    \else
      \let\caption@position\@firstoftwo
    \fi
%   = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
```

```
  \else
    \caption@Debug{no \protect\prevdepth}%
    \let\caption@position\@secondoftwo
%   = \caption@setposition b%
  \fi}
```

`\caption@setautoposition` `\caption@setautoposition{`⟨*position*⟩`}`
replaces the above algorithm by a different one (or a fixed position setting).

```
\newcommand*\caption@setautoposition[1]{%
  \def\caption@autoposition{\caption@setposition{#1}}}
```

1.16 Hooks

`\AtBeginCaption` `\AtBeginCaption {`⟨*code*⟩`}`
`\AtEndCaption` `\AtEndCaption {`⟨*code*⟩`}`
These hooks can be used analogous to `\AtBeginDocument` and `\AtEndDocument`.

```
\newcommand*\caption@beginhook{}
\newcommand*\caption@endhook{}
\newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
\newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}
```

1.17 Lists

```
\DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
\DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}
```

`\caption@setlist` `\caption@setlist{`⟨*boolean*⟩`}`

```
\newcommand*\caption@setlist{\caption@set@bool\caption@iflist}
```

`\DeclareCaptionListFormat` `\DeclareCaptionListFormat{`⟨*name*⟩`}{`⟨*code with #1 and #2*⟩`}`

```
\newcommand*\DeclareCaptionListFormat[2]{%
  \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
\@onlypreamble\DeclareCaptionListFormat

\DeclareCaptionOption{listformat}{\caption@setlistformat{#1}}
```

`\caption@setlistformat` `\caption@setlistformat{`⟨*name*⟩`}`
Selecting a caption list format simply means saving the code (in `\caption@lstfmt`).

```
\newcommand*\caption@setlistformat[1]{%
  \@ifundefined{caption@lstfmt@#1}%
    {\caption@Error{Undefined list format `#1'}}%
    {\expandafter\let\expandafter\caption@lstfmt
       \csname caption@lstfmt@#1\endcsname}}
```

There are five pre-defined list formats, taken from the subfig package.

```
\DeclareCaptionListFormat{empty}{}
\DeclareCaptionListFormat{simple}{#1#2}
\DeclareCaptionListFormat{parens}{#1(#2)}
\DeclareCaptionListFormat{subsimple}{#2}
\DeclareCaptionListFormat{subparens}{(#2)}

\def\caption@lstfmt@default{\caption@lstfmt@subsimple}
```

1.18 Debug option

```
\DeclareCaptionOption{debug}[1]{%
  \caption@set@bool\caption@ifdebug{#1}%
  \caption@ifdebug
    {\def\caption@Debug{\PackageInfo{caption}}}%
    {\let\caption@Debug\@gobble}}

\DeclareOption{debug}{\setkeys{caption}{debug}}

\setkeys{caption}{debug=0}
```

1.19 Document classes & Babel support

1.19.1 The standard LaTeX classes

```
\caption@CheckCommand\@makecaption{%
  % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
  \long\def\@makecaption#1#2{%
    \vskip\abovecaptionskip
    \sbox\@tempboxa{#1: #2}%
    \ifdim \wd\@tempboxa >\hsize
      #1: #2\par
    \else
      \global \@minipagefalse
      \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
    \fi
    \vskip\belowcaptionskip}}
```

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

```
\@ifundefined{@captionheadfont}{}{%
  \caption@CheckCommand\@makecaption{%
    % amsart|amsproc|amsbook [2004/08/06 v2.20]
    \long\def\@makecaption#1#2{%
      \setbox\@tempboxa\vbox{\color@setgroup
        \advance\hsize-2\captionindent\noindent
        \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
            {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
        \unskip\kern-2\captionindent\par
        \global\setbox\@ne\lastbox\color@endgroup}%
      \ifhbox\@ne % the normal case
        \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
      \fi
      \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
        \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
      \else % tempboxa contained more than one line
        \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
            \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
      \fi
      \ifnum\@tempcnta<64 % if the float IS a figure...
        \addvspace\abovecaptionskip
        \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
      \else % if the float IS NOT a figure...
        \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
        \nobreak
        \vskip\belowcaptionskip
```

```
      \fi
    \relax
    }}
  \caption@CheckCommand\@makecaption{%
    % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publies par
    \long\def\@makecaption#1#2{%
      \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
      \setbox\@tempboxa\vbox{\color@setgroup
        \advance\hsize-2\captionindent\noindent
        \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
            {\@cdr#2\@nil}{\@addpunct{.}\@captionfont\upshape\enspace#2}%
        \unskip\kern-2\captionindent\par
        \global\setbox\@ne\lastbox\color@endgroup}%
      \ifhbox\@ne % the normal case
        \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
      \fi
      \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
        \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
        \@tempdima\wd\@ne\advance\@tempdima-\captionindent
        \wd\@ne\@tempdima
      \else % tempboxa contained more than one line
        \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
            \unvbox\@tempboxa\parskip\z@skip
            \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
      \fi
      \ifnum\@tempcnta<64 % if the float IS a figure...
        \addvspace\abovecaptionskip
        \noindent\kern\captionindent\box\@ne
      \else % if the float IS NOT a figure...
        \noindent\kern\captionindent\box\@ne
        \nobreak
        \vskip\belowcaptionskip
      \fi
    \relax
    }}
  \let\captionmargin\captionindent % set to 3pc by AMS class
  \begingroup\edef\@tempa{\endgroup
    \noexpand\caption@g@addto@list\noexpand\caption@sty@default
      {margin=\the\captionmargin
       \@ifundefined{smf@makecaption}{}{,maxmargin=.1\linewidth}}}
  \@tempa
  \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
  \DeclareCaptionLabelSeparator{default}{.\enspace}
  \DeclareCaptionDefaultFont{font}{\@captionfont}
  \DeclareCaptionDefaultFont{labelfont}{\@captionheadfont}
  \DeclareCaptionDefaultFont{textfont}{\@captionfont\upshape}
  \captionsetup[figure]{position=b}
  \captionsetup[table]{position=t}
}
```

1.19.3 The beamer class

```
\@ifclassloaded{beamer}{%
```

```
  \caption@CheckCommand\beamer@makecaption{%
    % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
    \long\def\beamer@makecaption#1#2{%
      \def\insertcaptionname{\csname#1name\endcsname}%
      \def\insertcaptionnumber{\csname the#1\endcsname}%
      \def\insertcaption{#2}%
      \nobreak\vskip\abovecaptionskip\nobreak
      \sbox\@tempboxa{\usebeamertemplate**{caption}}%
      \ifdim \wd\@tempboxa >\hsize
        \usebeamertemplate**{caption}\par
      \else
        \global \@minipagefalse
        \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
      \fi
      \nobreak\vskip\belowcaptionskip\nobreak}}
  \DeclareCaptionLabelFormat{default}{#1}
  \DeclareCaptionDefaultJustification{\raggedright}
  \DeclareCaptionDefaultFont{font}{%
    \usebeamerfont*{caption}%
    \usebeamercolor[fg]{caption}}
  \DeclareCaptionDefaultFont{labelfont}{%
    \usebeamercolor[fg]{caption name}%
    \usebeamerfont*{caption name}}
```

If the beamer document class is used, we offer a beamer template called ‘caption3’ which can be used with option ‘beamer’ or `\setbeamertemplate{caption}[caption3]`. (Note that this is of no use when the caption package is used, too.)

```
  \defbeamertemplate{caption}{caption3}{%
    \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
  \DeclareOption{beamer}{%
    % \usebeamertemplate**{caption} will set font
    \DeclareCaptionDefaultFont{font}{}%
    \setbeamertemplate{caption}[caption3]}
}{}
```

1.19.4 The KOMA-Script classes

```
\@ifundefined{scr@caption}{}{%
  \caption@CheckCommand\@makecaption{%
    % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
    \long\def\@makecaption#1#2{%
      \if@captionabove
        \vskip\belowcaptionskip
      \else
        \vskip\abovecaptionskip
      \fi
      \@@makecaption\@firstofone{#1}{#2}%
      \if@captionabove
        \vskip\abovecaptionskip
      \else
        \vskip\belowcaptionskip
      \fi}}
  \DeclareCaptionFormat{default}[#1#2#3\par]{%
```

```
    \ifdofullc@p
      \caption@ifin@list\caption@lsepcrlist\caption@lsepname
        {\caption@Error{%
           The option 'labelsep=\caption@lsepname' does not work\MessageBreak
           with \noexpand\setcaphanging (which is set by default)}}%
        {\caption@fmt@hang{#1}{#2}{#3}}%
    \else
      #1#2%
      \ifdim\cap@indent<\z@
        \par
        \noindent\hspace*{-\cap@indent}%
      \else\if@capbreak
        \par
      \fi\fi
      #3\par
    \fi}
  \DeclareCaptionLabelSeparator{default}{\captionformat}
  \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
  \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
}
```

1.19.5 The NTG Dutch classes

```
\@ifundefined{CaptionFonts}{}{%
  \caption@CheckCommand\@makecaption{%
    % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
    \long\def\@makecaption#1#2{%
      \vskip\abovecaptionskip
      \sbox\@tempboxa{{\CaptionLabelFont#1:} \CaptionTextFont#2}%
      \ifdim \wd\@tempboxa >\hsize
        {\CaptionLabelFont#1:} \CaptionTextFont#2\par
      \else
        \global \@minipagefalse
        \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
      \fi
      \vskip\belowcaptionskip}}
  \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
  \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
}
```

1.19.6 The thesis class

```
\@ifclassloaded{thesis}{%
  \caption@CheckCommand\@makecaption{%
    % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
    \long\def\@makecaption#1#2{%
     \vskip\abovecaptionskip
     \setbox\@tempboxa\hbox{{\cph@font #1:} {\cpb@font #2}}%
     \ifdim \wd\@tempboxa >\hsize
        \@hangfrom{\cph@font #1: }{\cpb@font #2\par}%
     \else
        \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
     \fi
     \vskip\belowcaptionskip}}
```

