

Controlling captions, fullpage and doublepage floats: hvfloat

Herbert Voß

Abstract The package hvfloat defines macros which place objects and captions of floats in different positions with different rotating angles for the object and caption. The object can fill a full column, a full page or full doublepage, with or without taking margins into account.

Sommario Il pacchetto hvfloat definisce macro che posizionano oggetti e didascalie di float in posizioni diverse con angoli di rotazione diversi per l'oggetto e la didascalia. L'oggetto può riempire un'intera colonna, una pagina intera o un'intera pagina doppia, con o senza considerare i margini.

1. Introduction

The well-known floating environments like `figure` and `table` are easy to handle if there is only one object and one caption which fits into the current page text layout. If you want a caption rotated and beside the object (an image, tabular, ...) then you need some \LaTeX knowledge or a package which does the rotation and the checking of the current page number if you want to place the rotated caption for a twocolumn document into the outer margin.

All this can be simplified by using the package hvfloat which has a variety of possible options for the floating object and caption. The package is loaded in the usual way:

```
\usepackage[options]{hvfloat}
```

The package has options `hyperref`, `nostfloats`, and `fbox`. The latter is only used for locating spacing problems in the document: objects and captions are framed, so unwanted whitespace can easily be seen. With `nostfloats` one can prevent the loading of the package `stfloats`, which allows bottom floats in a twocolumn document. This option is needed only in rare cases where a package conflict between `stfloats` and another package exists. With `hyperref` the package of that name is loaded.

If you would like to reset the default for the float position parameters to `htp` (here, top and page) (the default is `tbp`, top, bottom, and page), then you can load the helper package `hvfloat-fps`. It knows the optional arguments `table`, `figure`, and `all`. If you have a document with a large number of floats and relatively short text you can load the package with

```
\usepackage[all=!htb]{hvfloat-fps}
```

The exclamation allows \LaTeX to ignore the internal parameter settings for the floats, e.g. the number of floats on one page Mittelbach et al. 2004.

Usually several \LaTeX runs will be needed until hvfloat knows whether figures are on even or odd page and to get all the references correct. The usual warning 'Label(s) may have changed' will be shown if another compilation is needed.

2. Dependencies

The following packages are loaded by default:

afterpage, caption, expl3, graphicx, ifoddpage, multido, picture, stfloats, subcaption, trimclip, xkeyval.

3. The macros and optional arguments

The three main macros are `\hvFloat`, `\hvFloatSet`, and `\hvFloatSetDefaults`. The syntax for calling them is somewhat complex. Optional arguments are gray shaded:

```

\hvFloat * [options] + {float type}{floating object}
           [short caption] {long caption}{label}
\hvFloatSet{key=value list}
\hvFloatSetDefaults

```

The star version of `\hvFloat` is explained in section 4 on page 30 and the optional `+` is explained in section 7.2 on page 41.

The `\hvFloatSet` macro allows the global setting of parameters via the given keyword=value list, while `\hvFloatSetDefaults` sets all parameters to their default values, as shown in Table 2 on page 28.

If `\hvFloat` is given an empty second argument for *float type*, it switches by default to a nonfloat object and activates the option `onlyText` (see Table 2). The *short caption* is a second optional argument; if given, it specifies, as usual, the caption entry for the `\listof`. . . . All other arguments are mandatory but may be empty.

Some other macros are defined, mostly for use in the hvfloat implementation, but they can also be used for a user's own purposes. Only `\tabcaption` should be placed at the top of an object.

```

\figcaption [short caption] {long text}
\tabcaption [short caption] {long text}
\tabcaptionbelow [short caption] {long text}

```

They are used for the nonFloat keyword, where these macros write captions in the same way but outside of any float environment. The default caption cannot be used here. It is no problem to use the `\tabcaption` command to place a caption anywhere, for instance here in an inline mode:

Table 1. A caption with neither sense nor object.

In this case a label should be put inside of the argument and not after the command `\tabcaption`, so that a reference to the nonexistent object Table 1 will still work. Source for this:

It is no problem to use the `\verb|\tabcaption|` command ... here in an inline mode:
`\tabcaption[The caption without sense ...]{A caption with neither sense nor
object.\label{dummy}}`

In this case a label should be put inside the argument ... so that a reference to the nonexistent `Table~\ref{dummy}` will still work.

With the macro `\hvDefFloatStyle` one can define a style to be used instead of the individual setting. Internally the style is saved in a macro named `\hv@name`.

```
\hvDefFloatStyle{name}{setting}
```

The possible keywords are listed in the rotated and full page Table 2 on the next page. To make this table, we first save it in the predefined box `hvOBox` as a `tabularx` with the tabular width of the current `textheight`. A `tabularx` cannot be used as an argument to `\hvFloat`. This is the reason we use the intermediate box:

```
\begin{lrbox}{\hvOBox}\small
  \begin{tabularx}{\textheight}{@{} l>\small\ttfamily}cX @{} \toprule
    \emph Keyword & \emph Default ...
  [...]
  \end{tabularx}
\end{lrbox}
```

Then, to typeset the table, we use the keyword `rotAngle`, which rotates object and caption together:

```
\hvFloat*[floatPos=p,rotAngle=90,capPos=top,capWidth=w,useOBox=true]{table}{}
  {The optional keywords for the \texttt{\textbackslash hvFloat} macro.}
  {tab:options}
```

3.1. Caption positioning

By default the caption is set below the object and the macro `\hvFloat` behaves like the usual `figure` or `table` environment. With the keyword `capPos` and the value `before`, the caption can be placed beside the object. For small objects (smaller than a column/page), `before` is equivalent to `left`. Thus, here is the code for our first example:

```
\hvFloat[capPos=left]{figure}{\includegraphics{frosee}}{A short caption beside a  

figure [...] without a label.}
```

If the caption is shorter than the possible width it is horizontally centered. The vertical position is by default also centered. This can be changed by the optional argument `capVPos`. The formatting can be modified by the optional arguments of the (already-loaded) package `caption`. They can be specified to `\hvFloat` via the optional argument `capFormat` (see Figure 2 on page 29). The caption is also rotated by setting `capAngle=90`, which is a counter-clockwise rotation:

```
\hvFloat[capPos=right, capAngle=90,capWidth=h, capFormat={font=sf}]{figure}
  {\includegraphics{frosee}}{A caption in sans [...], to the right [...], as  

wide as [...], and rotated by 90\textdegree [...]}{fig:1}
```

Table 2. The optional keywords for the \hvFloat macro.

