The floatrow package*

Olga Lapko
Lapko.O@g23.relcom.ru

2008/03/28

Abstract

This package was created as extension of the float package. The floatrow package borrows core code from the float[1] and rotfloat[2] packages, so you *must not* load these packages.

The float package has a good mechanism for the creation (and easy modification) of common layout for all floats of one type without adding any repeated code in the document; besides, this package allows to create new float types; it deals only with alone (plain) combinations "object (float contents)—caption".

The rotfloat package changes environments of rotated floats (the `sideways...` environment of rotating package) to adapt them to float's settings.

The package floatrow extends these possibilities and, at last, it allows:

- to use mechanism, borrowed from float package, for creation of new float types;
- to change width of float box, either to a fixed value or to the width of object;
- to put caption beside object;
- to put a few floats side by side on the row;
- to put footnotes inside float box (using `minipage`-like mode); and also to put legend-like text;
- to create and/or modify special layout for each type of float and for different positioning of float and its components, e.g. two-column or rotated float.

The floatrow package is cooperated with caption package (needs version 3.0**q** or later, *the better* cooperation will be with version **3.1*x***). Also the floatrow package (like caption one) uses keyval package mechanism for layout settings.

I do my best to follow this idea and I hope that someone likes it: helps to maintain this idea in any way, or finds bugs and absurdities in this package or documentation.

*This file has version number v0.2c, last revised 2008/03/28.

[1] float package, version v1.3d dated 2001/11/08, © 1991–2000 Anselm Lingnau.

[2] rotfloat package, version v1.2 dated 2004/01/04, © 1995–2004 Axel Sommerfeldt.

Document Terminology

float (float box) could include *object*, *caption*, and *foot material*; *float* is created by `figure` or `table` environments (*plain float*), or by `\floatbox` command and its modifications (*float box*);

float type means standard environment `figure` or `table`, also their *layout subtypes*, like e.g. `wrapfigure` (wrapfig package), `sidewaysfigure` (rotating and rotfloat packages), `longtable` (longtable package) etc.;

object means `tabular` or graphics, as contents of table (`table`) or figure (`figure`) or other type of float;

caption means text in `\caption`;

foot material could include explications, legends and/or footnotes inside *float box* (`\footnote`/`\mpfootnotemark`/`\footnotetext`, and `\floatfoot` macros).

Frequently Appeared Design

Contents

List of Figures

List of Tables

List of Examples

List of Programs

1 Introduction

During creation of document, you usually type figures and tables as *floating objects* (*floats*), i.e. put their contents inside `figure` and `table` environments consequently. The simplest floating environment looks like:

```
\begin{⟨float type⟩}
⟨float contents (object)⟩
\caption{⟨caption contents⟩}
\end{⟨float type⟩}
```

or (if you want to put caption above):

```
\begin{⟨float type⟩}
\caption{⟨caption contents⟩}
⟨float contents (object)⟩
\end{⟨float type⟩}
```

1.1 Loading The Package

Just now you have loaded the floatrow package:

```
⟨preamble⟩
  \usepackage{floatrow}  .
⟨preamble⟩
```

In the time, when this package was loaded, all float contents in the document will be centered (unless another alignment command appears inside the float contents). All captions appear below float contents, regardless of how they were typed in source file. But, I'm almost sure, that you want to put table captions above table material. If you put in the next line the `\floatsetup` command:

```
⟨preamble⟩
  \usepackage{floatrow}
  \floatsetup[table]{style=plaintop}  ,
⟨preamble⟩
```

after that, again, you will get all table captions above table material, regardless of how they were typed in source file. These first minimal settings will arrange all floats contents and their captions accordingly to the real typographic rules. (The section 3 describes and demonstrates various layouts, which you can get with the settings of `\floatsetup` command.)

But surely the settings above are still not sufficient to you, because you need to get the table caption width equal to the width of table material. Also you may want to put some figure captions beside graphics. Besides that, it is better to put small floats beside in one row. For all these reasons this package offers special commands for building of float boxes and a special environment to put these float boxes beside each other.

1.1.1 Float Box Commands

One of the first macros of this package for creation of float boxes is a macro which builds contents of the table environment with caption above (`\ttabbox`). The width of caption

Caption above table object

equals to the width of contents, e.g. of tabular (see table 1). (The first example uses plain LaTeX layout—the caption and floatrow packages loaded without package setting options; the options at the end of \usepackage command define dates of package versions which support correct work of this tandem today.)

```
⟨preamble⟩
  \usepackage{caption}[2007/04/11]
  \usepackage{floatrow}[2007/08/24]
⟨preamble⟩

\begin{table}
\ttabbox
  {\caption{A small table ...}\label{...}}
  {\begin{tabular}...\end{tabular}}
\end{table}
```

Table 1: A small table with caption text above (\ttabbox) with plain LaTeX layout

First column	Second column	Third column
A	B	C
D	E	F

Another command which creates figures—\ffigbox (figure 1)—puts caption below contents. The default width of caption equals to the width of text. (In the following example the most popular layout settings for captions were added.)

```
⟨preamble⟩
  \usepackage[font=small,labelfont=bf,labelsep=period,
        justification=centerlast]{caption}
  \usepackage{floatrow}
⟨preamble⟩

\begin{figure}
\ffigbox
  {\caption{A simple figure ...}\label{...}}
  {...}
\end{figure}
```

Figure 1. A plain figure box with long long long long long long long long long long long long long long long long long multilined caption

Caption's width equals to object

The example above shows that a float box, created by the `\ffigbox` command looks similar to the plain `figure` environment. But if you set, for example, the option `[\FBwidth]` like below:

```
...
\begin{figure}
\ffigbox[\FBwidth]
  {\caption{A figure}\label{...}}
  {...}
\end{figure}
```

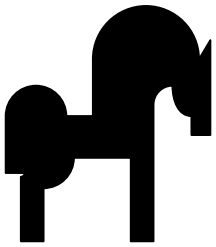


Figure 2. A figure with the width equal to graphics with long long long long long multilined caption

you'll get a caption width equal to the width of picture (figure 2).