```
  \DeclareCaptionDefaultFormat{hang}
  \DeclareCaptionDefaultFont{labelfont}{\cph@font}
  \DeclareCaptionDefaultFont{textfont}{\cpb@font}
}{}
```

1.19.7 The frenchb Babel option

```
\@ifundefined{FB@makecaption}{}{%
  \caption@CheckCommand\@makecaption{%
    % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
    % frenchb.ldf [2007/10/05 v2.0e French support from the babel system]
    \long\def\@makecaption#1#2{%
      \vskip\abovecaptionskip
      \sbox\@tempboxa{#1\CaptionSeparator #2}%
      \ifdim \wd\@tempboxa >\hsize
        #1\CaptionSeparator #2\par
      \else
        \global \@minipagefalse
        \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
      \fi
      \vskip\belowcaptionskip}}
  \ifx\@makecaption\STD@makecaption
    \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
    \def\caption@frenchb{% supress frenchb warning
      \let\STD@makecaption\@makecaption
      \let\FB@makecaption\@makecaption}
  \else
    \ifx\@makecaption\@undefined\else
      \PackageInfo{caption}{%
        The definition of \protect\@makecaption\space
        has been changed,\MessageBreak
        frenchb will NOT customize it}%
    \fi
  \fi
}
```

1.19.8 The frenchle/pro package

```
\@ifundefined{frenchTeXmods}{}{%
  \caption@CheckCommand\@makecaption{%
    % french(le).sty [2006/10/03 The french(le) package /V5,9991/]
    % french(le).sty [2007/06/28 The french(le) package /V5,9994/]
    \def\@makecaption#1#2{%
      \ifFTY%
        \def\@secondofmany##1##2\void{##2}%
        \def\@tempa{\@secondofmany#2\void}%
        \ifx\@tempa\empty%
          \let\captionseparator\empty%
        \fi%
        \@mcORI{#1}{\relax\captionfont{#2}}%
      \else
        \@mcORI{#1}{#2}%
      \fi}}
```

```
\caption@CheckCommand\@makecaption{%
  % french(le).sty [2007/02/11 The french(le) package /V5,9993/]
  \def\@makecaption#1#2{%
    \ifFTY%
      \def\@secondofmany##1##2\void{##2}%
      \protected@edef\@tempa{\@secondofmany#2\void}%
      \ifx\@tempa\empty%
        \let\captionseparator\empty%
      \fi%
      \@mcORI{#1}{\relax\captionfont{#2}}%
    \else
      \@mcORI{#1}{#2}%
    \fi}}
\DeclareCaptionDefaultFont{textfont}{\itshape}%
\DeclareCaptionLabelSeparator{default}{\captionseparator\space}%
}
```

1.20 Execution of options

```
\captionsetup{style=default,position=default,%
              list,listformat=default,twoside=\if@twoside 1\else 0\fi}
\ProcessOptions*
\caption@IfCheckCommand{%
  \caption@setbool{documentclass}{1}%
}{%
  \caption@setbool{documentclass}{0}%
  \PackageInfo{caption}{%
         Unknown document class (or package),\MessageBreak
         standard defaults will be used}%
  \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
}
```

1.21 Making an 'List of' entry

`\caption@addcontentsline` `\caption@addcontentsline{`⟨*type*⟩`}{`⟨*list entry*⟩`}`

Makes an entry in the list-of-whatever, if requested, i.e. the argument ⟨*list entry*⟩ is not empty and `listof=` was set to `true`.

```
\newcommand*\caption@addcontentsline[2]{%
  \caption@iflist
    {\def\@tempa{#2}}%
    {\let\@tempa\@empty}%
  \ifx\@tempa\@empty \else
    {\let\\\space
     \addcontentsline{\csname ext@#1\endcsname}{#1}%
                     {\protect\numberline
                       {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
                       {\ignorespaces #2}}}%
  \fi}
```

1.22 Typesetting the caption

`\ifcaption@star` If the starred form of `\caption` is used, this will be set to `true`. (It will be reset to `false` at the end of `\caption@@make`.)

```
\newif\ifcaption@star
```

`\caption@fnum` `\caption@fnum{`⟨*float type*⟩`}`
Typesets the caption label; as replacement for `\fnum@`⟨*float type*⟩.

```
\newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}
```

`\caption@make` `\caption@make{`⟨*float name*⟩`}{`⟨*ref. number*⟩`}{`⟨*text*⟩`}`
Typesets the caption.

```
\newcommand\caption@make[2]{\caption@@make{\caption@lfmt{#1}{#2}}}
```

`\caption@@make` `\caption@@make{`⟨*caption label*⟩`}{`⟨*caption text*⟩`}`

```
\newcommand\caption@@make[2]{%
  \begingroup
  \caption@stepcounter
  \caption@beginhook
```

Check margin, if `\caption@minmargin` or `\caption@maxmargin` is set

```
  \ifx\caption@maxmargin\@undefined \else
    \ifdim\captionmargin>\caption@maxmargin\relax
      \captionmargin\caption@maxmargin\relax
    \fi
  \fi
  \ifx\caption@minmargin\@undefined \else
    \ifdim\captionmargin<\caption@minmargin\relax
      \captionmargin\caption@minmargin\relax
    \fi
  \fi
```

Special single-line treatment (option `singlelinecheck=`)

```
  \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%
```

Typeset the left margin (option `margin=`)

```
  \caption@calcmargin
  \@tempdima\captionmargin
  \ifdim\captionmargin@=\z@ \else
    \caption@ifoddpage{}{\advance\@tempdima\captionmargin@}%
  \fi
  \caption@ifh{\advance\@tempdima\caption@indent}%
  \hspace\@tempdima
```

We actually use a `\vbox` of width `\captionwidth - \caption@indent` to typeset the caption.
Note: `\captionindent` is *not* supported if the caption format was defined with `\DeclareCaptionFormat*`.

```
  \@tempdima\captionwidth
  \caption@ifh{\advance\@tempdima-\caption@indent}%
  \caption@parbox\@tempdima{%
```

Typeset the indention (option `indention=`)
Bugfix 04-05-05: `\hskip-\caption@indent` replaced by `\ifdim\caption@indent=\z@...`

```
    \caption@ifh{%
      \ifdim\caption@indent=\z@
        \leavevmode
      \else
        \hskip-\caption@indent
      \fi}%
```

Typeset the caption itself and close the `\caption@parbox`

```
    \caption@@@make{#1}{#2}%
```

Typeset the right margin (option `margin=`)

```
  \@tempdima\captionmargin
  \ifdim\captionmargin@=\z@ \else
    \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
  \fi
  \hspace\@tempdima

  \caption@endhook
  \endgroup

  \global\caption@starfalse}
```

`\caption@calcmargin` `\caption@calcmargin`

Calculate `\captionmargin` & `\captionwidth`, so both contain valid values.

```
\newcommand*\caption@calcmargin{%
  \ifdim\captionwidth=\z@
    \captionwidth\linewidth
    \advance\captionwidth by -2\captionmargin
    \advance\captionwidth by -\captionmargin@
  \else
    \captionmargin\linewidth
    \advance\captionmargin by -\captionwidth
    \divide\captionmargin by 2
    \captionmargin@\z@
  \fi

  \caption@Debug{%
    \string\hsize=\the\hsize,
    \string\linewidth=\the\linewidth,\MessageBreak
    \string\leftmargin=\the\leftmargin,
    \string\rightmargin=\the\rightmargin,\MessageBreak
    \string\margin=\the\captionmargin,
    \string\margin@=\the\captionmargin@,
    \string\width=\the\captionwidth}%

}
```

`\caption@slc` `\caption@slc{`⟨*label*⟩`}{`⟨*text*⟩`}{`⟨*width*⟩`}{`⟨*extra code*⟩`}`

This one does the single-line-check.

```
\newcommand\caption@slc[4]{%
  \caption@Debug{Begin SLC}%
  \begingroup
  \caption@singleline
  \let\caption@hj\@empty
  \caption@calcmargin % calculate #3 if necessary
  \caption@prepareslc
  \sbox\@tempboxa{\caption@@@make{#1}{#2}}%
  \ifdim\wd\@tempboxa>#3%
    \endgroup
  \else
    \endgroup
    \caption@singleline
    #4%
```

```
  \fi
  \caption@Debug{End SLC}}
\newcommand*\caption@singleline{%
  \caption@xsetup\caption@opt@singleline
  \let\caption@fmt\caption@slfmt}
```

\caption@prepareslc `\caption@prepareslc`

Re-define anything which would disturb the single-line-check.

```
\newcommand*\caption@prepareslc{%
  \let\@footnotetext\@gobble\let\@endnotetext\@gobble
  \def\label{\caption@withoptargs\@gobbletwo}%
  \let\stepcounter\caption@l@stepcounter
  \let\refstepcounter\stepcounter\let\H@refstepcounter\stepcounter}
\newcommand*\caption@l@stepcounter[1]{\advance\csname c@#1\endcsname\@ne\relax}
```

\caption@parbox `\caption@parbox{`⟨*width*⟩`}{`⟨*contents*⟩`}`

This macro defines the box which surrounds the caption paragraph.

```
\newcommand*\caption@parbox{\parbox[b]}
```

\caption@@@make `\caption@@@make{`⟨*caption label*⟩`}{`⟨*caption text*⟩`}`

This one finally typesets the caption paragraph, without margin and indention.

```
\newcommand\caption@@@make[2]{%
```

If the label is empty, we use no caption label separator.