Keyword	Default	Description
floatPos	tbp	This is the same default placement setting as in standard \LaTeX ; maybe not always the best setting.
rotAngle	0	The value for the angle if both the object and the caption should be rotated together.
capWidth	n	The width of the caption. Can be n for a natural width given by the current linewidth, w for the width of the object, h for the height of the object, or a scale factor for $\code{columnwidth}$.
capAngle	0	The integer value for the angle if the caption should be rotated. Positive is counter-clockwise.
capPos	bottom	The position of the caption relative to the object. Possible values: before: <i>always before (left) from the object.</i> left: <i>always before (left) from the object, but on the same page in after: always after (right) from the object.</i> twocolumn mode. bottom: <i>always on the bottom of the object.</i> right: <i>always after (right) from the object, but on the same page in twocolumn mode.</i> outer: <i>in twoside mode always typeset at the outer margin.</i> inner: <i>in twoside mode always typeset at the inner margin.</i> evenPage: <i>in twoside mode with fullpage objects always on an even page.</i> oddPage: <i>in twoside mode with fullpage objects always on an odd page.</i>
capVPos	center	Only used when capPos=left right; in these cases, the caption can be vertically placed at the bottom, center or top.
objectPos	center	Horizontal placement of the object relative to the document. Possible values are (l)eft, (c)enter, (r)ight.
objectAngle	0	Integer value for the angle if the object should be rotated. Positive is counter-clockwise.
floatCapSep	5pt	Additional space between the object and a left- or right-placed caption.
useOBox	false	Instead of passing the object to \hvFloat, with useOBox=true the contents of the predefined box \hvOBox is used.
onlyText	false	The caption is printed as normal text with no entry in any list of ...
nonFloat	false	The object isn't put in a floating environment, but printed as standard text with an additional caption. The float counter is increased as usual and can be referenced.
wide	false	The float can use $\text{\textwidth} + \text{\marginparwidth}$ as horizontal width.
objectFrame	false	Put a frame with no separation around the float object.
style	none	Use a defined style.
capFormat	none	Define formatting options for \caption; see documentation of package caption.
subcapFormat	none	Define formatting options for \subcaption.
fullPage	false	Use a complete column in twocolumn mode.
FullPage	false	Use the full text area for the object.
FULLPAGE	false	Use the full paper width/height for the object.
doublePage	false	Use the text area on a doublepage with additional text.
doublePAGE	false	Use the text area on a doublepage without additional text.
doubleFULLPAGE	false	Use the paperwidth on a doublepage without additional text.
vFill	false	Put a \vfill between every two objects in a multi- or subfloat.
sameHeight	false	use the same text height on both pages for a doublePage object.

Figure 1. A short caption beside a figure object (capPos=left) without a label.

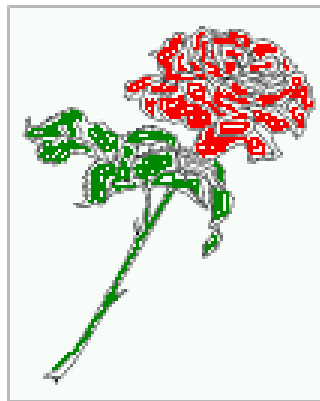
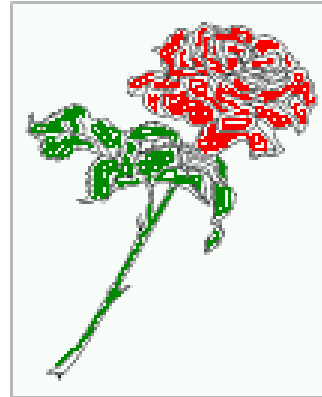
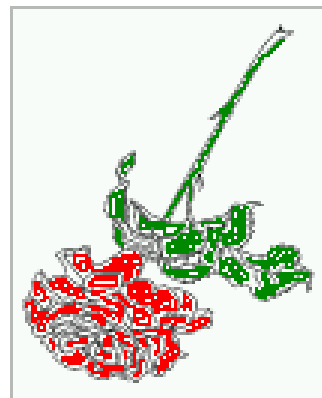


Figure 2. A caption in sans serif (capFormat={font=sf}), to the right of the object (capPos=right), as wide as the object (capWidth=h), and rotated by 90° (capAngle=90).

The caption's vertical position is controlled by the keyword capVPos which accepts the values top, center, and bottom. The capPos=inner setting is explained later (§5.2, p. 33). Typographically, a side caption for images should usually be at the bottom and for a table at the top of the object (Figure 3).

```
\hvFloat[capPos=inner,capVPos=bottom,objectAngle=180]{figure}
{\includegraphics{frose}}{This caption is at the inner margin [...],and
vertically at the bottom [...], and the object is rotated [...]}{fig:11}
```

Figure 3. This caption is at the inner margin (capPos=inner, see p. 33), vertically at the bottom of the object (capVPos=bottom), and the object is rotated 180° (objectAngle=180).

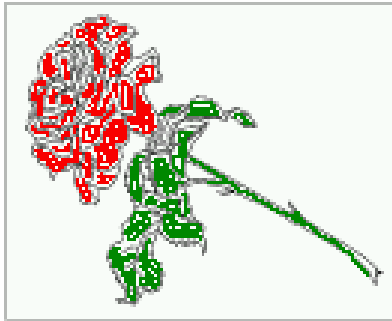


3.2. The caption width

For a caption beside the object the horizontal justification is by default centered if the total width of object and caption are less than the current column/line width. The caption width itself can be controlled by the keyword `capWidth`, which can be set to `n` (natural width), `w` (width of the object), `h` (height of the object), or a value by which to scale `\columnwidth`. Figure 2 on the previous page shows the use of `capWidth=h`, which is used for rotated captions beside the object and Figure 4 shows a caption above the object with the same width.

```
\hvFloat[capWidth=w, capPos=top, capAngle=180, objectAngle=90]{figure}
{\includegraphics{froese}}{A 180°-rotated caption above [...] with the
same width.}{fig:1a}
```

Figure 4. A 180°-rotated caption above a 90°-rotated object, with the same width.



4. The star version `\hvFloat*`

In `twocolumn` mode the floating environment can occupy both columns using the star version `\hvFloat*`. This is analogous to the environments `figure*` and `table*`.



Figure 5. A longer caption to the right of the object (`capPos=right`), and vertically at the bottom of the object (`capVPos=bottom`). It spans both columns (`\hvFloat*`) and may be at the top or bottom of the page.

Table 3. Additional keywords for the `\includegraphics` macro.

<i>name</i>	<i>width=</i>	<i>height=</i>	<i>keepaspectratio=</i>
<code>fullpage</code>	<code>\columnwidth</code>	<code>\textheight</code>	<code>false</code>
<code>FullPage</code>	<code>\textwidth</code>	<code>\textheight</code>	<code>false</code>
<code>FULLPAGE</code>	<code>\paperwidth</code>	<code>\paperheight</code>	<code>false</code>
<code>doublefullpage</code>	<code>2\paperwidth-2in</code> <code>-2\evensidemargin</code>		<code>true</code>
<code>doubleFULLPAGE</code>	<code>2\paperwidth</code>	<code>\paperheight</code>	<code>false</code>
<code>doubleFULLPAGEbindCorr</code>	<code>2\paperwidth-2\bindCorr</code>	<code>\paperheight</code>	<code>false</code>

If possible, the floating environment will be placed at the top of the following page or at the bottom of the current page. The latter needs the package `stfloats` which is loaded by `hvfloat` by default. (`stfloats` cannot place a float at the bottom of the first page of an article or chapter when using the core L^AT_EX document classes; these classes also include code that prevents placement of a float at the top of the first page.) Placing the float across both columns within the text area is not possible. Here is the code for the following example (Figure 5 on the preceding page):

```
\hvFloat*[capVPos=bottom,capPos=right]{figure}
{\includegraphics{froze} \includegraphics[angle=180,origin=c]{froze}}
{A caption to the right [...], It spans both columns [...]}{fig:2}
```

The same can be seen in Table 3, which also spans two columns (we'll discuss the content of that table later). Internally the number of possible floating objects on top of the page is controlled by the parameters `\topnumber` (onecolumn mode) and `\dbltopnumber` (twocolumn mode). They are preset for this documentclass (*ArsT_EXnica*) to 2 and 2 and differ for other document classes. For doublepage objects the values will temporarily be changed to 1.