The third macro—`\fcapside` (figure 3)—puts caption beside. (In the next example the float layout settings were added, which put captions to the binding margin and changed value of separation space between caption and object to `\quad`.)

⟨*preamble*⟩
```
  ...
  \usepackage[capbesideposition=inside,
      facing=yes,capbesidesep=quad]{floatrow}
```
⟨*preamble*⟩

```
\begin{figure}
\fcapside
  {\caption{...}\label{...}}
  {...}
\end{figure}
```

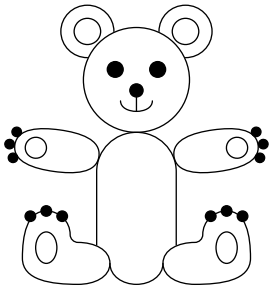


Figure 3. Beside caption (width of caption equals to the width of object) and more text and some more text and a bit more text and a little more text and a little piece of text to fill space

The width of text, by default, divided into two columns, their width equals to the half text width (figure 3) float margins and horizontal space (or width of the separation material) between float and caption are taken into account. The one column is occupied by the object, the other by the caption and foot material (explications or legends and footnotes).

The width of object box equals to object

If you set the `[\FBwidth]` option:

```
...
\fcapside[\FBwidth]
...
```

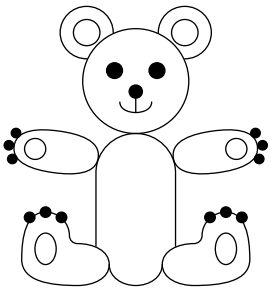


Figure 4. Beside caption (the caption text occupies the rest space beside float object) and more text and some more text and a bit more text and a little more text and a little piece of text to fill space

the graphic box width will be equal to the width of the graphics and the caption will occupy the rest space (see figure 4).

The examples above show the most frequent and most simple variants of float creation. Read section 2.1 about usage of these commands in different ways and how to create new commands for float creation.

1.1.2 Float Boxes In The Row

Floats of one type side by side

If you need to put two or more floats of one type side by side, you may use the `floatrow` environment.

```
⟨preamble⟩
  \DeclareCaptionLabelFormat{rightline}{\rightline
          {\bothIfFirst{#1}{ }#2}}
  \captionsetup[table]{labelformat=rightline,labelsep=newline,
        labelfont={md,sl},textfont=bf}
  \usepackage[font=small,floatrowsep=qquad,captionskip=5pt]{floatrow}
  \floatsetup[table]{style=Plaintop}
⟨preamble⟩

\begin{table}
 \begin{floatrow}
  \ttabbox
    {\caption{...}\label{...}}
    {...}

  \ttabbox
    {\caption{...}\label{...}}
    {...}
```

```
 \end{floatrow}
\end{table}
```

Table 2

Beside table I with long long long long long long and top aligned caption

Left Column Head	Data	
	I	II
First row	1	2
Second row	3	4
Third row	6	8
Fourth row	10	16

Table 3

Beside table II with top aligned caption

Column Head	Data		
	I	II	III
First row	1	2	1
Second row	3	4	6
Third row	6	8	28

As you see in the example with tables 2 and 3, you *need* to use commands `\ttabbox`, which build box for each table.

In the example with beside floats the special settings for table captions were applied (see caption package documentation). Float layout: The value of the separation space between beside floats have been changed to `\qquad`, the vertical skip between captions and float objects was changed to 5pt. For the tables the style `Plaintop` was used which not only puts captions above, but also aligns them by top line (see section 3 of current documentation).

1.2 Do Not Write That With floatrow Package

The floatrow package offers many features, and it causing some limitations for writing code of float contents in source file, too. If you'll write something like

```
⟨preamble⟩
  \usepackage{floatrow}
⟨preamble⟩

\begin{table}\captionsetup{position=top}
  \caption{A table caption must be placed above, ...}
  \centering \begin{tabular}{cc} A & B  \\ C & D \end{tabular}
\end{table}
```

please do not expect that the caption appears at the top of table:

A	B
C	D

Table 4

A table caption must be placed above, wrong expect

So if you want to put table captions above its contents 1) change code, using command `\ttabbox`, like in table 1; 2) write `\floatsetup[table]{style=plaintop}` in the preamble (section 3); or 3) restore the standard LaTeX behavior with the `\RawFloats` command or the package option `rawfloats` (section 2.4).

The next example. If you put beside floats by following way:

```
...
\begin{figure}
\begin{minipage}{0.45\textwidth}
  \centering ...
  \caption{The figure caption, disappeared, ...}
\end{minipage}\hfill
\begin{minipage}{0.45\textwidth}
  \captionof{table}{The table caption ...}}
  \centering ...
\end{minipage}
\end{figure}
```

you'll get error message about lost caption. Here you may: 1) to put table contents inside `\ttabbox` resp. the figure contents inside `\ffigbox`; then both floats put inside `floatrow` environment, and, since there is mixed row (it includes floats of different types, and also with different caption position), put the `\killfloatstyle` command before "foreign" float `\ttabbox`, and `\CenterFloatBoxes` command before `floatrow` environment (see section 2.3.1 about mixed rows); or 2) to restore the standard LaTeX behavior, using command `\RawFloats` or package option `rawfloats` (section 2.4).

2 Macros for Building Floats

2.1 The \floatbox Macro

\floatbox

The examples in Introduction (section 1.1.1) use three commands \ttabbox, \ffigbox and \fcapside. All these commands were built using the \floatbox macro. This macro creates the float box with defined positioning of its elements (object, caption, foot material) and applies the layout of current float type. The usage of the \floatbox macro looks like:

\floatbox[⟨*preamble*⟩]{⟨*captype*⟩}[⟨*width*⟩][⟨*height*⟩][⟨*vert pos*⟩]
{⟨*caption*⟩}{⟨*object*⟩}

The \floatbox's arguments:

⟨***preamble***⟩ there could be \capbeside command which places caption beside float contents; \nocapbeside (to put caption above/below, accordingly to float type's style); \captop (to put caption above); or another systematic command (even with usage of \captionsetup and \thisfloatsetup, see examples in documentation and appendix).