```
  \sbox\@tempboxa{#1}%
  \ifdim\wd\@tempboxa=\z@
    \let\caption@lsep\relax
%   \@capbreakfalse
  \fi
```

If the text is empty, we use no caption label separator, too.

```
  \caption@ifempty{#2}{%
    \let\caption@lsep\relax
%   \@capbreakfalse
%   \let\caption@ifstrut\@secondoftwo
  }%
```

Take care that `\caption@parindent` and `\caption@hangindent` will be used to typeset the paragraph.

```
  \@setpar{\@@par\caption@@par}\caption@@par
```

Finally typeset the caption.

```
  \caption@hj\captionfont\captionsize\caption@fmt
    {\ifcaption@star\else{\captionlabelfont#1}\fi}%
    {\ifcaption@star\else{\caption@iflf\captionlabelfont\caption@lsep}\fi}%
    {{\captiontextfont
      \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}%
      \nobreak\hskip\z@skip % enable hyphenation
      \caption@tfmt{#2}%
%     \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}%
      \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}%
      \par}}}
```

\caption@ifempty `\caption@ifempty{⟨text⟩}{⟨true⟩}` (*no ⟨false⟩*)

This one tests if the ⟨*text*⟩ is actually empty.

Note: This will be done without expanding the text, therefore this is far away from being bullet-proof.

Note: This macro is re-defining itself so only the first test (in a group) will actually be done.

```
\newcommand\caption@ifempty[1]{%
  \caption@if@empty{#1}%
  \caption@ifempty\@unused}
\newcommand\caption@if@empty[1]{%
  \def\caption@tempa{#1}%
  \ifx\caption@tempa\@empty
    \let\caption@ifempty\@secondoftwo
  \else
    \expandafter\def\expandafter\caption@tempa\expandafter{%
      \caption@car#1\caption@if@empty\caption@nil}%
    \def\caption@tempb{\caption@if@empty}%
    \ifx\caption@tempa\caption@tempb
      \let\caption@ifempty\@secondoftwo
    \else
      \def\caption@tempb{\ignorespaces}%
      \ifx\caption@tempa\caption@tempb
        \expandafter\caption@if@empty\expandafter{\@gobble#1}%
      \else
        \def\caption@tempb{\label}%
        \ifx\caption@tempa\caption@tempb
          \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
        \else
          \def\caption@tempb{\index}%
          \ifx\caption@tempa\caption@tempb
            \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
          \else
            \def\caption@tempb{\glossary}%
            \ifx\caption@tempa\caption@tempb
              \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
            \else
              \let\caption@ifempty\@gobbletwo
            \fi
          \fi
        \fi
      \fi
    \fi
  \fi}
\long\def\caption@car#1#2\caption@nil{#1}% same as \@car, but \long
```

\caption@@par `\caption@@par`

This command will be executed with every `\par` inside the caption.

```
\newcommand*\caption@@par{%
  \parindent\caption@parindent\hangindent\caption@hangindent}%
```

1.23 Types & sub-types

\DeclareCaptionType `\DeclareCaptionType[⟨options⟩]{⟨environment⟩}[⟨name⟩][⟨list name⟩]`

```
\newcommand*\DeclareCaptionType{%
  \@testopt\@DeclareCaptionType{}}
\@onlypreamble\DeclareCaptionType

\def\@DeclareCaptionType[#1]#2{%
  \def\caption@type{#2}%
  \caption@Debug{New type `#2'}%
  \newcounter{#2}\@namedef{theH#2}{}%
  \KV@caption@DCT@within\caption@within@default
  \KV@caption@DCT@placement{tbp}%
  \@ifundefined{c@float@type}%
    {\newcounter{float@type}%
     \setcounter{float@type}{\@ifundefined{c@figure}14}}%
    {}%
  \caption@Debug{float type `#2'=\the\value{float@type}}%
  \expandafter\xdef\csname ftype@#2\endcsname{\the\value{float@type}}%
  \addtocounter{float@type}{\value{float@type}}%
  \KV@caption@DCT@fileext{lo#2}%
  \@namedef{fnum@#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
  \newenvironment{#2}{\@float{#2}}{\end@float}%
  \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
  \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
  \@ifundefined{l@figure}%
    {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
    {\expandafter\let\csname l@#2\endcsname\l@figure}%

  \expandafter\newcommand\csname #2name\endcsname{}%
  \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
  \uppercase\expandafter{\@tempa}%
  \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
  \@tempb
  \expandafter\let\csname #2name\endcsname\@tempa
  \expandafter\newcommand\csname list#2name\endcsname{}%
  \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%

  \@cons\caption@typelist{{#2}}%
  \caption@setkeys[caption]{caption@DCT}{#1}%

  \@ifundefined{float@exts}{\newtoks\float@exts}{}%
  \let\float@do=\relax
  \edef\@tempa{\noexpand\float@exts{\the\float@exts\float@do{\@nameuse{ext@#2}}}}%
  \@tempa
  \@ifundefined{float@addtolists}{%
    \newcommand\float@addtolists[1]{%
      \def\float@do####1{\addtocontents{####1}{##1}}\the\float@exts}%
    \@ifundefined{@chapter}{}{\caption@PatchChapter}}{}%

  \@ifnextchar[\@@DeclareCaptionType\relax}
\@onlypreamble\@DeclareCaptionType

\def\@@DeclareCaptionType[#1]{%
  \KV@caption@DCT@name{#1}%
  \@ifnextchar[\@@@DeclareCaptionType\relax}
\@onlypreamble\@@DeclareCaptionType

\def\@@@DeclareCaptionType[#1]{%
  \KV@caption@DCT@listname{#1}}
\@onlypreamble\@@@DeclareCaptionType
```

```
\let\DeclareFloatingEnvironment\DeclareCaptionType % old command name
\@onlypreamble\DeclareFloatingEnvironment
```

`\caption@within@default` The default 'within' value.

```
\newcommand*\caption@within@default{\@ifundefined{c@chapter}{none}{chapter}}
\@onlypreamble\caption@within@default
```

`\caption@listof` `\caption@listof{`⟨*float type*⟩`}`

```
\newcommand*\caption@listof[1]{%
  \begingroup
    \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
    \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
    \let\caption@ORI@starttoc\@starttoc
    \renewcommand*\@starttoc[1]{%
      \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
    \listoffigures
  \endgroup}
```

`\caption@typelist` An `\@elt`-list containing the caption types defined with `\DeclareCaptionType`.

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

The available ⟨*options*⟩ are: `fileext=`⟨*file extension*⟩, `listname=`⟨*list name*⟩, `name=`⟨*prosa name*⟩, `placement=`⟨*htbp*⟩, `within=`⟨*none,chapter,section*⟩, and `without`.

```
\define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
\@onlypreamble@key{caption@DCT}{fileext}
\define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
\@onlypreamble@key{caption@DCT}{listname}
\define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
\@onlypreamble@key{caption@DCT}{name}
\define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
\@onlypreamble@key{caption@DCT}{placement}
\define@key{caption@DCT}{within}{%
  \@ifundefined{c@chapter}{}{\@removefromreset\caption@type{chapter}}%
  \@removefromreset\caption@type{section}%
  \begingroup
    \caption@setkeys[caption]{caption@within}{#1}%
  \endgroup}
\@onlypreamble@key{caption@DCT}{within}
\define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
\@onlypreamble@key{caption@DCT}{without}
```

```
\define@key{caption@within}{none}[]{%
  \caption@within{}{}}
\@onlypreamble@key{caption@within}{none}
\@ifundefined{c@chapter}{%
  \define@key{caption@within}{section}[]{%
    \@addtoreset\caption@type{section}%
    \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
}{%
  \define@key{caption@within}{chapter}[]{%
    \@addtoreset\caption@type{chapter}%
    \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
  \@onlypreamble@key{caption@within}{chapter}
  \define@key{caption@within}{section}[]{%
```

```
    \@addtoreset\caption@type{chapter}%
    \@addtoreset\caption@type{section}%
    \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi
                    \ifnum\c@section>\z@ \thesection.\fi}{%
                    \theHchapter.\theHsection.}}
}\@onlypreamble@key{caption@within}{section}
```

`\caption@within` `\caption@within{`⟨*thecode*⟩`}{`⟨*theHcode*⟩`}`

```
\newcommand*\caption@within{%
  \expandafter\caption@within@\expandafter{\caption@type}}
\@onlypreamble\caption@within
\newcommand*\caption@within@[3]{%
  \global\@namedef{the#1}{#2\arabic{#1}}%
  \@ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
    {\global\@namedef{theH#1}{#3\arabic{#1}}}}
\@onlypreamble\caption@within@
```

`\@removefromreset` This code was taken from the remreset package which is part of the 'carlisle' package bundle. (Copyright 1997 David Carlisle)

```
\providecommand*\@removefromreset[2]{{%
  \expandafter\let\csname c@#1\endcsname\@removefromreset
  \def\@elt##1{%
    \expandafter\ifx\csname c@##1\endcsname\@removefromreset
    \else
      \noexpand\@elt{##1}%
    \fi}%
  \expandafter\xdef\csname cl@#2\endcsname{%
    \csname cl@#2\endcsname}}}
```

`\caption@PatchChapter` We try to patch `\@chapter` so `\float@addtolists` will be supported. (Note: The KOMA-Script classes already support `\float@addtolists`.)