5. Full column or fullpage objects

As mentioned in Table 2 there are three keywords for fullpage objects:

- `fullpage` for a complete column or page in a onecolumn mode,
- `FullPage` for a complete text area of a page or both columns in a twocolumn mode, and
- `FULLPAGE` for the complete paper area without leaving any margin.

This refers to the reserved space which `\hvFloat` will use when typesetting the object and caption. The object itself can be smaller than a full column or page. Package `hvfloat` defines five additional optional arguments for the package `graphicx` which can be used together with `\includegraphics` to make the code a bit shorter. They are listed in Table 3. The so-called bind correction is additional free space at the inner margins of a twoside document.

In general, the interface is the same whether using the complete text area or the complete paper area for the floating object; the only difference is `fullpage` vs. `FULLPAGE`. By default, such a page will have no page number, no header, and no footer, and the pagestyle is empty.

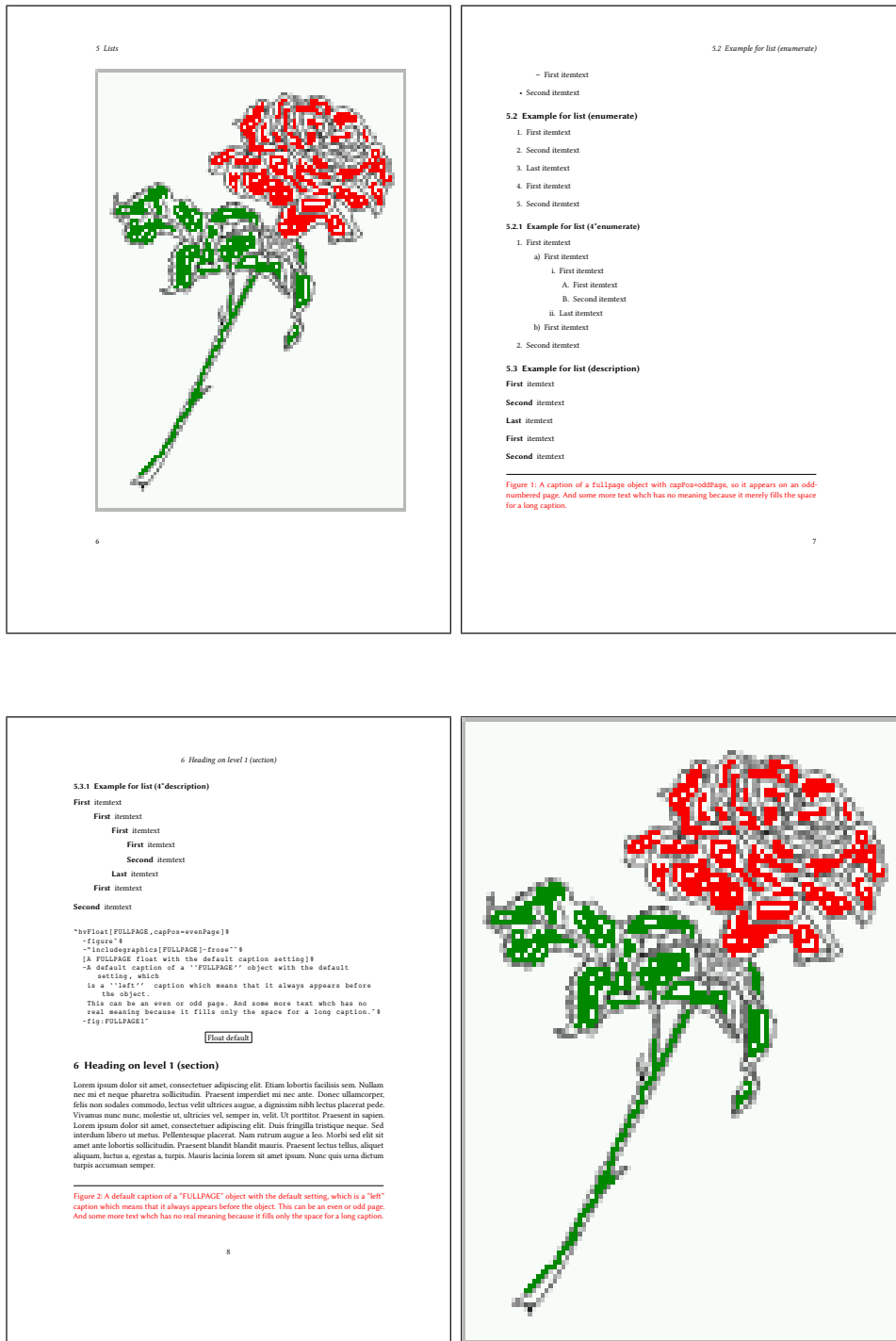


Figure 6. Twoside documents, onecolumn mode.
 Top: a fullpage float and capPos=oddPage (example document odd2s1c.tex, pp. 6–7);
 bottom: capPos=evenPage and a FULLPAGE float (example document paper-even2s1c.tex, pp. 8–9).

The following pages are printed in the `\twocolumn` mode to show that all full column, page, and double page examples also work in this mode.

Setting the keyword `keepaspectratio` to `false` only makes sense for images which have nearly the same ratio as the current column or page height/width. Using a full column or page for an object implies to put the caption on the preceding or following column/page. For a `twocolumn` document this should *always* be the opposite column on the *same* page and for `twoside` documents the opposite page. Only for `doublepage` objects (left–right pages) the caption must be on the preceding or following column/page, by default at the bottom of that page or column.

A label, which is defined by `\hvFloat` always points to the image, not to the caption. This makes no difference for the default floats, where the image and caption are on the same page. For `fullpage` or `doublepage` objects, however, the macro internally defines additional labels; one pointing to the caption (defined by label `\label{label}-cap`) and, if it is a `doublepage` object, another pointing to the second (right) part of the object (defined by label `\label{label}-2`).

All labels, the given one `\label{label}` and the internal ones `\label{label}-cap` and `\label{label}-2`, will point to the same object counter, but possibly to different page numbers. An example is shown in section 6, where Figure 13, defined with label `fig:dP`, has its caption on page 39 and its image on pages 38 and 39. The following table shows the behavior:

	<code>fig:dP</code>	<code>fig:dP-cap</code>	<code>fig:dP-2</code>
<code>\ref{...}</code>	13	13	13
<code>\pageref{...}</code>	38	39	39

5.1. *Twoside and onecolumn mode*

In a `twoside` document with `onecolumn` mode, a `fullpage` object and the corresponding caption should be on facing pages (left–right). This can be specified with the keyword `capPos`

and the values `evenPage` or `oddPage`. To save space we show only the output of two example documents (Figure 6 on the facing page). The upper pair of pages uses the following settings:

```
\hvFloat[fullpage, capPos=evenPage]
{figure}
{\includegraphics[fullpage]{froese}}
{A caption of a \texttt{fullpage}
 object with \texttt{capPos=oddPage}
 ... for a long caption.}
{fig:fullpage1}
```

The lower two pages in Figure 6 are similar, except `capPos=evenPage` and the object is set as `FULLPAGE` instead of `fullpage`.

The captions here (and throughout) are typeset in red to make them more visible in the examples, which are often reduced in size. The complete code for all examples is on CTAN (mirror.ctan.org/macros/latex/contrib/hvfloat/doc/examples).