⟨***captype***⟩ the type of float this command is created for. Since this command is supposed to appear outside floating environments or in "foreign" environments (see section 2.3.1 below), we write here, usually, the *actual* name of float type;

⟨***width***⟩ the width of object—caption box (in case of caption above or below object), or width of object box (if caption stays beside object). The empty width option, [], and option [\hsize] mean the same;

⟨***height***⟩ the height of object—caption box (in case of caption above or below object), or height of object box (if caption stays beside object). With the empty height option, [], is used the natural height of object;

⟨***vert pos***⟩ vertical alignment of object contents in object's box in case of the ⟨*height*⟩ argument differs from the natural value of object height, or in the float layout there are used settings for common (max) height for float objects inside floatrow environment. Arguments are analogous to minipage's ones:

- t aligns objects by top line;
- c aligns objects by center line;
- b aligns objects by bottom line;
- s stretches objects by full height (if it is possible).

⟨***caption***⟩ text of caption; you may also use the \footnote/\mpfootnotemark/\footnotetext stuff for footnotes inside float, and/or \floatfoot command;

⟨***object***⟩ contents of float; you may also use the \footnote/\mpfootnotemark/\footnotetext stuff and/or \floatfoot command.

Note. The order of the two last mandatory arguments, ⟨*caption*⟩ and ⟨*object*⟩, and their contents makes no difference during building of float box. The \floatbox macro historically needs two mandatory arguments, but they could be filled freely, i.e. you may fill only one mandatory argument with object contents, caption etc. and left another one empty.

2.1.1 Float Box Width Equals to The Width of Object Contents

\FBwidth
Caption's width equals to object

The [\FBwidth] option in the ⟨*width*⟩ argument allows usage of natural width of float contents: 1) for full float box in the case of caption above/below; 2) in the case of caption beside float object, the natural width of float object expands to the object box only.

Note. If you use the \FBwidth command in the optional argument ⟨*width*⟩, please get sure that object contents can be placed in \hbox command. (You only allowed to use \vspace (not \vskip!) command at the very beginning and very end of object contents for fine tuning of vertical spaces and position of contents.)

\FBheight

The similar command, [\FBheight], was created for the ⟨*height*⟩ argument. The usage of this command makes sense, e.g., when calc is loaded: you may define height option like [\FBheight+1cm].

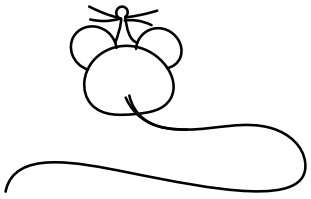
2.1.2 Complex Example of Usage of \floatbox Command

The next example shows figure environment with beside caption. In this example the ⟨*preamble*⟩ argument consists of rather complex definition. The ⟨*width*⟩ option includes the \FBwidth command, so the object box has its natural width, the width of caption box equals to 4cm, and all lines in caption justified, but the last one flushed to the right.

```
\begin{figure}
⟨preamble⟩
  \newcommand\rightlast{\leftskip0ptplus1fil
    \rightskip0ptplus-1fil\parfillskip0ptplus1fil}
  \DeclareCaptionJustification{rightlast}{\rightlast}
⟨preamble⟩

\begin{figure}
\floatbox[{\capbeside
     \captionsetup[capbesidefigure]{labelsep=newline,
          justification=rightlast}%
     \thisfloatsetup{capbesideposition={left,center},
      capbesidewidth=4cm}}]{figure}[\FBwidth]
  {\caption{...}\label{...}}
  {...}
\end{figure}
```

Figure 5
Beside caption and some more text and a bit more text and a little more text to fill space



Please note that complex preamble options, which contain more than one command, must be placed inside curly braces. (See section 3 about settings for floats with \floatsetup.)

2.2 Creation of Personal Commands for Float Boxes

The usage of \floatbox command with options (which could be cumbersome) is sometimes rather complex. The Introduction demonstrates the three already defined

commands-abbreviations of this command. You may define commands-abbreviations (or redefine existing) for your own purposes and include some additional style definitions and settings there.

\newfloatcommand
\renewfloatcommand

The definition of new float abbreviation looks like:

\newfloatcommand{⟨*command*⟩}{⟨*captype*⟩}[⟨*preamble*⟩][⟨*default width*⟩]

where:

⟨***command***⟩ the user's command name (without backslash);

⟨***captype***⟩ the name of floating environment this command is created for;

⟨***preamble***⟩ you may use commands, mentioned in page 15 and other layout commands, like was shown in examples; you may try to add any other regular command (e.g. \captionsetup or \thisfloatsetup stuff);

⟨***default width***⟩ the main purpose of this optional argument is setting it to \FBwidth, which is already included in definition of \ttabbox—the command for building tables. You may also use any dimensions like 6cm or \textwidth here.

For example you may define command for figure 5 like following:

```
\newfloatcommand{fcapbesideleft}[{\capbeside
     \captionsetup[capbesidefigure]{labelsep=newline,
          justification=rightlast}%
     \thisfloatsetup{capbesideposition={left,center},
      capbesidewidth=4cm}}][\FBwidth]
```

2.2.1 Usage of Personal Float Box Commands

Your defined commands can be used in the following way (example for \ffigbox):

\ffigbox[⟨*width*⟩][⟨*height*⟩][⟨*vert pos*⟩]{⟨*caption*⟩}{⟨*object*⟩}

where the options are:

⟨***width***⟩ the width of object—caption box (in case of caption above or below object), or width of object box (if caption stays beside object). The empty width option, [], and option [\hsize] mean the same. The [\FBwidth] option sets natural object width;

⟨***height***⟩ the height of object—caption box (in case of caption above or below object), or height of object box (if caption stays beside object). The [\FBheight] option sets natural object height. With the empty height option, [], is used the natural height of object;

⟨***vert pos***⟩ vertical alignment of object contents in object's box in the case of ⟨*height*⟩ argument has a different value than natural height of object contents, or in the float layout there are used settings for common (max) heights of float elements (object or/and caption) inside floatrow environment. Arguments are analogous to minipage's: t, c, b, s (see above).