```
\newcommand*\caption@PatchChapter{%
  \providecommand*\@chapterlistsgap{10\p@}%
  % report.cls [2005/09/16 v1.4f Standard LaTeX document class]
  \caption@patch@chapter{report}{%
    \ifnum \c@secnumdepth >\m@ne
      \refstepcounter{chapter}%
      \typeout{\@chapapp\space\thechapter.}%
      \addcontentsline{toc}{chapter}%
        {\protect\numberline{\thechapter}##1}%
    \else
      \addcontentsline{toc}{chapter}{##1}%
    \fi
    \chaptermark{##1}%
    \addtocontents{lof}{\protect\addvspace{10\p@}}%
    \addtocontents{lot}{\protect\addvspace{10\p@}}%
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi
  }{%
```

```
    \ifnum \c@secnumdepth >\m@ne
      \refstepcounter{chapter}%
      \typeout{\@chapapp\space\thechapter.}%
      \addcontentsline{toc}{chapter}%
        {\protect\numberline{\thechapter}##1}%
    \else
      \addcontentsline{toc}{chapter}{##1}%
    \fi
    \chaptermark{##1}%
    \ifdim \@chapterlistsgap>\z@
      \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
      \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
      \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
    \fi
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi}%
  % book.cls [2005/09/16 v1.4f Standard LaTeX document class]
  \caption@patch@chapter{book}{%
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \refstepcounter{chapter}%
        \typeout{\@chapapp\space\thechapter.}%
        \addcontentsline{toc}{chapter}%
          {\protect\numberline{\thechapter}##1}%
      \else
        \addcontentsline{toc}{chapter}{##1}%
      \fi
    \else
      \addcontentsline{toc}{chapter}{##1}%
    \fi
    \chaptermark{##1}%
    \addtocontents{lof}{\protect\addvspace{10\p@}}%
    \addtocontents{lot}{\protect\addvspace{10\p@}}%
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi
  }{%
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \refstepcounter{chapter}%
        \typeout{\@chapapp\space\thechapter.}%
        \addcontentsline{toc}{chapter}%
          {\protect\numberline{\thechapter}##1}%
      \else
        \addcontentsline{toc}{chapter}{##1}%
      \fi
    \else
```

```
      \addcontentsline{toc}{chapter}{##1}%
    \fi
    \chaptermark{##1}%
    \ifdim \@chapterlistsgap>\z@
      \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
      \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
      \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
    \fi
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi}%
  % amsbook.cls [2004/08/06 v2.20]
  % smfbook.cls [1999/11/15 v1.2f Classe LaTeX pour les monographies editees par
  \caption@patch@chapter{ams/smfbook}{%
    \refstepcounter{chapter}%
    \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
    \else \let\@secnumber\thechapter \fi
    \typeout{\chaptername\space\@secnumber}%
    \def\@toclevel{0}%
    \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
    \else \@tocwriteb\tocchapter{chapter}{##2}\fi
    \chaptermark{##1}%
    \addtocontents{lof}{\protect\addvspace{10\p@}}%
    \addtocontents{lot}{\protect\addvspace{10\p@}}%
    \@makechapterhead{##2}\@afterheading
  }{%
    \refstepcounter{chapter}%
    \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
    \else \let\@secnumber\thechapter \fi
    \typeout{\chaptername\space\@secnumber}%
    \def\@toclevel{0}%
    \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
    \else \@tocwriteb\tocchapter{chapter}{##2}\fi
    \chaptermark{##1}%
    \ifdim \@chapterlistsgap>\z@
      \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
      \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
      \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
    \fi
    \@makechapterhead{##2}\@afterheading}%
  % scrreprt/scrbook.cls
  \@ifundefined{KOMAClassName}{}{%
    \caption@Debug{document class `\KOMAClassName' detected}%
   \let\caption@patch@chapter\@gobblethree}%
  % rapport1/3.cls [2004/06/07 v2.1a NTG LaTeX document class]
  \caption@patch@chapter{rapport}{%
    \ifnum \c@secnumdepth >\m@ne
      \refstepcounter{chapter}%
      \typeout{\@chapapp\space\thechapter.}%
      \addcontentsline{toc}{chapter}%
```

```
        {\protect\numberline{\thechapter}\toc@font0 ##1}%
    \else
      \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
    \fi
    \chaptermark{##1}%
    \addtocontents{lof}{\protect\addvspace{10\p@}}%
    \addtocontents{lot}{\protect\addvspace{10\p@}}%
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi
  }{%
    \ifnum \c@secnumdepth >\m@ne
      \refstepcounter{chapter}%
      \typeout{\@chapapp\space\thechapter.}%
      \addcontentsline{toc}{chapter}%
        {\protect\numberline{\thechapter}\toc@font0 ##1}%
    \else
      \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
    \fi
    \chaptermark{##1}%
    \ifdim \@chapterlistsgap>\z@
      \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
      \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
      \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
    \fi
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi}%
  % boek(3).cls [2004/06/07 v2.1a NTG LaTeX document class]
  \caption@patch@chapter{boek}{%
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \refstepcounter{chapter}%
        \typeout{\@chapapp\space\thechapter.}%
        \addcontentsline{toc}{chapter}%
          {\protect\numberline{\thechapter}\toc@font0 ##1}%
      \else
        \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
      \fi
    \else
      \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
    \fi
    \chaptermark{##1}%
    \addtocontents{lof}{\protect\addvspace{10\p@}}%
    \addtocontents{lot}{\protect\addvspace{10\p@}}%
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
```

```
      \@makechapterhead{##2}%
      \@afterheading
    \fi
  }{%
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \refstepcounter{chapter}%
        \typeout{\@chapapp\space\thechapter.}%
        \addcontentsline{toc}{chapter}%
          {\protect\numberline{\thechapter}\toc@font0 ##1}%
      \else
        \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
      \fi
    \else
      \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
    \fi
    \chaptermark{##1}%
    \ifdim \@chapterlistsgap>\z@
      \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
      \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
      \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
    \fi
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi}%
  % thesis.cls [1996/25/01 1.0g LaTeX document class (wm).]
  \caption@patch@chapter{thesis}{%
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \refstepcounter{chapter}%
        \typeout{\chaptername\space\thechapter.}
        \if@thema
          \ifx\@shortauthor\@empty
            \addcontentsline{toc}{chapter}{%
            \protect\numberline{\thechapter.}##1}%
          \else
            \addcontentsline{toc}{chapter}{%
            \protect\numberline{\thechapter.}%
            \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
          \fi
        \else
          \addcontentsline{toc}{chapter}{%
          \protect\numberline{\thechapter.}##1}%
        \fi
      \else
        \addcontentsline{toc}{chapter}{##1}
      \fi
    \else
      \addcontentsline{toc}{chapter}{##1}
    \fi
    \chaptermark{##1}
```

```
    \addtocontents{lof}{\protect\addvspace{10pt}}
    \addtocontents{lot}{\protect\addvspace{10pt}}
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]
    \else
      \@makechapterhead{##2}
      \@afterheading
    \fi
  }{%
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \refstepcounter{chapter}%
        \typeout{\chaptername\space\thechapter.}%
        \if@thema
          \ifx\@shortauthor\@empty
            \addcontentsline{toc}{chapter}{%
            \protect\numberline{\thechapter.}##1}%
          \else
            \addcontentsline{toc}{chapter}{%
            \protect\numberline{\thechapter.}%
            \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
          \fi
        \else
          \addcontentsline{toc}{chapter}{%
          \protect\numberline{\thechapter.}##1}%
        \fi
      \else
        \addcontentsline{toc}{chapter}{##1}%
      \fi
    \else
      \addcontentsline{toc}{chapter}{##1}%
    \fi
    \chaptermark{##1}%
    \ifdim \@chapterlistsgap>\z@
      \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
      \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
      \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
    \fi
    \if@twocolumn
      \@topnewpage[\@makechapterhead{##2}]%
    \else
      \@makechapterhead{##2}%
      \@afterheading
    \fi}%
  \ifx\caption@patch@chapter\@gobblethree \else
    \caption@Debug{%
      Unsupported document class detected,\MessageBreak
      or \noexpand\@chapter was redefined by another package}%
  \fi
  \let\caption@PatchChapter\@undefined}
\@onlypreamble\caption@PatchChapter

\newcommand\caption@patch@chapter[3]{%
  \begingroup
```

```
%   \let\if@twocolumn\iffalse
    \let\if@mainmatter\iffalse
    \let\if@thema\iffalse
    \def\@tempa[##1]##2{#2}%
    \ifx\@tempa\@chapter
      \caption@Debug{document class `#1' detected}%
      \gdef\@chapter[##1]##2{#3}%
      \global\let\caption@patch@chapter\@gobblethree
    \fi
  \endgroup}
\@onlypreamble\caption@patch@chapter

\long\def \@gobblethree #1#2#3{}
```

\DeclareCaptionSubType

\DeclareCaptionSubType[⟨*numbering scheme*⟩]{⟨*type*⟩}
\DeclareCaptionSubType*[⟨*numbering scheme*⟩]{⟨*type*⟩}
The starred variant provides the numbering format ⟨*type*⟩ . ⟨*subtype*⟩ while the non-starred variant simply uses ⟨*subtype*⟩.

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

\newcommand*\@DeclareCaptionSubType[1]{%
  \@testopt{\@@DeclareCaptionSubType{#1}}{alph}}
\@onlypreamble\@DeclareCaptionSubType

\def\@@DeclareCaptionSubType#1[#2]#3{%
  \@ifundefined{c@#3}%
    {\caption@Error{No float type '#3' defined}}%

    {\@ifundefined{c@sub#3}%
       {\caption@Debug{New subtype `sub#3'}%
        \newcounter{sub#3}%
        \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
        \@ifundefined{l@chapter}%
          {\edef\@tempa{\expandafter\expandafter\expandafter\noexpand
                        \expandafter\@car\l@subsubsection\@nil}%
           \def\@tempb{\@dottedtocline}%
           \ifx\@tempa\@tempb % \l@subsubsection starts with \@dottedtocline
             \expandafter\edef\csname l@sub#3\endcsname{%
               \noexpand\@dottedtocline{2}%
               \expandafter\expandafter\expandafter\noexpand
               \expandafter\@gobbletwo\l@subsubsection}%
           \else
             \@namedef{l@sub#3}{\@dottedtocline{2}{3.8em}{3.2em}}%
           \fi}%
          {\expandafter\let\csname l@sub#3\endcsname\l@subsection}%
        \@cons\caption@subtypelist{{#3}}}%
       {\caption@Debug{Modify caption `sub#3'}}%

     \@namedef{sub#3name}{}%
     \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
     #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
     {\@namedef{p@sub#3}{}%
      \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
     {\@namedef{p@sub#3}{\csname the#3\endcsname}%
      \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
```

```
      \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}%
    }}
\@onlypreamble\@@DeclareCaptionSubType
```

`\caption@subtypelist` An `\@elt`-list containing the subtypes defined with `\DeclareCaptionSubType`.