5.2. *Twoside and twocolumn mode*

In contrast, in a `twoside` document in `twocolumn` mode, by default a caption appears *before* the `fullpage` or `fullcolumn` object, independent of an even or odd column or page. Figure 8 on page 35 shows the output of this example code:

```
\hvFloat[fullpage, capPos=inner]
{figure}
{\includegraphics[fullpage]{froese}}
[A short caption for the LoF.]
{A caption on the inner side of a
 twoside and twocolumn document
 (\texttt{capPos=inner}). This can
 be an even or odd page. And ...
 long caption.}
{fig:full0}
```

The caption is in the inner column, which is the second one for an even (left) page and the first for an odd (right) page. For a `twoside` document it also makes sense to have the caption on the even (left) page in the second

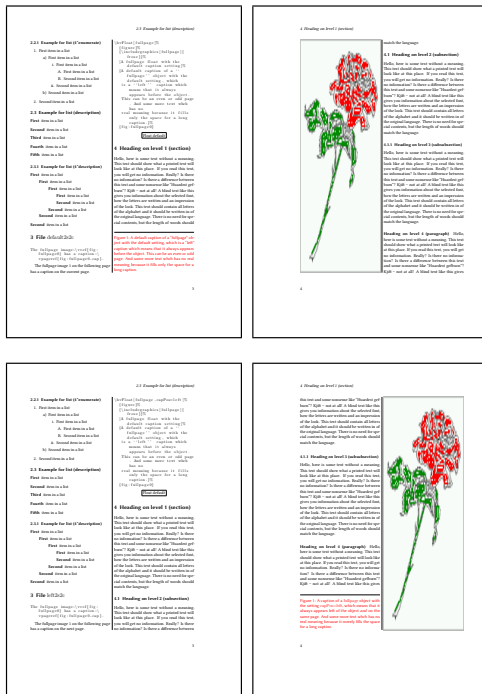


Figure 7. Twoside and twocolumn documents. Top: `capPos=before` (default); bottom: `capPos=left`. Pages 3–4 of example documents `default2s2c.tex` and `left2s2c.tex`, respectively.

margin and the object on the odd page (right) in the first margin. This can be achieved with the setting `capPos=inner`.

You can expect problems if you use the full column setting on a page which has full-width (double column) floats at the top. In such a case it is left to the user to modify the text structure to prevent such situations. You’ll find many examples on CTAN (<https://mirror.ctan.org/macros/latex/contrib/hvfloat/doc/examples>) or in the documentation directory of your T_EX distribution.

In twoside and twocolumn modes the keyword setting `capPos=left` is different from `capPos=before`. For this setting it makes no difference on what page and column the caption appears, it will always be *before* the object. For `capPos=left` the caption will *always*

be left of the object *and* on the same page! Figure 7 shows this behavior.

6. Doublepage objects

A doublepage object makes sense only for twoside documents. Then the doublepage object can be placed on facing left–right pages and the caption perhaps on the right page or, in a case where the complete paper width is used, below the right part of the image, or, if need be, on the bottom of the preceding or following page. For example: suppose a doublepage object uses the complete paper area ($2\backslash\text{paperwidth} \times \backslash\text{paperheight}$) on the (left–right) pages 80–81; then the caption can be printed at the bottom of page 79 or page 82 (see Figure 11 on page 37). It is also possible to print the caption *over* the right part of the object (image) on the bottom or rotated at the right (see Figure 12 on page 37).

With using the keyword `doublePage`, additional document text may appear below the doublepage object, that is, the object does not occupy the entire textheight. The other two possibilities `doublePAGE` (using the doublepage text area) and `doubleFULLPAGE` (use the doublepage paperwidth) have no additional document text on the two pages, but are still floating environments. We’ll now describe all these in detail.

6.1. Keyword `doublePage`

This is the same as putting two different floats, one each at the top of the left and right pages. The package `hvfloat` clips an image which would be wider than the paperwidth. Otherwise it makes no sense to use a doublepage float.

For `doublePage` the object starts at the left top of the text area and ends on the right page, depending on its width. The inner margins of the two-sided document are ignored, but a binding correction (`bindCorr`) can be set and

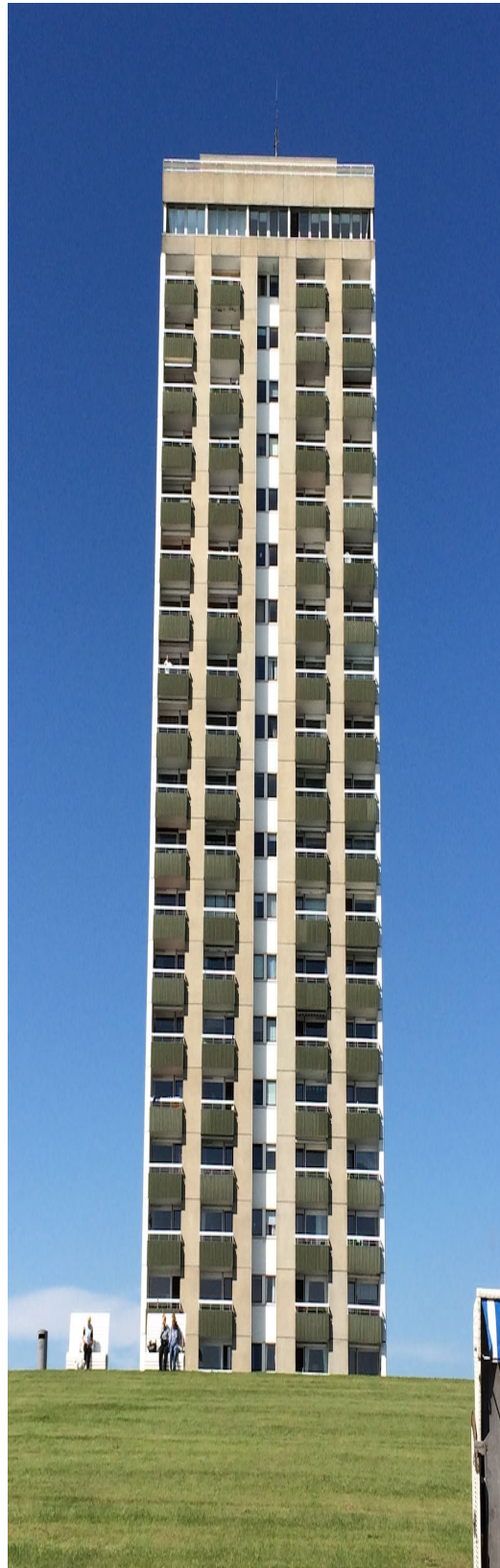
will be taken into account. The caption will *always* be on the right page either beside, rotated or not, or below the object. For example, in Figure 13 on page 38 the caption is on the right (`capPos=right`) and rotated by 90° (`capAngle=90`). The left part of the image is on page 38, the right part on page 39 and the caption is on page 39. Incidentally, the internally-created labels described earlier were used to print this information. The label for the figure is `fig:dP`, and so the source for the previous sentence is:

The left part of the image is on `page~\pageref{fig:dP}`, the right part on `page~\pageref{fig:dP-2}` and the caption is on `page~\pageref{fig:dP-cap}`.