See examples with usage of all options on the page 92 and in Appendix.

2.2.2 Predefined Float Box Commands

Let's repeat three already defined commands-abbreviations, defined in package:

```
\newfloatcommand{ffigbox}{figure}[\nocapbeside]
\newfloatcommand{fcapside}{figure}[\capbeside]
\newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

You may see that these commands-abbreviations are equivalent to the following code:

`\ttabbox` —`\floatbox[\captop]{table}[\FBwidth]`;
`\ffigbox` —`\floatbox{figure}` (simplest definition); and
`\fcapside` —`\floatbox[\capbeside]{figure}`.

The first two are defined for figures, and the third one for tables. You may redefine existing macros using `\renewfloatcommand` command (it uses the same arguments as `\newfloatcommand` one).

Note. In the documentation text below the name of the `\floatbox` command means both itself and all commands-abbreviations, defined with `\(re)newfloatcommand`.

Some explanation. The strange "stammering" names of float boxes, with doubled first letters, `\ffigbox` and `\ttabbox` were created, because of the expected names, `\figbox` and `\tabbox`, are already used by the floatflt package, which creates figures and tables which do not span the full width of a page and are filled around by text (i.e. *wrapped* floats, see section 7.4). Also there were founded `\figbox` in formlett and `\tabbox` in automata package among styles in LaTeX folder.

2.3 Building Float Row

The `floatrow` environment allows to put two or more floats beside. The usage of it looks like:

```
\begin{floatrow}[⟨number of beside floats⟩]
\floatbox...
\floatbox...
...
\end{floatrow}
```

Please note that *for each float box* inside `floatrow` you must use `\floatbox`, `\ffigbox`, `\ttabbox` or your own command, created with `\newfloatcommand` macro.

The `floatrow` environment creates necessary number of "columns", the default number is two, where floats are placed (during the calculation of width of column the widths of the separations between beside floats and margins around the float row are taken into account). You may redefine the width of each float box, e.g. the boxes of tables 2 and 3 (page 13) have the width of their contents (remember, the `[\FBwidth]` is default option of `\ttabbox`).

During building each float box inside float row, the `floatrow` environment calculates the rest space in the row and writes this value at the special parameter `\Xhsize`, which you may use inside ⟨*width*⟩ option of `\floatbox` command. The next exam-

ple with figures uses `[\FBwidth]` command in option for the left float, and `[\Xhsize]` command—for the right.

Float occupies the rest space in the row

```
...
\begin{figure}
 \begin{floatrow}
  \ffigbox[\FBwidth]
    {...}{\caption{...}\label{...}}
  \ffigbox[\Xhsize]
    {...}{\caption{...}\label{...}}
 \end{floatrow}
\end{figure}
```

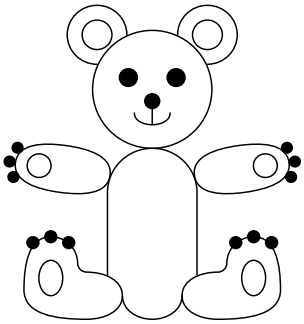


Figure 6. Left beside figure, the width of graphic

Figure 7. Beside figure at the right side of simple figure row, the box width occupies the rest space of row

Usually the command `\Xhsize` is used for the last float box to occupy the rest space of the row. But if you use calc package you may try to use `\Xhsize` earlier, if the *absolute* value of the width of float boxes to the right in float row is known. Another variant: you may set something in ⟨*width*⟩ argument something like `\Xhsize/2` and then `\Xhsize` for two last float boxes—the next example just uses it: the first float has default width equal to "column" width, the next uses width of included graphic (uses command `\FBwidth` in optional argument ⟨*width*⟩), the last two floats divide the rest horizontal space of page into two equal pieces which were calculated by command `\Xhsize` and calc package.

```
⟨preamble⟩
  \usepackage{calc}
  \makeatletter\@mparswitchfalse\makeatother
  \DeclareMarginSet{hangleft}{\setfloatmargins
      {\hskip-\marginparwidth\hskip-\marginparsep}{\hfil}}
  \floatsetup[widefigure]{margins=hangleft}
⟨preamble⟩

\begin{figure*}
\begin{floatrow}[4]
```

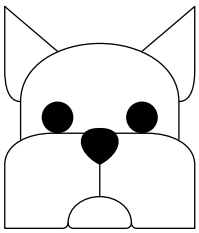


Figure 8. Figure I in the row (`floatrow`), "column" width

Figure 9. Figure II in the row (`floatrow`), graphics width

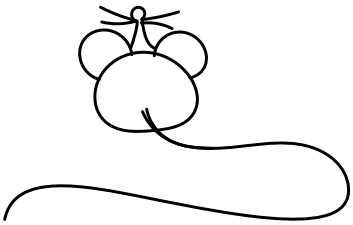


Figure 10. Figure III in the row, float's width box has the half of the rest space of row

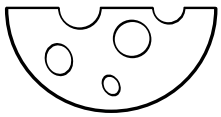


Figure 11. Figure IV in the row, occupies the rest space of row