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

`\caption@For` `\caption@For{`⟨*elt-list*⟩`}{`⟨*code with #1*⟩`}`
`\caption@For*{`⟨*elt-list*⟩`}{`⟨*code with #1*⟩`}`

```
\newcommand*\caption@For{\caption@withoptargs\caption@@For}
\@onlypreamble\caption@For
\newcommand\caption@@For[3]{%
  \caption@AtBeginDocument#1{%
    \def\@elt##1{#3}%
    \@nameuse{caption@#2}%
    \let\@elt\relax}}%
\@onlypreamble\caption@@For
```

1.24 subfig package adaptions

We have to make several adaptions to the caption package *v3.1* here.

```
\caption@AtBeginDocument{%
  \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
  \ifx\subref\@tempa
    \PackageInfo{caption3}{subfig package 1.2 or 1.3 is loaded\@gobble}%
    \let\caption@setfloattype\@gobble
    \let\@dottedxxxline\sf@NEW@dottedxxxline
    \let\sf@subfloat\sf@NEW@subfloat
```

This is a very small bugfix for *v1.2* and *v1.3* or the subfig package, making `\subref` robust, so it works in captions, too.

```
    \DeclareRobustCommand*\subref{\@ifstar\sf@@subref\sf@subref}%
  \fi
  \let\sf@NEW@dottedxxxline\@undefined
  \let\sf@NEW@subfloat\@undefined}
\def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
  \begingroup
    \caption@setfloattype{#1}%
    \caption@setoptions{subfloat}%
    \caption@setoptions{sub#1}%
    \ifnum #3>\@nameuse{c@#2depth}\else
      \@dottedtocline{\z@}{#4}{#5}{#6}{#7}%
    \fi
  \endgroup}
\def\sf@NEW@subfloat{%
  \begingroup
    \caption@setfloattype\@captype
    \sf@ifpositiontop{%
      \maincaptiontoptrue
    }{%
      \maincaptiontopfalse
```

```
1661    }%
1662    \caption@setoptions{subfloat}%
1663    \caption@setoptions{sub\@captype}%
1664    \let\sf@oldlabel=\label
1665    \let\label=\subfloat@label
1666    \ifmaincaptiontop\else
1667      \advance\@nameuse{c@\@captype}\@ne
1668    \fi
1669    \refstepcounter{sub\@captype}%
1670    \setcounter{sub\@captype @save}{\value{sub\@captype}}%
1671    \@ifnextchar [%  %] match left bracket
1672      {\sf@@subfloat}%
1673      {\sf@@subfloat[\@empty]}}
```

2 Main package

2.1 Identification

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{caption}[2008/08/24 v3.1j Customizing captions (AR)]
%\@ifundefined{PackageRedefines}{}{\PackageRedefines{caption}{caption}}
```

\caption@Info *Note:* The \@gobble at the end of the 2nd argument of \PackageInfo suppresses the line number info. See TLC2[1], A.4.7, p885 for details.

```
\newcommand*\caption@Info[1]{\PackageInfo{caption}{#1\@gobble}}
\@onlypreamble\caption@Info
```

2.2 Loading the kernel

```
\RequirePackage{caption3}[2008/08/24] % needs v3.1j or newer
```

2.3 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}}{}}
```

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

```
\newcommand*\caption@DisablePositionOption[1]{%
  \caption@Info{%
    '#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
```

2.4 Check document class

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

2.5 Adaption to the $\mathcal{AMS}$ & SMF document classes

```
\@ifundefined{@captionheadfont}{}{%
  \caption@Info{AMS or SMF document class}%
```

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

2.6 Emulation of the KOMA-Script commands

```
\@ifundefined{scr@caption}{}{%
  \caption@Info{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
```

\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

```
  \g@addto@macro\@captionabovetrue{\let\caption@position\@firstoftwo}
  \g@addto@macro\@captionabovefalse{\let\caption@position\@secondoftwo}
```

\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}}

  \@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
    \@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
}
```

2.7 Declaration of options

2.7.1 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}}
\DeclareCaptionOption{tablename}{\caption@SetName{table}{#1}}
\DeclareCaptionOption{name}{\caption@setname\@captype{#1}}

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

\caption@SetName `\caption@SetName{`⟨*cmd*⟩`}{`⟨*value*⟩`}`

```
\newcommand*\caption@SetName[2]{%
  \caption@setname{#1}{#2}%
  \begingroup
    \@ifundefined{languagename}{}{%
      \@ifundefined{captions\languagename}{}{%
        \expandafter\g@addto@macro\csname captions\languagename\endcsname
          {\caption@setname{#1}{#2}}}}%
  \endgroup}

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

\caption@AtBeginDocument{\let\caption@SetName\caption@setname}

\DeclareCaptionOption{figurewithin}{\caption@Within{figure}{#1}}
\@onlypreamble@key{caption}{figurewithin}
\DeclareCaptionOption{figurewithout}{\KV@caption@figurewithin{none}}
\@onlypreamble@key{caption}{figurewithout}

\DeclareCaptionOption{tablewithin}{\caption@Within{table}{#1}}
\@onlypreamble@key{caption}{tablewithin}
\DeclareCaptionOption{tablewithout}{\KV@caption@tablewithin{none}}
\@onlypreamble@key{caption}{tablewithout}

\DeclareCaptionOption{within}{%
  \@ifundefined{c@figure}{}{\caption@Within{figure}{#1}}%
  \@ifundefined{c@table}{}{\caption@Within{table}{#1}}%
  \def\caption@within@default{#1}}
\@onlypreamble@key{caption}{within}
\DeclareCaptionOption{without}{\KV@caption@within{none}}
\@onlypreamble@key{caption}{without}
```

\caption@within

```
\newcommand*\caption@Within[1]{\def\caption@type{#1}\KV@caption@DCT@within}
\@onlypreamble\caption@Within
```

2.7.2 Miscellaneous options

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

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

2.7.3 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*{anne}{%
  \caption@setformat{plain}%
  \caption@setjustification{centerlast}}
\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}}
```

2.7.4 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}}
```

2.7.5 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}}
```

2.7.6 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\@tempa{\@car#2\@nil}%
  \caption@setbool{FP#1cap}{\if c\@tempa 1\else 0\fi}}
\@onlypreamble\caption@setFPoption
```

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

2.7.7 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}
```

2.8 Processing of options

```
\caption@ProcessOptions*{caption}
```

2.9 `\captionof` and `\captionlistentry`

```
\caption@AtBeginDocument{%
```

```
  \DeclareCaptionOption{type}{\caption@settype{#1}}%
  \DeclareCaptionOption{type*}{\caption@settype*{#1}}%
```

```
  \DeclareCaptionOption{subtype}[sub\@captype]{\caption@setsubtype{#1}}%
  \DeclareCaptionOption{subtype*}[sub\@captype]{\caption@setsubtype*{#1}}%
```

```
}
```

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.

`\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@@settype{}}
\newcommand*\caption@setsubtype{%
  \caption@iftype
    {\caption@@settype{sub}}%
    {\caption@Error{Option `subtype=' outside float}}}%
\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"
  \@ifundefined{c@#3}%
    {\caption@Error{No float type `#3' defined}}%
    {\caption@Debug{#1type=#3}%
     \caption@checkgrouplevel{#1}{%
       \captionsetup{#1type#2*\@empty=...}#2{ or
                     \@backslashchar#1captionof}{}}%
     \edef\@tempa{#3}%
     \expandafter\ifx\csname @#1captype\endcsname\@tempa \else
       \ifcaptionsetup@star\else\@nameuse{caption@#1type@warning}\fi
     \fi
     \expandafter\let\csname @#1captype\endcsname\@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
     \let\caption@ifrefstepcounter\@secondoftwo
     #2{}{%
       \let\@currentlabel\caption@undefinedlabel
%      \let\@currentHlabel\@undefined
       \ifx\caption@ORI@label\@undefined
         \let\caption@ORI@label\label
         \let\label\caption@xlabel
       \fi
       \caption@start}}}
```

`\caption@typehook` Hook, 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{%
  \@ifundefined{@captype}{\let\@captype\@undefined\@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)

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname currentgrouplevel\endcsname\relax
  \caption@Debug{TeX engine: TeX}
  \let\caption@checkgrouplevel\@gobbletwo
\else
  \caption@Debug{TeX engine: e-TeX}
  \newcommand*\caption@checkgrouplevel[2]{%
    \@ifundefined{#1caption@grouplevel}{%
       \@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
    }{}}
\fi
```

`\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}}
\newcommand*\caption@xref[2]{%
  \caption@WarningNoLine{\noexpand\label before \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[1]{%
  \caption@@xlabel
  \def\caption@labelname{#1}%
  \caption@ORI@label{#1}}
\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}}
```

```
1979 \newcommand*\caption@of[2]{\caption@settype*{#2}#1}
```

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

```
1980 \newcommand*\captionlistentry{%
1981   \caption@teststar\@captionlistentry\@firstoftwo\@secondoftwo}
1982 \newcommand*\@captionlistentry[1]{%
1983   \@testopt{\caption@listentry{#1}}\@captype}
1984 \def\caption@listentry#1[#2]#3{%
1985   \@bsphack
1986     #1{\def\@currentlabelname{#3}}%
1987       {\caption@refstepcounter{#2}%
1988        \caption@makecurrent{#2}{#3}}%
1989     \caption@addcontentsline{#2}{#3}%
1990   \@esphack}
```

2.10 \ContinuedFloat

\ContinuedFloat `\ContinuedFloat`
`\ContinuedFloat*`

This mainly decrements the appropriate counter and increments the continuation counter instead. Furthermore we set `\caption@resetContinuedFloat` to `\@gobble` so the continuation counter will not be reset to zero inside `\caption@refstepcounter`. Please forget about the optional argument, it was never working well, is incompatible to the subfig package, but is still there for compatibility reasons.

Note: The definition of `\ContinuedFloat` itself is compatible to the one inside the subfig package, except for the starred variant and the optional argument.

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.