A `doublePage` object allows for document text in addition to the two parts of the object. As for the caption, with `capWidth=n` and `capPos=right` the caption will be set to the right of the object with a natural width (from object to margin). This makes sense if the object is narrower than `\paperwidth + \textwidth`. Figure 13 on page 38 shows this, as well as (at a greatly reduced size) Figure 9 on the next page. The source for Figure 13 is as follows.

```
\hvFloat[doublePage, capWidth=n,
  capPos=right, capVPos=bottom]
{figure}
{\includegraphics[width=2\textwidth]
  {images/seiser}}
[A short caption for the LoF]
{A caption for a \texttt{doublePage}
  object, which will be placed on
  the right side of the right-hand
  part of the image. The image
  begins on the left edge of the
```

Figure 8. A caption on the inner side of a twoside and twocolumn document (`capPos=inner`). This can be an even or odd page. And some more text with no real meaning because it merely fills the space for a long caption.



```
paper [...] The photo was taken
[...]}{fig:dP}
```

In some cases it makes sense to have some whitespace, a binding correction, between the two split parts of the object. With the keyword `bindCorr` you can define a length value for the whitespace to be added both to the right of the left part and to the left of the right part (so the total whitespace added is $2 \times \text{bindCorr}$).

The source for Figure 9 is the same as Figure 13, except for the addition of the setting `bindCorr=1cm` (and the label name).

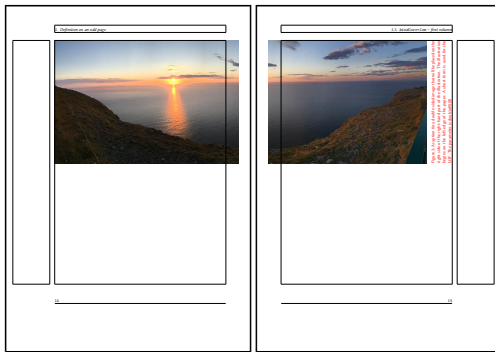


Figure 9. A `doublePage` object (the same image as Figure 13) with a binding correction of 1 cm. Pages 14–15 of example document `doublepage2s2c.tex`.

6.2. Keyword `doublePAGE`

A `doublePAGE` object appears alone on two facing pages, except for an optional caption. No additional document text will be printed on these two pages; this is the only difference between `doublePage` and `doublePAGE`. Figure 10 shows an example. The caption is below the object in the first column of the right (odd) page.

Figure 10 also shows an example of using the optional keyword `bindCorr` to specify whitespace between the parts of the split object. In this case, we use the inner margin for the binding correction to get the two images exactly fitting the textwidth. The value for the inner margin is computed internally:

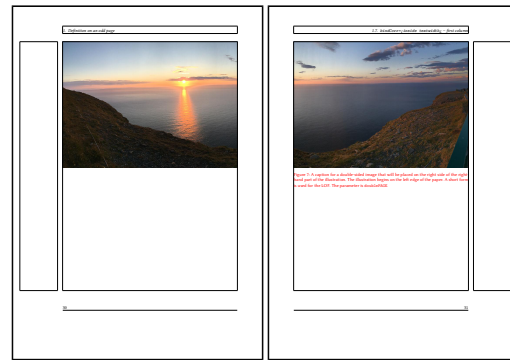


Figure 10. A `doublePAGE` image with `bindCorr` set to the inner margin. Pages 29–30 of example document `doublepage2s2c.tex`.

```
bindCorr=inner
```

Here is the source for Figure 10:

```
\hvFloat[doublePAGE, capWidth=n,
  bindCorr=inner]{figure}
{\includegraphics[width=2\textwidth]
  {images/sonne-meer}}
[A doublepage image with a
caption ...]
{A caption for a double-sided
image ... The parameter is
\texttt{doublePAGE}}
{fig:doublePAGE3}
```

6.3. Keyword `doubleFULLPAGE`

A floating object specified with the keyword `doubleFULLPAGE` always starts in the upper left corner of the left (even) page. The defined text area has no meaning, it will be completely ignored for these two floating pages. The caption can be printed before, after, below, or superimposed on the object.

Table 3 on page 31 lists the corresponding two optional keywords for the command `\includegraphics`, namely `doubleFULLPAGE` and `doubleFULLPAGEbindCorr`, with a preset of `keepaspectratio` to `false`. These keywords may make code more readable but have otherwise no special meaning



Figure 11. A doubleFULLPAGE object with capPos=after, so the caption is on the following page. Pages 80–82 of example document doublefullpage2s2c.tex.



Figure 12. A doubleFULLPAGE object with capPos=right, so the caption appears on the right page. Pages 72–73 of example document doublefullpage2s2c.tex.

the doublepage (left–right) object. For a two-column document there exists the keyword twoColCaption which can be used to span both columns. This will only work for twocolumn documents which define the column mode using \twocolumn, such as the TUGboat document class. The multicol package is *not* supported.

Figure 12 shows two pages with an image spread across the double page which is small enough to get a rotated caption on the right of the page which, for our demonstration, is printed in red as usual. The page layout is also printed as frames, which makes it easier to understand and choose values for the full page mode. These frames are shown by loading the package showframe.

The code for Figure 12 is:

```
\hvFloat[doubleFULLPAGE, capPos=right]
{figure}
{\includegraphics[height=\paperheight]
{images/rheinsberg}}
[A doublepage image ...]
[A caption for a double-sided image ...
The parameter is
\texttt{doubleFULLPAGE}}
{fig:doubleFULLPAGEon}
```

for any objects other than images, e.g. a tabular or something else.

The object can have any width and height but it should be at least as wide as the given \paperwidth and not less than 50% of the \paperheight. For smaller objects, use one of the other two possibilities, doublePage or doublePAGE.

The caption can be superimposed on the object or, as an alternative, printed on the bottom of the page preceding or following



If the image has nearly the same ratio as the current $\text{\paperwidth}/\text{\paperheight}$, then a caption can reasonably appear at the bottom of the following page. This is specified with `capPos=after`; Figure 11 on the previous page shows the result. Similarly, the setting `capPos=before` would put the caption on the preceding page.

Here is the code for Figure 11, specifying the option `doubleFULLPAGE` option to both `\hvFloat` and `\includegraphics`:

```
\hvFloat[doubleFULLPAGE, capPos=after,
  twoColumnCaption]{figure}
  {\includegraphics[doubleFULLPAGE]
    {rheinsberg}}
  {A caption for a double-sided
   image ... The parameter is
   \texttt{doubleFULLPAGE}}
  {fig:doubleFULLPAGE02ndnn}
```

7. Subfloats and multifoats

A floating environment can have any content except another floating environment. The only

requirement for the content is that it must be smaller than one page spread. The content itself can be any combination of text, equations, tabulars, and/or images. We call it a *subfloat* if the content has *one* main caption and several subcaptions for any object. We call it a *multifloat* if the content has no main caption of its own, but the objects have their own captions.

Table 4 on the facing page gives the two keywords, `subFloat` and `multiFloat`, which introduce such special content. They can be placed as a default floating environment, full column, full page, or full doublepage.

The syntax for the macro which defines such sub- or multifoats is somewhat complex. Only the keyword defines whether the float is a multifloat or subfloat; the syntax of the macro shows no difference. With the optional argument `vFill` the objects in a column (two column) or a page (one column) are stretched over the given height \texttheight . The default is no stretching so that extra whitespace appears at the bottom of the column/page.



Figure 13. A caption for a doublePage object, which will be placed on the right side of the right-hand part of the image. The image begins on the left edge of the paper. A short form can be used for the LoF. The photo was taken in the Italian Alps at the Alpe di Siusi (Seiser Alm).