```
  \ffigbox
    {\caption{Beside figure~I...}...}{...}
  \ffigbox[\FBwidth]
    {\caption{Beside figure~II...}...}{...}
  \ffigbox[\Xhsize/2]
    {\caption{Beside figure~III...}...}{...}
  \ffigbox[\Xhsize]
    {\caption{Beside figure~IV...}...}{...}
\end{floatrow}
\end{figure*}
```

The result you see in the row of figures 8–11. Please note that in the examples with rows, the vertical alignment of floats lays on the bottom of upper part (here: objects) of float and the top of lower part (captions).

The current example uses the starred `figure*` environment, which demonstrates here the possibility of creation and usage of the alternative layout for the float type (here for the figure). It sets the special margin settings, which allow to expand to the left margin (see page 43 about margins settings in `\floatsetup` command). The first command in this example, between `\makeatletter` and `\makeatother` commands, switch of facing margins in twoside document: margins on all pages appear on the left side (like in current document).

2.3.1 Mixed Row

Problems. 1) Sometimes, for example, it is necessary to put beside figure and table. The problem of such mixed row is that you must put different types of float in one floating environment, which sets its own layout for included float box(es).

2) Another problem is that figures usually have captions below graphics, but tables could have caption *above* their contents. The alignment of all floats is similar: the bottom of upper part and top of lower part. In this case if you want to put such beside figure and table you'll get an undesirable result.

`\killfloatstyle` **Solutions**. 1) For creation of right layouts for each float type in mixed row, you ought to write `\killfloatstyle` command just before each "foreign" (for current floating environment) `\floatbox` macro.

\CenterFloatBoxes
\TopFloatBoxes
\BottomFloatBoxes

2) For correct vertical alignment of different float types, which put captions in different positions, you may use one of the following commands:

```
\CenterFloatBoxes
\TopFloatBoxes
\BottomFloatBoxes
```

which align *full* float boxes by center, top or bottom lines. There is also \PlainFloatBoxes which restores standard behavior of \floatbox'es.

\buildFBBOX

These macros were created by \buildFBBOX macro, which can be written like

\buildFBBOX{⟨*starting code of the box*⟩}{⟨*finishing code of the box*⟩}

just before any \floatbox command (or floatrow environment). For example, definition of \CenterFloatBoxes looks almost like following:

```
\newcommand\CenterFloatBoxes{%
  \buildFBBOX{\hbox\bgroup$\vcenter\bgroup\vskip0pt}%
             {\vskip0pt\egroup$\egroup}}
```

The other two commands use \vtop and \vbox boxes consequently. (see also example with usage of \buildFBBOX command on the page 87).

In the next example we use \CenterFloatBoxes command before floatrow and \killfloatstyle just before \ttabbox macro (mixed float row with figure 12 in Boxed style, and table 5):

```
⟨preamble⟩
  \floatsetup[figure]{style=Boxed}
⟨preamble⟩

\begin{figure}\CenterFloatBoxes
\begin{floatrow}
  \ffigbox[\FBwidth]
     ...
  \killfloatstyle\ttabbox
     ...
```

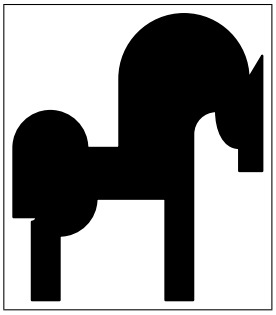


Figure 12. A Boxed figure in the mixed row

Table 5
A table in the mixed row

A	B
C	D

Note. Both figure and table boxes have got width equal to contents of objects: the \ffigbox command in the example has optional argument [\FBwidth], but \ttabbox does not have any option—it uses [\FBwidth] option as default (see definitions on page 18).

2.4 Running Floats in the Raw LaTeX Mode

The floatrow package redefines floating environments for the case of creation of common layout for all floats. This redefinition creates some limitations for source document file, which were mentioned in introduction (see section 1.2). If you still need a raw behavior of floating environment, you may do that by one of the following three ways.

\RawFloats 1) If you want LaTeX behavior *just for one environment*, input a \RawFloats command *inside* environment:

```
⟨preamble⟩
  \floatsetup[figure]{style=Boxed}% please note, it does nothing here
⟨preamble⟩
\begin{figure}\RawFloats
\captionsetup[table]{position=top}
\begin{minipage}{0.45\textwidth}
  \centering ...
  \caption{...}\label{...}
\end{minipage}
\begin{minipage}{0.45\textwidth}
  \captionof{table}{...}\label{...}
  \centering ...
\end{minipage}
\end{figure}
```

And you'll get figure 13 and table 6.

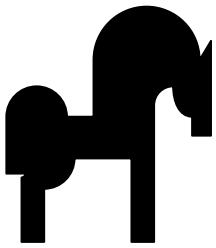


Figure 13. A figure in raw LaTeX's mode

Table 6

A beside table in raw LaTeX's mode

A C	B D

Compare this example with example in the section 2.3.1 and the following figure 14 and table 7.

```
⟨preamble⟩
  \floatsetup[figure]{style=Boxed}
⟨preamble⟩

\begin{figure}\RawFloats\CenterFloatBoxes
\begin{floatrow}
  \ffigbox[\FBwidth]
   {...}
   {\caption{...}\label{...}}
```