```
1991 \def\ContinuedFloat{%
1992   \@ifnextchar[\@Continued@Float\@ContinuedFloat}
1993 \def\@Continued@Float[#1]{\addtocounter{#1}\m@ne}
1994 \def\@ContinuedFloat{%
1995   \caption@iftype
1996     {\addtocounter\@captype\m@ne
1997      \caption@ContinuedFloat\@captype}%
1998     {\caption@Error{\noexpand\ContinuedFloat outside float}}}
1999 \def\caption@ContinuedFloat#1{%
2000   \@ifstar{\caption@Continued@Float@{#1}}{\caption@Continued@Float{#1}}}
```

```
\def\caption@Continued@Float@{%
  \addtocounter\@captype\@ne
  \@stpelt{ContinuedFloat}\stepcounter{ContinuedFloat}%
  \def\caption@resetContinuedFloat##1{\xdef\caption@CFtype{##1}}%
  \caption@@ContinuedFloat}
\def\caption@Continued@Float#1{%
  \edef\@tempa{#1}%
  \ifx\@tempa\caption@CFtype
    \stepcounter{ContinuedFloat}%
    \let\caption@resetContinuedFloat\@gobble
    \caption@@ContinuedFloat{#1}%
    \sf@ContinuedFloat{#1}%
  \else
    \caption@Error{Continued '#1' after '\caption@CFtype'}%
  \fi}
\def\caption@@ContinuedFloat#1{%
  \expandafter\l@addto@macro\csname the#1\endcsname\theContinuedFloat
  \@ifundefined{theH#1}{}{%
    \expandafter\l@addto@macro\csname theH#1\endcsname{%
      \@alph\c@ContinuedFloat}}%
  \caption@setoptions{ContinuedFloat}%
  \caption@setoptions{continued#1}}
\providecommand*\sf@ContinuedFloat[1]{}
\newcommand*\caption@CFtype{??}
```

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

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

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

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

2.11 Internal helpers

`\caption@refstepcounter` Resets the continuation counter, increments the float (i.e. `figure` or `table`) counter, and sets the `refstepcounter` flag.

```
\newcommand*\caption@refstepcounter[1]{%
  \caption@resetContinuedFloat{#1}%
  \caption@@refstepcounter{#1}%
  \let\caption@ifrefstepcounter\@firstoftwo}
\newcommand*\caption@@refstepcounter{\refstepcounter}
\let\caption@ifrefstepcounter\@secondoftwo
```

`\caption@dblarg` A `\relax` was added compared to `\@dblarg` so `\caption{}` will be expanded to `\caption[\relax]{}` (and not to `\caption[]{}`).

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

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

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

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

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

4. Make an entry in the list-of-whatever.
5. Set `\caption@ifempty` according argument ⟨*heading*⟩.

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

`\caption@end` `\caption@end` closes the group.

```
\newcommand*\caption@end{%
  \endgroup
  \let\caption@position\caption@fixedposition}
```

`\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{%
  \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@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` `\caption@makeanchor` `\caption@start` `\caption@@start` `\caption@freezeHref` `\caption@defrostHref` All these are needed for support of the hyperref package.

```
\newcommand*\caption@makecurrent[2]{}
\let\caption@makeanchor\@firstofone
\let\caption@start\relax
\let\caption@@start\relax
\let\caption@freezeHref\relax
\let\caption@defrostHref\relax
```

2.12 `\caption`, `\@caption`, and `\@makecaption`

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 captcont package, which brings it's own definition of \caption*, was quite common these days.
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@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)]
    \renewcommand\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@IfCheckCommand{}{%
    \caption@Info{%
      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\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{%
    % 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{%
    % 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@Info{%
      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.

```
  \@ifundefined{caption@ifcompatibility}%
    {\let\caption@ifcompatibility\caption@ifincompatible
     \let\@tempa\caption@WarningNoLine}%
    {\let\@tempa\@gobble}% suppress warning
  \caption@ifcompatibility{%
    \@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}%
```

`\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
      \@ifundefined{caption@ORI@capstart}{}{%
        \caption@Debug{%
          Restore hypcap definition of \string\capstart\@gobble}%
        \let\capstart\caption@ORI@capstart}%
      \@ifundefined{caption@ORI@float@makebox}{}{%
        \caption@Debug{%
          Restore hyperref redefinition of \string\float@makebox\@gobble}%
        \let\float@makebox\caption@ORI@float@makebox}%
    }%
```

`\caption@star` We define `\caption@star` here, too, so it's defined but does not make any harm.

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

\caption@star A helper macro which processes the optional * after \caption.
Note: We set \caption@startrue globally so it works with the sidecap package, too.

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

\@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#1[#2]#3{%
  \ifcaption@star \else
    \caption@prepareanchor{#1}{#2}%
  \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}%
```

\caption@prepareanchor

```
\newcommand*\caption@prepareanchor[2]{%
  \caption@makecurrent{#1}{#2}%
  \caption@ifhypcap\caption@@start{}}
}%
\caption@AtBeginDocument*{%
  \let\caption@ORI@capstart\@undefined
  \let\caption@ORI@float@makebox\@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}}%
}
```

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

2.13 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@c@#1}{0}%
  \@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[18]...

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

\caption@subtypehook Hook, will be used inside \caption@setsubtype.

```
\newcommand*\caption@subtypehook{%
  \ifx\caption\caption@subcaption \else
    \caption@ifrefstepcounter{}{%
      % no \caption or \subcaption in this (floating) environment yet
      \caption@Debug{Increment \@captype\ counter =\the\value\@captype}%
      \caption@l@stepcounter\@captype
      \let\addcontentsline\caption@addsubcontentsline}%
    \ifnum\csname caption@c@\@captype\endcsname=\value\@captype \else
      \caption@Debug{Reset sub\@captype\ counter}%
      \expandafter\xdef\csname caption@c@\@captype\endcsname{%
        \the\value\@captype}%
      \@stpelt\@subcaptype
    \fi
```

```
    \c@ContinuedFloat=0\relax
    \let\caption@resetContinuedFloat\@gobble
    \let\caption@addcontentsline\caption@kernel@addcontentsline
    \let\caption@setfloatcapt\@firstofone
    \caption@clearmargin
    \caption@iflist{}{\let\caption@setlist\@gobble}%
    \caption@setoptions{sub}%
    \caption@setoptions{subfloat}% for subfig-package compatibility
    \let\caption\caption@subcaption
  \fi}%
```

`\caption@subcaption` Makes a sub-caption.

```
\newcommand*\caption@subcaption{%
  \caption@iftype
    {\caption@checkgrouplevel{sub}\subcaption
     \caption@star
       {\caption@refstepcounter\@subcaptype}%
       {\caption@dblarg{\@caption\@subcaptype}}}%
    {\caption@Error{\noexpand\subcaption outside float}}}
```

`\caption@addcontentsline` We extend `\caption@addcontentsline` so it handles sub-captions, too.
Note: `\sf@ifpositiontop` & `\@listsubcaptions` are defined by the subfigure & subfig packages.

```
\let\caption@kernel@addcontentsline\caption@addcontentsline
\renewcommand*\caption@addcontentsline[2]{%
  \sf@ifpositiontop{\@listsubcaptions{#1}}{}%
  \caption@kernel@addcontentsline{#1}{#2}%
  \sf@ifpositiontop{}{\@listsubcaptions{#1}}%
  \caption@addsubcontentslines{#1}}
\newcommand*\caption@addsubcontentslines[1]{%
  \begingroup
    \caption@subcontentslines
  \endgroup
  \caption@clearsubcontentslines}%

\caption@AtBeginDocument*{%
  \@ifundefined{sf@ifpositiontop}{\let\sf@ifpositiontop\@gobbletwo}{}%
  \caption@clearsubcontentslines
  \g@addto@macro\caption@typehook{\caption@checksubcontentslines}%
  \AtEndDocument{\caption@checksubcontentslines}}%
```

`aption@addsubcontentsline` Add a pending sub-caption list entry.

```
\newcommand*\caption@addsubcontentsline[3]{%
  \begingroup
  \let\label\@gobble \let\index\@gobble \let\glossary\@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
      \protect\addcontentsline{#1}{#2}{#3}}}%
  \@tempa}
```

ion@checksubcontentslines Checks if the list of pending sub-captions is empty, if not, a warning will be issued.

```
\newcommand*\caption@checksubcontentslines{%
  \ifx\caption@subcontentslines\@empty \else
    \caption@Error{%
      Something's wrong--perhaps a missing \protect\caption\MessageBreak
      in the last figure or table}%
    \caption@clearsubcontentslines
  \fi}
```

ion@clearsubcontentslines Clear pending sub-caption list entries.

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

2.14 Document class & Babel package support

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

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

2.14.2 The beamer class

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

Since the beamer class do not offer a 'list of figures' we switch this support in the caption package off.