Table 4. Keywords subFloat and multiFloat for multiple objects in a float.

Name	Description
subFloat	For multiple objects with one main caption and several subcaptions.
multiFloat	For multiple objects, each with its own caption.

7.1. Subfloats

A subfloat page can have only one type of object which will have one main caption and individual subcaptions. (For completeness: If you define no subcaption then it does not matter what kind of object we have.) The syntax for subfloats and multifloats is similar, but some arguments are ignored for a subfloat, so can be left empty.

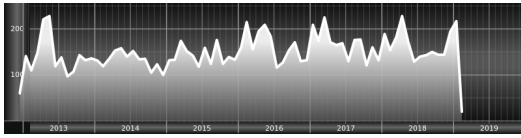
```
\hvFloat [subFloat, ...]
+{main float type}{ } [short caption]
  {long caption}{label}
+{}{sub floating object} [short caption]
  {long caption}{label}
: ...
+{}{sub floating object} [short caption]
  {long caption}{label}
```

The first line defines only the floating type and the main caption, the object entry is ignored! All additional lines will have the same float type; this is why the float type entry is ignored.

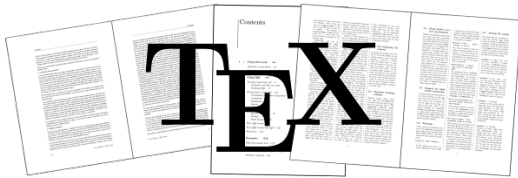
The + symbol defines an additional object which will be part of the same floating environment. It's up to the user to be sure that one page or one column can hold all defined objects.

The code for Figure 14 on the next page, which comprises the subfigures 14a to 14e, is as follows:

```
\hvFloat[subFloat,vFill,fullpage,
```



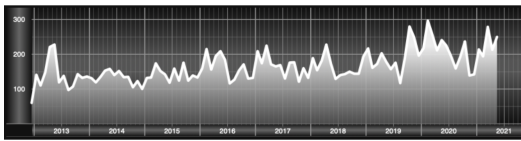
(a) Subcaption A of a fullpage subobject.



(b) Subcaption B of a fullpage subobject, a little longer for no particular reason.



(c) Subcaption C of a fullpage subobject.



(d) Subcaption D of a fullpage subobject.



(e) The last subcaption E of a fullpage subfloat object, which has subcaptions 14a–14e, and the main caption is beside (to the right of) this full column object.

```
capPos=after]
+{figure}[Short caption of the
subfloat]{The main caption of a
fullpage subfloat, which appears
in the left or right column. This
can be an even or odd page. The
\texttt{vFill} option is set,
so vertical space is distributed
between the subobjects.}{sub:demo}
+{\includegraphics[columnWidth]
{CTAN}}[Short caption A]{Subcaption
A of a fullpage subobject.}{sub:demo0}
+{\includegraphics[columnWidth]
{CTAN1}}[Subcaption B of a fullpage
subobject, a little longer for no
particular reason.]{sub:demo1}
+{\includegraphics[columnWidth]
{CTAN2}}[Subcaption C of a fullpage
subobject.]{sub:demo2}
+{\includegraphics[columnWidth]
{CTAN3}}[Subcaption D of a
fullpage subobject.]{sub:demo3}
+{\includegraphics[trim=0 1.5cm
0 5mm,clip,columnWidth]{TUGboat}}
{The last subcaption E of a
fullpage subfloat object, which
has subcaptions \ref{sub:demo0}%
--\ref{sub:demo5}, and the main
caption is beside (to the right
of) this full column object.}
{sub:demo5}
```

The keyword `subFloat` defines the following images or tabulars as subfloats. The keyword `figure` in the second line of the code defines the main type of the floating environment; all subobjects must be of the same type. This is the reason why all following arguments are empty: `+{}{...}`

The package `subcaption` is loaded by default and is usually activated with `\captionsetup[sub][singlelinecheck]`.

Figure 14. The main caption of a fullpage subfloat, which appears in the left or right column. This can be an even or odd page. The `vFill` option is set, so vertical space is distributed between the subobjects.

The main label of the subfloat is `sub:demo`, which points to the object `column` on page 40. In this case the internal label `sub:demo-cap` points to the same page 40, because object and caption are in different columns but on the same page. Both refer to the same object: `\ref{sub:demo} → 14` and `\ref{sub:demo-cap} → 14`.

7.2. Multifloats

With a `multiFloat` object, no main caption is given. Every object gets its own caption, which is the reason that figures, tabulars, etc., can be mixed. All individual captions are listed before or after the full column/page, at the bottom of the column/page (see example on the following page).

```
\hvFloat [multiFloat, ...]
+{float type}{floating object}
  [short caption] {long caption}
  {label}
+{float type}{floating object}
  [short caption] {long caption}
  {label}
: ...
+{float type}{floating object}
  [short caption] {long caption}
  {label}
```

The `+` symbol defines an additional object which will be part of the same floating environment. For a multifloat object all parameters are valid. It's up to the user to be sure that one page or one column can hold all defined objects.

The captions of Figures 15–18 and of Tables 5 and 6 are on page 41, and all objects also appear on the same page. All of these figures and tables are part of the same multifloat. Here is the code of the multifloat example:

```
\captionsetup[singlelinecheck=false]
\hvFloat[multiFloat,vFill,
         fullpage,capPos=before]
```

```
+{figure}
{\includegraphics[columnWidth]{dove}}
[Short caption A]
{Caption A of a fullpage multifloat
 object, which follows in the left or
 right column. This can be an even or
 odd page. And some more text with no
 real meaning because it merely fills
 the space for a long caption.}
{img:demo}
+{table}{\begin{tabular}
 @{}lrcp{2cm}@{} \hline
 Left & Right & Centered & Parbox \\
 \hline
 L & R & C & P \\
 left & right & center & Text
 with line breaks \\
 L & R & C & P \\
 left & right & center & Text
 with line breaks \\
 \multicolumn{4}{c}{Centered
 multicolumn over all columns} \\
 \hline
 \end{tabular}}
[Short example caption B1]
{Caption B of a fullpage object, a
 tabular in this case.}{tab:demo}
+{figure}
{\includegraphics[columnWidth]{CTAN1}}
{Caption C of a fullpage object.}
{img:demo1}
+{figure}
```

Figure 15. Caption A of a fullpage multifloat object, which follows in the left or right column. This can be an even or odd page. And some more text with no real meaning because it merely fills the space for a long caption.

Table 5. Caption B of a fullpage object, a tabular in this case.

Figure 16. Caption C of a fullpage object.

Figure 17. Caption D of a fullpage object.

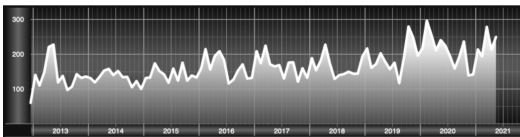
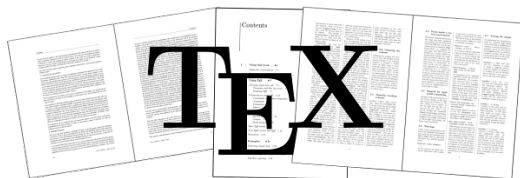
Figure 18. Caption E of a fullpage object.

Table 6. Caption B2 of a fullpage object, another tabular repeating Table 5.