```
  \ttabbox
   {...}
   {\caption{...}\label{...}}
\end{floatrow}
\end{figure}
```

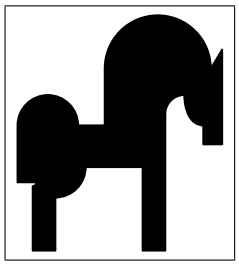


Figure 14. A figure in `\ffigbox` and inside `floatrow` in raw LaTeX's mode

Table 7
A table in `\ttabbox` and inside `floatrow` in raw LaTeX's mode

A	B
C	D

2) Canceling of floatrow's behavior for *all floats of one type or subtype* should be done outside any floating environment, usually in the preamble of the document. In this case the `\RawFloats` command needs optional argument with name(s) of float type. You may set that by two ways:

`\RawFloats[`⟨*type,type,...*⟩`]` or
`\RawFloats[`⟨*type*⟩`][`⟨*subtype,subtype,...*⟩`]`

So if you set `\RawFloats[figure]`, that will return the plain LaTeX mode to all figures in all subtype environments (`figure`, `figure*`, `sidewaysfigure`, `wrapfigure`, etc., see page 28). If there is also a table, `\RawFloats[figure,table]`, you also will set the same for all table subtypes.

The second way, with second optional argument, cancels floatrow's behavior for mentioned float "subtype(s)" of *one* float type in second optional argument you may use `float`, `widefloat`, `rotfloat`, `widerotfloat`—the meaning of this options analogous to options of `\floatsetup` macro (see section 3, but you may use here only options which include "float" word).

`rawfloats` 3) This option stores the plain LaTeX mode (i.e. stores usage of standard LaTeX float macros) for all *standard and new defined* float types. This option can be used only in `\usepackage` line.

Notes.

1) Please note that with `\RawFloats[...]` command and `rawfloats=` key you will cancel layout (`\floatsetup`) settings of all chosen float types/subtypes (section 3) for plain floats.

2) The `floatrow` environment (section 2.3) and commands of `\floatbox` stuff (section 2.1) still work after `\RawFloats[...]` command and `rawfloats=` key (see example with figure 14 and table 7). Also note that a) the layout settings of the package, written in `\usepackage` line and inside `\floatsetup{...}` command, and settings for

main types of floats like \floatsetup[figure]{...} or \floatsetup[table]{...} still can work inside \floatbox commands; b) for the figures inside \fcapside command and similar ones (with the \capbeside command inside the \floatbox's ⟨*preamble*⟩ option)—the settings \floatsetup[capbesidefloat]{...} and \floatsetup[capbesidefigure]{...} or \floatsetup[capbesidetable]{...} work; c) inside the floatrow environment—the settings \floatsetup [floatrow]{...} and \floatsetup[figurerow]{...} or \floatsetup [tablerow]{...} are added to the settings for \floatbox'es inside; d) also you may use \thisfloatsetup settings in the case of usage of \floatbox commands.

The settings for all other layout subtypes (see section 3) will be canceled.

2.4.1 Raw Caption—Printing in Unusual Way

\RawCaption This command allows to "release" caption contents from special box register created by floatrow package for the creation of necessary layout. The caption is placed as argument of \RawCaption:

```
\RawCaption{\caption\marg{contents}\label{...}}  .
```

In this case the settings of float layout of current type will be stored, but you may put caption in non-standard way. For example in the free corner of the graphics (figure 15):

```
⟨preamble⟩
  \floatsetup[figure]{style=plain}
⟨preamble⟩
\begin{figure}
\framebox(70,60){...}\hspace{2\unitlength}%
\framebox(70,60){...}\vspace{2\unitlength}\par
\framebox(70,60){...}\hspace{2\unitlength}%
\parbox[b][60\unitlength]{70\unitlength}%
          {\RawCaption{\caption{...}\label{...}}}
\end{figure}
```

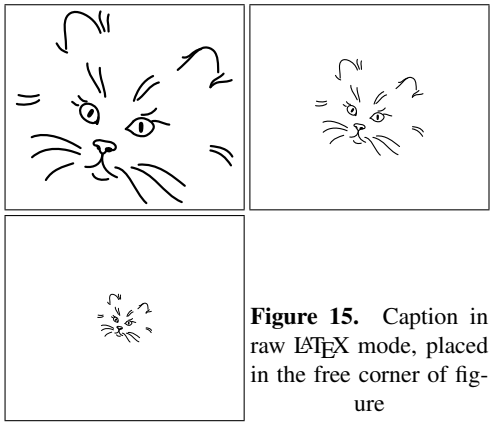


Figure 15. Caption in raw LaTeX mode, placed in the free corner of figure

The more suitable example of usage of the \RawCaption command see on the page 74 (figure 63 with modified BOXED style).

2.5 Usage of Footnotes Inside Float Environment

Sometimes table or figure contents have material, which authors mark and then write some explanation like footnotes. This package has a mechanism which allows to put footnotes inside floating environments, in the same way as is in LaTeX's `minipage` environment.

In the case of few elements have the same footnote, we cannot use standard `\footnotemark`—`\footnotetext` combination, because `\footnotemark` in standard LaTeX always creates the sign of main text footnote. For these cases current package offers `\mpfootnotemark` macro instead of `\footnotemark`. (The same macro also is defined in footmisc package. The floatrow package doubles this definition.)

Footnotemark inside float

```
⟨preamble⟩
  \floatsetup[table]{...,footnoterule=none,footskip=.35\skip\footins,...}
⟨preamble⟩
\begin{table}
\ttabbox
 {\caption{...}\label{...}}%
 {\begin{tabular}{...}
... & 2\mpfootnotemark[1] \\
...
  \end{tabular}%
  \footnotetext[1]{Even numbers.}}
\end{table}
```

Table 8

Table with footnote

Column head	Data I	Data II
First row	1	2[a]
Second row	6[a]	4[a]
Third row	28[a]	8[a]

[a] Even numbers.

The `\floatbox` macro uses special definition of footnote rule (the `footnoterule=` key, see also page 51 for variants of footnote rule) and skip before footnotes and explications or legends (the `footskip=` key).

2.6 The Legend-Like Macro

In the case of table or figure have some additional explanations which could not put in caption contents and they are definitely not a footnote you may use the `\floatfoot` command. The `\floatfoot` is build by usage of `\caption` stuff and uses by default caption's text justification:

```
\begin{table}
\ttabbox
  {\caption{...}\label{...}}
  {\begin{tabular}{...}
  ...\end{tabular}%
```

```
  \floatfoot{‘Data I’ column ...}}
\end{table}
```

Table 9
Table with foot material (e.g. legend)

Column head	Data I	Data II
First row	1	2
Second row	6	4
Third row	28	8

‘Data I’ column—numbers which equal to sum of all their divisors; ‘Data II’ column—2^n values

The star form (`\floatfoot*`) prints its contents as plain unindented paragraph (see table 10).