```
  \captionsetup{list=false}
  \DeclareCaptionOption{list}[1]{}
  \DeclareCaptionOption{listof}[1]{}
```

\figure \table We redefine `figure` & `table` so our type-specific options will be used, a hyperref anchor will be set 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}}

}{}
```

2.14.3 The KOMA-Script classes

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

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

2.14.4 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}
```

2.14.5 The frenchle/pro package

```
\caption@AtBeginDocument{\@ifundefined{frenchTeXmods}{}{%
  \caption@Info{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`.

```
  \@ifundefined{GOfrench}%
    {\let\@tempa\@firstofone}%
    {\def\@tempa{\g@addto@macro\GOfrench}}%
  \@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\@tempa\@empty
    \ifx\f@ffrench\fnum@figure
      \l@addto@macro\@tempa{\let\fnum@figure\f@ffrench}%
    \fi
    \ifx\f@tfrench\fnum@table
      \l@addto@macro\@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}%
    \@tempa
```

```
  }%
}}
```

2.15 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@@ifpackageloaded{#1}[#2]{#3}{#4}}}
\@onlypreamble\caption@@IfPackageLoaded

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

\long\def\caption@@ifpackageloaded#1[#2]{%
  \@ifpackageloaded{#1}{%
    \caption@Info{#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@@ifpackageloaded
```

\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@freeze `\caption@freeze*`

Used by the fltpage & sidecap package support.

```
  \newcommand*\caption@freeze{%
    \caption@teststar\caption@@freeze\@gobble\@firstofone}%

  \newcommand*\caption@@freeze[1]{%
    \global\let\caption@SCcontinued\relax
```

```
    \global\let\caption@SCsetup\@undefined
    \global\let\caption@SClentry\@undefined
    \global\let\caption@SCtext\@undefined
    \global\let\caption@SClabel\@undefined
    \let\caption@ORI@ContinuedFloat\ContinuedFloat
    \def\ContinuedFloat{%
      \caption@withoptargs\caption@SC@ContinuedFloat}%
    \def\caption@SC@ContinuedFloat##1{%
      \let\caption@ORI@setcounter\setcounter
      \let\caption@ORI@addtocounter\addtocounter
      \def\setcounter####1####2{\csname c@####1\endcsname####2\relax}%
      \def\addtocounter####1####2{\advance\csname c@####1\endcsname ####2\relax}%
      \caption@ORI@ContinuedFloat##1%
      \global\let\caption@SCcontinued\caption@ORI@ContinuedFloat
      \let\setcounter\caption@ORI@setcounter
      \let\addtocounter\caption@ORI@addtocounter}%
    \let\caption@ORI@setup\captionsetup
    \def\captionsetup{%
      \caption@withoptargs\caption@SC@setup}%
    \def\caption@SC@setup##1##2{%
      \caption@g@addto@list\caption@SCsetup{##2}%
      \caption@ORI@setup##1{##2}}%
    \let\caption@ORI\caption
    \def\caption{%
      \def\caption{\caption@Error{%
        Only one \noexpand\caption can be placed in this environment}}%
      \let\captionsetup\caption@setup
      \let\caption@@refstepcounter\caption@l@stepcounter
      \caption@ORI}%
    \long\def\@caption##1[##2]##3{%
      \@bsphack
        \gdef\caption@SClentry{##2}%
        \gdef\caption@SCtext{##3}%
      \@esphack}%
    #1{% is \@gobble in star form, and \@firstofone otherwise
      \def\label##1{\@bsphack\gdef\caption@SClabel{##1}\@esphack}}%
}%
```

\caption@defrost

```
\newcommand*\caption@defrost{%
  \ifx\caption@ORI@ContinuedFloat\@undefined
    \caption@defrost@setup
    \ifx\caption@SCtext\@undefined \else
      \expandafter\expandafter\expandafter\caption
        \expandafter\expandafter\expandafter[%
        \expandafter\expandafter\expandafter{%
        \expandafter\caption@SClentry\expandafter}\expandafter]%
        \expandafter{\caption@SCtext}%
    \fi
    \ifx\caption@SClabel\@undefined \else
      \expandafter\label\expandafter{\caption@SClabel}%
    \fi
  \else
```

```
      \caption@Error{Internal Error:\MessageBreak
        \noexpand\caption@defrost in same group as \string\caption@freeze}%
    \fi}%
  \newcommand*\caption@defrost@setup{%
    \caption@SCcontinued
    \ifx\caption@SCsetup\@undefined \else
      \expandafter\captionsetup\expandafter{\caption@SCsetup}%
    \fi}%
  }{}%
  \caption@undefbool{needfreeze}}
```

2.15.1 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@ifpackageloaded{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{%
        \let\@makecaption\caption@@make
        \global\setbox\@floatcapt\vbox}%
```

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}}%
```

`\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@ifstyle` `\float@ifstyle{⟨type⟩}{⟨if-clause⟩}{⟨else-clause⟩}`
Checks if the given ⟨*type*⟩ (e.g. `figure`) is associated with a float style (e.g. `boxed`).

```
\providecommand*\float@ifstyle[1]{%
  \expandafter\ifx\csname fst@#1\endcsname\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi}%
```

`\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@setstyle` `\float@setstyle{`⟨*type*⟩`}{`⟨*style*⟩`}`
Sets or changes the float style associated with ⟨*type*⟩.

```
\providecommand*\float@setstyle[2]{%
  \expandafter\edef\csname fst@#1\endcsname{%
    \expandafter\noexpand\csname fs@#2\endcsname}}%
```

`\float@dostyle` `\float@dostyle{`⟨*type*⟩`}`

```
\providecommand*\float@dostyle[1]{%
  \@nameuse{fst@#1}\@float@setevery{#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@ifstyle[1]{\@secondoftwo}%
  \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}
```

2.15.2 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@setoptions{floating#1}%
    \let\caption@boxrestore\@parboxrestore}%

}{}
```

2.15.3 The fltpage package

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

\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}%
}%
```

```
2684  \renewcommand*\FP@floatBegin[1]{%
2685    \def\@captype{#1}%
2686    \let\FP@LabelText\@empty
2687    \begin{lrbox}{\FP@floatCorpusBOX}%
2688    \caption@ifFPrefcap
2689      {\caption@freeze\relax}%
2690      {\def\label##1{\@bsphack\gdef\FP@LabelText{##1}\@esphack}%
2691       \caption@freeze*}}%
```

\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{%
    \end{lrbox}%
    \stepcounter{FP@\@captype C}%
    \caption@label\FP@positionLabel
    \FP@helpNote\@captype\FP@positionLabel
    \edef\FP@RestoreCounter{%
      \noexpand\setcounter{\@captype}{\the\value\@captype}%
      \noexpand\setcounter{ContinuedFloat}{\the\value{ContinuedFloat}}}%
    \FP@float
      {\FP@positionLabel}% location label test
      {\begin\@captype[p!]%
         \usebox\FP@floatCorpusBOX
         \caption@defrost@setup
         \caption@ifFPlistcap
           {\caption@refstepcounter\@captype
            \expandafter\caption@makecurrent\expandafter\@captype
                                        \expandafter{\caption@SClentry}}%
           {\expandafter\captionlistentry\expandafter{\caption@SClentry}}%
         \caption@makeanchor\relax
         \ifx\FP@LabelText\@empty \else
           \expandafter\label\expandafter{\FP@LabelText}%
         \fi
       \end\@captype}%
      {\FP@RestoreCounter
       \@ifundefined{theH\@captype}{}{%
         \expandafter\l@addto@macro\csname theH\@captype\endcsname{.FP}}}%
      {\begin\@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 captio
         \setlength\belowcaptionskip{2pt plus 2pt minus 1pt}% length below captio
         \caption@setoptions{FP\@captype}%
         \FP@separatorCaption
         \caption@ifFPlistcap{}{\let\caption@addcontentsline\@gobbletwo}%
         \caption@defrost
       \end\@captype}%
  }%

  \caption@For{typelist}{%
    \newenvironment{FP#1}{\FP@floatBegin{#1}}{\FP@floatEnd}}%
}{%
  \let\caption@ifFPlistcap\@undefined
  \let\caption@ifFPrefcap\@undefined
}
```

2.15.4 The hyperref package

```
\caption@IfPackageLoaded{hyperref}[2003/11/30 v6.74m]{%
  \@ifundefined{hyper@makecurrent}{% hyperref has stopped early
    \caption@WarningNoLine{%
      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}%
      \def\@currentlabelname{#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
        \hyper@@anchor{\@currentHref}{#1}%
      \else
        \Hy@raisedlink{\hyper@@anchor{\@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 `\capstrart` our variant is called `\caption@start` and is controlled by the option `hypcap=false/true`.

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

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[10] 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
```

```
2789          \caption@Debug{hypcap anchor: #1 (horizontal mode)}%
2790          \caption@anchor{#1}%
2791        \fi}%
```

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

```
2792    \newcommand*\caption@anchor[1]{%
2793      \ifmeasuring@ \else
2794        \caption@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
2795      \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*.)

```
2796    \ifx\HyperRaiseLinkLength\@tempdima
2797      \def\caption@raisedlink#1{\ifvmode#1\else\Hy@raisedlink{#1}\fi}%
2798    \else
2799      \let\caption@raisedlink\Hy@raisedlink
2800    \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.

```
2801    \def\caption@@start{%
2802      \@ifundefined{hc@currentHref}{%
2803        \caption@Warning{%
2804          The option 'hypcap=true' will be ignored for this\MessageBreak
2805          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`.