Left	Right	Centered	Parbox
L	R	C	P
left	right	center	Text with possible line breaks
L	R	C	P
left	right	center	Text with possible line breaks

Centered multicolumn over all columns



Left	Right	Centered	Parbox
L	R	C	P
left	right	center	Text with line breaks
L	R	C	P
left	right	center	Text with line breaks

Centered multicolumn over all columns

```

{\includegraphics[columnWidth]{CTAN2}}
{Caption D of a fullpage object.}
{img:demo2}
+{figure}
{\includegraphics[columnWidth]{CTAN3}}
{Caption E of a fullpage object.}
{img:demo3}
+{table}{\begin{tabular}
{@{}lrcp{2cm}@{}}\hline
Left & Right & Centered & Parbox\\
\hline
L & R & C & P\\
left & right & center & Text
with line breaks\\
L & R & C & P\\
left & right & center & Text
with line breaks\\
\multicolumn{4}{c}{Centered
multicolumn over all columns}\hline
\end{tabular}}
[Short example caption B2]
{Caption B2 of a fullpage object,
another tabular repeating
Table~\ref{tab:demo}.}
{tab:demo2}

```

8. Splitting tables across two pages

By default a table can only be split in the vertical direction, as a so-called longtable. Large tables can be rotated on a page (see Table 2 on page 28), but splitting it automatically in the horizontal direction is currently (T_EXLive 2022) not supported by core L^AT_EX.

However, saving a table without page breaks into a box is no problem and such a box can be handled like an image, which is also like a box. The only problem is that the table must be split horizontally between two columns, as a split column may likely be unreadable.

The package hvfloat provides the box \hvOBox for public use. We can save a table into this box:

```

\savebox\hvOBox{%
\begin{tabular}{l @{} *{18}r}
... the table ...

```

`\end{tabular}}`

and then use it in the same way as a double-page image, with the table split in two pieces. If the split occurs at an unfavorable point in the table, e.g. in the middle of a column, then insert some horizontal space between the two columns with `@{\hspace{...}}`. For example (the output is shown in Table 7):

lume 0 (9999), No. 0				TU/Gboat, Volume 0 (9					
1981	1982	1983	1984	1985	1986	1987	1988	1989	1990
0	0	0	20	0	2	2	2	1	
0	2	1	3	4	4	6	4	2	
0	0	1	5	3	1	7	7	3	
2	6	0	1	0	3	7	2	1	
1	2	0	5	2	2	5	4	2	
0	0	1	1	0	2	5	4	3	
3	2	1	2	1	3	5	3	4	
0	0	0	4	2	1	4	5	2	
1	1	0	1	1	1	4	4	1	
0	0	2	6	1	0	2	1	1	
0	0	0	0	0	0	1	0	3	
0	1	0	2	0	0	2	2	2	
0	0	0	2	0	1	3	0	2	
0	0	0	3	3	2	1	1	0	
1	0	0	4	0	0	3	1	1	
0	0	0	0	0	3	5	0	1	
6	3	5	23	10	8	15	13	1	

Figure 19. The table column 1985 appears between the two pages and would not be readable.

```
\begin{tabular}
  {\lll@{\hspace{1cm}}ll}\hline
  1 & 2 & 3 & 4 & 5 \\
  1 & 2 & 3 & 4 & 5 \\
  1 & 2 & 3 & 4 & 5 \\
  1 & 2 & 3 & 4 & 5 \\
\end{tabular}
```

Figure 19 shows how the table looks in the middle of the doublepage (the text shown at the bottom of the page is just filler). The column with 1985 will be cut and not readable. There are two solutions to split the table

Table 7. Adding space between two columns

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

at a better position: insert some space *before* this column, or use the `bindCorr` keyword to insert a binding correction space. For Table 8 on the following page both possibilities are used. Inserting more space:

```
\begin{tabular}
  {l @{} *{13}r @{\quad}*8r}
```

and using 8 mm for the binding correction (shown below) which was found by trial and error.

The code for the split table on a double page is:

```
\hvFloat[doublePage,capWidth=n,
  capPos=right,capVPos=bottom,
  useOBox,% use the defined box
  bindCorr=8mm]
{table}
{}% no need for an object
[A doublepage tabular.]
{A caption for a doublePage tabular that
will be placed on the right side of the
right-hand part of the tabular. The
table begins on the left edge of the
text area of the left page. The
additional space between the columns
1984 and 1985 is \texttt{\textbackslash
quad}, which is the same as 1\,em. The
binding correction is set to 8\,mm,
which gives additional whitespace of
16\,mm.}{tab:dP}
```

and the output is Table 8 on pages 44 and 45. It depends on the way the document is printed whether more or less space between the two pages makes sense.

	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982
Line No 1	1	3	1	1	1	0	1	1	0	0	0
Line No 2	1	1	3	1	0	0	0	0	0	0	2
Line No 3	2	1	2	1	0	0	0	0	0	0	0
Line No 4	1	0	5	1	2	0	0	0	0	2	6
Line No 6	2	1	1	0	0	0	0	0	0	1	2
Line No 5	0	0	4	2	1	2	2	1	0	0	0
Line No 8	0	1	1	0	0	0	1	1	0	3	2
Line No 9	0	0	0	0	0	1	2	1	0	0	0
Line No10	0	1	3	0	1	0	1	0	0	1	1
Line No11	0	2	2	1	1	0	1	0	0	0	0
Line No12	2	0	2	4	1	0	4	0	0	0	0
xyz	2	3	0	0	0	0	0	0	0	0	1
Line No13	0	1	0	0	1	0	3	0	0	0	0
Line No14	0	1	0	0	0	0	0	0	0	0	0
Line No15	0	0	0	0	0	0	0	0	0	1	0
Line No16	0	0	0	0	0	1	0	0	0	0	0
Some numbers	2	6	13	8	4	3	5	4	0	6	3

9. Todo list

The macro `\hvFloat` only checks the position of its definition if it is defined on an odd or even page. This is done with the help of the macro `\checkoddpage` from the package `ifoddpage`. Together with the internal \LaTeX macro `\if@firstcolumn` it knows exactly the position of its definition in the source of the document: left or right page, first or second column. But it doesn't know if the current page is completely empty, which is the case if `\hvFloat` is the first command on a new page. If this is also an even page, then a `doublepage` object can be placed immediately. But the current code always uses the *next* even-odd page combination. In a future release there should be a test like `\if@newpage`.

More checks for the correct use of the parameters would be useful. For example: if one uses the keyword `doubleFULLPAGE` with an object which is narrower than `\textwidth`, then the output will be rubbish.

The optional argument `wide` as shown in Figure 21 on page 46 works only in `oneside` mode if you also use `twocolumn` mode (see Figure 20 on the facing page). For `twoside` mode we have different margins for a possible wide float in the first or the second column; this is not recognized by `hvfloat`. However, if you need wide floats in a `twoside` and `twocolumn` mode you can move the macro `\hvFloat` to places in the source where the output is always in the outer column, which uses the `marginpar` width. Using the argument `nonFloat`, as shown in Figure 20 on the next

1983	1984	1985	1986	1987	1988	1989	1990	1991	1992
0	20	0	2	2	2	1	2	1	0
1	3	4	4	6	4	2	2	1	0
1	5	3	1	7	7	3	2	1	0
0	1	0	3	7	2	1	2	1	0
0	5	2	2	5	4	2	2	1	0
1	1	0	2	5	4	3	2	1	0
1	2	1	3	5	3	4	2	1	0
0	4	2	1	4	5	2	2	1	0
0	1	1	1	4	4	1	2	1	0
2	6	1	0	2	1	1	2	6	0
0	0	0	0	1	0	3	2	6	0
0	2	0	0	2	2	2	2	6	0
0	2	0	1	3	0	2	2	6	0
0	3	3	2	1	1	0	2	6	0
0	4	0	0	3	1	1	2	6	0
0	0	0	3	5	0	1	2	6	1
5	23	10	8	15	13	1	32	51	1

Table 8. A caption for a doublePage tabular that will be placed on the right side of the right-hand part of the tabular. The table begins on the left edge of the text area of the left page. The additional space between the columns 1984 and 1985 is `\quad`, which is the same as 1 em. The binding correction is set to 8 mm, which gives additional whitespace of 16 mm.

page, the float appears exactly at the place of the definition.