```
  ...\end{tabular}%
  \floatfoot*{‘Data I’ column ...}}
\end{table}
```

Table 10
Table with foot material (e.g. legend) printed as unindented paragraph

Column head	Data I	Data II
First row	1	2
Second row	6	4
Third row	28	8

‘Data I’ column—numbers which equal to sum of all their divisors; ‘Data II’ column—2^n values

For defining of explication font use `footfont=` option in `\floatsetup` (page 34). You may try to define special settings for float foot using `\captionsetup[floatfoot]` (see section 3).

Notes. 1) The float package defines additional optional argument after main caption text, possibly for explications. Since this possibility didn’t declared in user part of documentation the current version of caption (3.0 and later), and also floatrow package, doesn’t support this possibility. You may use `\floatfoot` and `\footnote`/`\mpfootnotemark`/`\footnotetext` stuff instead.

2) If you use both commands `\floatfoot` and `\footnote` inside one float box, the `\floatfoot` appears above `\footnote` contents.

3) Foot material (footnotes and text in floatfoot) can be placed in several variants: at the very bottom of float box, below caption (even if caption is above float object; see description of `footposition=` key on the page 39 and sample file `frsample01.tex`). In case of caption beside float object, footnotes and foot text are always placed below caption.

2.7 Fine Tuning of Vertical Spaces of Float

At the final variant of document you may need to correct vertical spaces between float and main text, between float object and caption.

To change space between float box and main text, you may use two simple commands `\FBaskip` and `\FBbskip`. For example define

```
\renewcommand\FBaskip{-4pt}
\begin{figure}
  ...
\end{figure}
```

to move up float box up (or reduce space above) by 4pt. Or write

```
\renewcommand\FBbskip{-5pt}
\begin{figure}[t]
  ...
\end{figure}
```

to reduce space below (here: distance between figure and main text) by 5pt. In current document the `\FBaskip` command was necessary for moving up some of wrapped figures.

Use `\vspace` command for vertical space correction around float object[1].

Note. If you'll write something like:

```
⟨preamble⟩
  \usepackage{floatrow}
⟨preamble⟩

\begin{figure}
  ...
\caption{...}
\vspace{-6pt}
\end{figure}
```

in *plain* floats like in example above, you will change space between caption and object (in the case of caption below object). Again, for layout with caption above:

```
⟨preamble⟩
  \usepackage[capposition=top]{floatrow}
⟨preamble⟩

\begin{figure}
\vspace{-6pt}
\caption{...}
  ...
\end{figure}
```

you will get the reduced space between caption above and object contents.

[1] The plain floating environment allows usage of `\vskip` command. But `\floatbox` stuff (`\floatbox` itself, `\ffigbox` etc.) in case of usage of the `\FBwidth` option, gets error message when `\vskip` appears.

3 Float Layout Settings

The idea of floatrow package is to avoid a lot of repeated code for creation of desired layout for floats inside the document text. If you ought to change the layout of one float type or even of all float types, the package allows also to make these modifications of layout much easier. In this case you only have to care about the *markup* of floats and their contents.

The easy modification of common layout of all float types or only for one float type is possible because of the borrowed code from the float package, which allows to modify layout of floats of one type as a whole.

The common layouts and modification for captions for all float types as a whole, for each float type separately, and other special settings are supported by caption package, version 3.*x*.

The layout settings of floatrow package are built similarly to the settings from the caption 3.*x* package. So the layout settings of the `\floatsetup`[1] command are built in similar way as layout settings of the `\captionsetup` command[2].

You may use the layout settings as floatrow option in the `\usepackage` line in the preamble of codument.

```
⟨preamble⟩
  \usepackage[⟨options⟩]{floatrow}  .
⟨preamble⟩
```

You may write

```
⟨preamble⟩
  \usepackage[style=boxed,font=small]{floatrow}  .
⟨preamble⟩
```

`\floatsetup` The same result you get with the `\floatsetup` command:

```
⟨preamble⟩
  \usepackage{floatrow}
  \floatsetup{style=boxed,font=small}  .
⟨preamble⟩
```

The lines above declare the `boxed` float style (this style creates the frame around float object which is built by LaTeX's `\fbox` command) and the `\small` font for contents of float objects. These settings are loaded for *all* float types.

The usage of the `\floatsetup` command has following form:

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

where option ⟨*float type*⟩ is the name of float type. You can use this optional argument for creating of special settings of chosen float type. The following command