```
2806    \def\caption@freezeHref{%
2807      \let\caption@ORI@start\caption@start
2808      \def\caption@start{\let\caption@start\caption@ORI@start}%
2809 %    \let\caption@ORI@@start\caption@@start
2810 %    \l@addto@macro\caption@subtypehook{%
2811 %      \let\caption@@start\caption@ORI@@start}%
2812      \global\let\caption@currentHref\@undefined
2813      \def\caption@@start{\global\let\caption@currentHref\@currentHref}%
2814      \let\caption@ORI@setfloatcapt\caption@setfloatcapt
2815      \renewcommand*\caption@setfloatcapt{%
2816        \ifx\caption@currentHref\@undefined \else
2817          \let\caption@makeanchor\@firstofone
2818        \fi
2819        \caption@ORI@setfloatcapt}}%
```

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

```
2820    \def\caption@defrostHref{%
2821      \ifx\caption@currentHref\@undefined \else
2822        \caption@startanchor\caption@currentHref
2823        \global\let\caption@currentHref\@undefined
2824      \fi}%
```

\float@makebox Do our own redefinition of \float@makebox, if it was redefined by the hyperref package.

```
    \@ifundefined{HyOrg@float@makebox}{}{%
      \caption@Debug{%
        Redefining \noexpand\float@makebox (again)\@gobble}%
      \let\caption@ORI@float@makebox\float@makebox % save for compatibility mode
      \renewcommand\float@makebox[1]{%
        \HyOrg@float@makebox{#1\relax \caption@defrostHref}}%
    }%
}}{}
```

2.15.5 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* \captions in the same environment, at least unless a new \capstart will be placed.

```
    \let\caption@ORI@capstart\capstart % save for compatibility mode
    \let\capstart\caption@start
    \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@ifhypcap1%
    \renewcommand*\caption@hypcapspace{\hypcapspace}%
  \fi}{}
```

2.15.6 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
```

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}{}%
```

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` Wrapper macros for typesetting the `caption=` resp. `title=` value.

`\lst@maketitle`

```
\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}%
}{}
```

2.15.7 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$\times$ `\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@resetContinuedFloat\LTcaptype
    \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}}%
}{}
```

2.15.8 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\caption@boxrestore\@parboxrestore
    \let\@makecaption\caption@@make
    \caption@setautoposition b%
    \caption@settype{#1}%
    \caption@clearmargin}%
```

`\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
    \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}}%
}{}
```

2.15.9 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.

```
  \@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@setoptions{parpic}%
    \let\caption@boxrestore\@parboxrestore
    \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}}}
\x
```

```
  \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
}
```

2.15.10 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 `minipage`. (Note that `margin=` is not supported here.)

```
  \long\def\@makerotcaption#1#2{%
    \ifdim\captionwidth=\z@
      \setcaptionwidth{.8\textheight}%
      \caption@slc{#1}{#2}{.8\vsize}{%
        \let\caption@makerot\caption@@make
        \caption@clearmargin
%       \long\def\caption@parbox##1##2{\hbox{\hsize=.8\textheight\relax##2}}%
%         (not needed because \rotatebox uses an \hbox anyway)
        \let\caption@parbox\@secondoftwo}%
      \caption@set@bool\caption@ifslc0% been there, done that
    \fi
    \rotatebox{90}{\caption@makerot{#1}{#2}}%
    \nobreak\hspace{12pt}}%
  \newcommand\caption@makerot[2]{%
    \begin{minipage}\captionwidth\caption@@make{#1}{#2}\end{minipage}}%
  \caption@For{typelist}{%
    \newenvironment{sideways#1}{\@rotfloat{#1}}{\end@rotfloat}%
    \newenvironment{sideways#1*}{\@rotdblfloat{#1}}{\end@rotdblfloat}}%
}{}
```

2.15.11 The sidecap package

```
\caption@IfPackageLoaded{sidecap}[1999/05/11 v1.4d]{%
  \caption@setbool{needfreeze}{1}%
```

\SC@caption First of all, we let sidecap use a current definition of \caption.
(This is only required for version 1.5d of the sidecap package.)

```
  \caption@AtBeginDocument{\let\SC@caption=\caption}%
```

\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 save & restore \caption so \caption@freeze will work correctly.

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

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@settype*{#2}%
    \caption@freeze*}%
```

\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{%
```

Note: \@captype isn't defined here, this will be done inside the original definition of \endSC@FLOAT. But \SC@captype is defined and can be used here, if needed.

```
    \let\caption@ORI@settype\caption@settype
    \def\caption@settype##1{% will be done in \@xfloat
      \caption@ORI@settype*{##1}% do not change \@currentlabel
      \caption@setSC@justify
%%%   \caption@setoptions{SCfloat}%
      \caption@setoptions{SC\@captype}%
      \caption@start}%
```

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
      \@ifundefined{SC@justify}{}{%
```

```
        \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@defrost}%
```

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

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

  \newcommand*\caption@For@SC[2]{%
    \def#1{b}% = \sidecaptionvpos{#2}{b} (v1.6)
    \newenvironment{SC#2}%
      {\SC@float[#1]{#2}}{\endSC@float}%
    \newenvironment{SC#2*}%
      {\SC@dblfloat[#1]{#2}}{\endSC@dblfloat}}%
  \@onlypreamble\caption@For@SC
  \caption@For{typelist}{%
    \expandafter\caption@For@SC\csname SC@#1@vpos\endcsname{#1}}%
}{}
```

2.15.12 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}}

}{}
```

2.15.13 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{%
    \def\@currentlabelname{#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{%
  \def\@currentlabelname{#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}}%
}{}
```

2.15.14 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=}%
}{}
```

2.15.15 The wrapfig package

```
\caption@IfPackageLoaded{wrapfig}{% ver 3.3  (Oct 12, 1999)
```

\float@ifstyle \float@ifstyle{⟨*type*⟩}{⟨*if-clause*⟩}{⟨*else-clause*⟩}
(see float package support for details)

```
\providecommand*\float@ifstyle[1]{%
  \expandafter\ifx\csname fst@#1\endcsname\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi}%
```

\caption@restylewrapfloat This one redefines the wrap#1 environment, e.g. wrapfigure. Our code uses \caption@setoptions{wrapfigure} so \captionsetup[wrapfigure]{...} will work.

But first we check if our redefinition was already done, this could happen inside `\float@restyle` when the wrapfig support of the float package was not installed successfully, so it has not redefined `\wrap#1` there.

```
\newcommand*\caption@restylewrapfloat[1]{%
  \expandafter\ifx\csname caption@OUR@wrap#1\expandafter\endcsname
                  \csname wrap#1\endcsname
    \caption@Error{%
      For a successful cooperation of the 'wrapfig' package\MessageBreak
      with the 'float' package you should load the 'wrapfig'\MessageBreak
      package *after* the 'float' package}%
  \else
    \expandafter\let\csname caption@ORI@wrap#1\expandafter\endcsname
                    \csname wrap#1\endcsname
    \@namedef{wrap#1}{\caption@wrapfloat{#1}}%
    \expandafter\let\csname caption@OUR@wrap#1\expandafter\endcsname
                    \csname wrap#1\endcsname
  \fi}%
```

`\caption@wrapfloat`

```
\newcommand*\caption@wrapfloat[1]{%
  \caption@settype*{#1}%
  \float@ifstyle{#1}{%
    \ifx\WF@floatstyhook\@undefined
      \caption@Error{%
        For a successful cooperation of the 'wrapfig' package\MessageBreak
        with the 'float' package you should use at least\MessageBreak
        'wrapfig' version 3.6}%
    \else
      \float@dostyle{#1}%
    \fi}{}%
  \caption@clearmargin
%%% \caption@setoptions{wrapfloat}%
  \caption@setoptions{wrap#1}%
  \@nameuse{caption@ORI@wrap#1}}%
```

Now we redefine the wrapfig environments we know about.
If someone has placed a `\newfloat` right between `\usepackage{wrapfig}` and `\usepackage{caption}` (or loads the caption package first, so all these patches will be done with `\AtBeginDocument`) we have bad luck since the float package do not offer a list of (re)styled floats. (This would finally lead to an error in `\caption@setfloatcapt`.)

```
\caption@restylewrapfloat{figure}%
\caption@restylewrapfloat{table}%

\caption@For{typelist}{%
  \newenvironment{wrap#1}{\wrapfloat{#1}}{\endwrapfloat}%
  \caption@restylewrapfloat{#1}}%

\ifx\WF@floatstyhook\@undefined \else % wrapfig v3.6
```

`\float@restyle` If the wrapfig package *v3.6* is used, we patch `\float@restyle` (if defined), too, so new or restyled floats will be handled correctly, too.

```
  \@ifundefined{float@restyle}{}{%
    \toks@=\expandafter{\float@restyle{#1}% (env may or may not be defined)
```

```
        \caption@restylewrapfloat{#1}}%
      \edef\@tempa{\def\noexpand\float@restyle##1{\the\toks@}}%
      \@tempa}% perform redefinitions
```

\wrapfloat An additional check of the package load order: If both, neither the wrapfig package nor the caption package haven't catch `\float@restyle`, we finally splash down at `\wrapfloat`.

```
    \let\caption@ORI@wrapfloat\wrapfloat
    \def\wrapfloat#1{%
      \float@ifstyle{#1}{%
        \caption@Error{%
          For a successful cooperation of the 'wrapfig' package\MessageBreak
          with the 'float' package you should load the 'wrapfig'\MessageBreak
          package *right after* the 'float' package}}{}%
      \caption@ORI@wrapfloat{#1}}%

\fi                                     % wrapfig v3.6
```

\WF@rapt We place our hyperref anchor here.
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:

```
\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}%
    \caption@start
  \ifdim\hsize>\z@ \@parboxrestore \else
  \setbox\z@\hbox\bgroup \let\wf@@caption\caption \let\caption\wf@caption
  \ignorespaces \fi}%

}{}
```

References

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

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

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

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

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

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

[7] Olga Lapko:
The floatrow package documentation,
2007/08/24

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[24] Heiko Oberdiek:
The refcount package,
2006/02/20