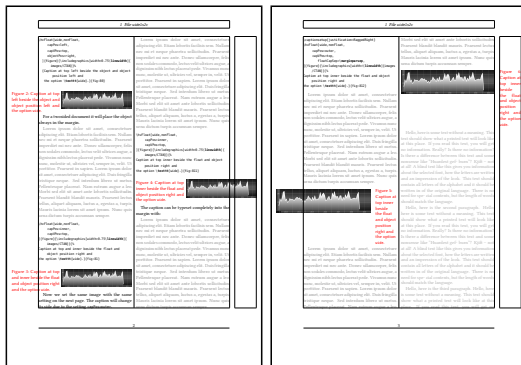


Figure 20. Pages 2–3 of example document `wide1s2c.tex`, onside with twocolumn and the wide option.

In some cases the option `useOBox` for a predefined savebox `\hvOBox` does not work. One can use instead `{\usebox\hvOBox}` as the argument for the object, which has the same effect. However, the box `\hvOBox` must have valid contents, and be set before it is used.

10. Conclusion

The package `hvfloat` should work with all kinds of documents, onside in one- or twocolumn mode, twoside in one- or twocolumn mode. It is much easier to place doublepage objects in a onecolumn document than a twocolumn document. Internally, \LaTeX puts two single pages together to one page with two columns. Only the optional header and footer are printed across these “two” pages.

The package `hvfloat` makes intensive use of the macro `\afterpage` Carlisle and The \LaTeX Team 2014. If one defines a doublepage object in the first column of a left (even) page, `\hvFloat` needs *three* nested `\afterpage` commands, one for each column, to let an object or a caption start on the next left (even) page. Until \LaTeX reaches this page for the object/caption, nearly two pages have to be filled with text or other objects which are defined after the macro `\hvFloat`. Especially in two-column mode you can expect problems, if you have too little text, images, tables or other simple objects to fill up these two pages until the doublepage object will be set. Such problems can only be solved by adding some text or moving the macro `\hvFloat` to another column of the document.

Just as with the standard floating environments `figure` and `table`, it is left to the user to ensure that the contents of the environment (object and caption) fit the page. If an object is wider than $2 \times \text{\paperwidth}$ or higher than `\paperheight` it cannot be placed on a doublepage and the output may be useless.

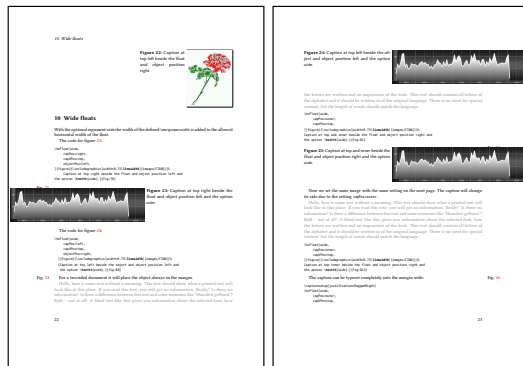


Figure 21. Pages 22–23 of the package documentation, showing examples using optional argument `wide` to use the margin space.

Before using a doublepage for an object, one should test if it might be sufficient to use the margin for additional space. `\hvFloat` knows the optional argument `wide` which al-

lows using the space of `\marginparwidth`. The caption can be placed in the usual way, above/below or left/right relative to the object. The use of the inner/outer position for twoside documents is also possible.

Figure 21 shows some examples of using the margin for a onecolumn document with the following use of `\hvFloat`.

```
\hvFloat[wide, capPos=inner, capVPos=top]
{figure}
{\includegraphics[width=0.75\linewidth]
{images/CTAN}}
{Caption at top inner beside the float ...
and the option \texttt{wide}.}{fig:wide}
```

The list of figures and list of tables are not affected by package `hvfloat` and should work as usual. For example, here is the list of tables for this article:

Another feature is that simple and non-floating objects can be placed by the environment `hvFloatEnv`, which has only one optional argument, giving the horizontal width. For the caption one has to use the defined macro `\captionof{type}{...}` or the (usually internal) macro `\tabcaption{...}` mentioned on page 26:

List of Tables

1	A caption with no object . . .	2
2	The optional keywords for the <code>\hvFloat</code> macro	4
3	Additional keywords for the <code>\includegraphics</code> macro	7
4	Keywords <code>subFloat</code> and <code>multiFloat</code> for multiple objects in a float	15
5	Short example caption B1	17
6	Short example caption B2	17
7	Adding space between two columns	19
8	A doublepage tabular	21
9	A short nonfloating table	23

```

\begin{hvFloatEnv}[0.5\columnwidth]
\centering
\captionof{table}{A short nonfloating table.}\label{tab:nonfloat}
\begin{tabular}{@{} l c r @{}}\hline
left & center & right \\
L & C & R \\
\end{tabular}
\end{hvFloatEnv}

```

Table 9. A short nonfloating table.

left	center	right
L	C	R

But pay attention to references if floating and non-floating environments are mixed on one page; they can point to wrong numbers. Moving the floating environment to another place in the document is one workaround for such a problem. Alternatively, using only floating environments is preferred, if your document is mainly text, with only some figures and/or tables.

References

- Carlisle, David, and The \LaTeX Team. 2014. “The `afterpage` package.” Execute command after the next page break, October 28, 2014. Accessed May 28, 2021. <https://ctan.org/pkg/afterpage>.
- Mittelbach, F., M. Goossens, J. Braams, D. Carlisle, and C. Rowley. 2004. *The \LaTeX Companion*. 2nd. Pearson Education. ISBN: 9780133387643.
- Scharrer, Martin. 2016. “The `ifoddpaper` package.” Determine if the current page is odd or even, April 23, 2016. Accessed May 29, 2021. <https://ctan.org/pkg/ifoddpaper>.
- . 2020. “The `adjustbox` package.” Graphics package-alike macros for “general” boxes, August 19, 2020. Accessed May 29, 2021. <https://ctan.org/pkg/adjustbox>.
- Sommerfeldt, Axel. 2020. “The `caption` package.” Customising captions in floating environments, October 26, 2020. Accessed May 28, 2021. <https://ctan.org/pkg/caption>.
- Tolušis, Sigitas. 2017. “The `stfloats` package.” Commands to control the presentation of floats, March 27, 2017. Accessed May 28, 2021. <https://ctan.org/pkg/stfloats>.
- Voß, Herbert. 2021. “The `hvfloat` package.” Rotating and controlling float captions and objects, June 29, 2021. Accessed June 29, 2021. <https://ctan.org/pkg/hvfloat>.

Herbert Voß
WASGENSTRASSE 21
14129 BERLIN, GERMANY
Herbert.Voss@fu-berlin.de