```
\floatsetup[table]{style=Plaintop}
```

[1] Some key and option names were changed from version 0.1d, the reason was to arrange and make names more memorable, and, sometimes, reduction of their names (see section 11.3).

[2] Look also at the caption documentation (version 3.0 and later)

sets a special float style for floating tables: captions are placed above float objects; in the case of floats are placed in one row, inside the `floatrow` environment, text of captions is aligned by the top lines.

The `[table]` or the `[figure]` options are not the only options you are allowed to use. The `\floatsetup` command allows usage of a number of special options for settings for floats in different positioning: plain floats, two-column floats (in one-column layout of the document, the starred environment like `figure*` can be used for alternative float layout, e.g. for wide floats, which expand to the margins) rotated floats, wrapped floats. There is also minor support for floats with captions placed beside float objects.

Below are lists of all possible options of the `\floatsetup` command. They are based, as example, on the `figure` environment. The "strength" of options in the lists below decreases from the previous item to the next one.

- Wide or two-column floats (`figure*`):
  - `\floatsetup[widefigure]`—the "strongest" settings; if they are absent, the settings from the next item will be used;
  - `\floatsetup[widefloat]`—these settings "stronger" than settings from next item (`\floatsetup[figure]`); if they are absent, the settings from the next item will be used;
  - `\floatsetup[figure]`; if they are absent, package uses settings from optional argument in `\usepackage` line or `\floatsetup{...}` command; if they are absent—the default package settings will be used (see page 53);
- Wrapped floats (`wrapfigure`, used with wrapfig package):
  - `\floatsetup[wrapfigure]`;
  - `\floatsetup[wrapfloat]`;
  - `\floatsetup[figure]`;
- Rotated floats (`sidewaysfigure`, used with rotating package):
  - `\floatsetup[rotfigure]`;
  - `\floatsetup[rotfloat]`;
  - `\floatsetup[figure]`;
- Wide or two-column rotated floats (`sidewaysfigure*`):
  - `\floatsetup[widerotfigure]`;
  - `\floatsetup[widerotfloat]`;
  - `\floatsetup[rotfigure]`;
  - `\floatsetup[rotfloat]`;
  - `\floatsetup[figure]`;

  Note. The settings for wide float (`widefloat`, `widefigure`) are skipped for rotated floats—use settings for `widerotfloat` and—here—`widerotfigure`;

- Beside floats:
  - `\floatsetup[floatrow]`;
  - `\floatsetup[figurerow]`;
  - settings of outer environment from previous items, e.g., `sidewaysfigure*`, `sidewaysfigure`, `figure*` and `figure`.
- Floats with beside captions (please note, that settings in these options are limited, see next section):
  - `\floatsetup[capbesidefigure]`;
  - `\floatsetup[capbesidefloat]`;
  - settings for the float row; settings of outer environment from previous items, e.g., `sidewaysfigure*`, `sidewaysfigure`, `figure*` and `figure`.

Notes.

1) You can also create and change special settings for captions of necessary float types or subtypes, using co-named ⟨*float type*⟩ options inside the `\captionsetup` command, e.g., `\captionsetup[widefigure]{...}`.

2) Please note that with `\RawFloats[...]` command and `rawfloats=` key (section 2.4) you will cancel all layout settings created as options in the `\usepackage` line or inside the `\floatsetup` command for all chosen float types/subtypes.

3) The `floatrow` environment (section 2.3) and `\floatbox` commands (e.g. `\ffigbox`, `\ttabbox`, see section 2.1) still work after both `\RawFloats` (`\RawFloats[...]`) command and `rawfloats=` key (see example width figure 14 and table 7). Also note that a) inside `\floatbox` commands still can work layout settings of the package, written in `\usepackage` line and inside `\floatsetup{...}` command, and settings for main types of floats like `\floatsetup[figure]{...}` or `\floatsetup[table]{...}`; b) for the figures inside `\fcapside` command and similar ones (with the `\capbeside` command inside the `\floatbox`'s ⟨*preamble*⟩ option) the settings `\floatsetup [capbesidefloat]{...}` and `\floatsetup[capbesidefigure]{...}` or `\floatsetup[capbesidetable]{...}` work; c) inside the `floatrow` environment the settings `\floatsetup[floatrow]{...}` and `\floatsetup[figurerow]{...}` or `\floatsetup[tablerow]{...}` are added to the settings for `\floatbox`'es inside; d) also you may use `\thisfloatsetup` settings in the case of usage of `\floatbox` commands.

The settings for all other layout subtypes (see section 3) will be canceled.

The next few sections describe keys of `\floatsetup` macro.

3.1 Floatsetup Keys

3.1.1 Float Style

`style` The *float style* could include settings of the justification (in particular) of float contents; margins (in particular the alignment of float boxes); separation material between objects and captions and between float boxes in a row (mainly spaces); frames or lines and other options.

The *float style* is specified by following way:

`style=`⟨*float style name*⟩ ,	the name of the ⟨*float style name*⟩ option you may take from table 11.
`...`	You may create your own options with the `\DeclareFloatStyle` command, see page 53.

As you may see in the table 11, the floatrow package includes all float styles which emulate co-named ones from the float package.

Please note, that usage of `style=` key for floats with beside captions, i.e. using `\floatsetup` settings with options like, e.g., `[capbesidefigure]` or `[capbesidefloat]` can destroy layout for this float subtype. For example that key cancels settings for beside position of caption. If you really need to create the alternative layout for floats with beside captions, for example to print float objects in frames, using the `Boxed` style: 1) if you are creating one-column document, revise your settings which were used for float creation, maybe you didn't use the settings for starred floating environments, like `figure*`, so you can load necessary settings for floats with beside caption inside `\floatsetup[widefigure]{...}`, and then use `figure*` environment for floats with beside captions; 2) if you can't follow advice of the previous item, you may use a bit risky variant with usage of `\killfloatstyle` command, see section 3.4.

The caption package uses its own settings and names for caption layout styles. The caption's `ruled` style is the only one from float package, which was predefined in caption package. (The `ruled` style is used by the floatrow package as well as other float package's styles.) To use caption settings of the `ruled` style, you may write

```
\captionsetup[figure]{style=ruled}
```
.

Table 11

Float layout styles

Style	\floatsetup keys	Description
Offered by floatrow package		
`plain`[1][2][3]	⟨*none*⟩	The style `plain` is standard LaTeX's layout. Puts captions always below float object's contents.
`plaintop`[1]	`capposition=top`	The style `plaintop` is the same as `plain` style, but puts captions above float object's contents—this style is analog to the co-named style from the float package.
`Plaintop`	`capposition=TOP`	Capitalized form, `Plaintop`, aligns captions of the floats, which were placed in one row (in the `floatrow` environment), by top line (see example on the page 13).

Continued on next page

Table 11 (*Continued*)

Style	\floatsetup keys	Description
ruled[13]	capposition=top, precode=thickrule, midcode=rule, postcode=lowrule, heightadjust=all	The first style, ruled, emulates co-named style from the float package. It places thick rule above float box, and thin rules between caption and object and below float. Rules are separated from contents by small 2pt skip (see example on the page 41).
Ruled[2]	style=ruled, capposition=TOP	Capitalized form, Ruled, aligns captions of the floats, which were placed beside in one row (in the floatrow environment), by top line (see example on the page 41).
boxed[1234]	captionskip=2pt, framestyle=fbox, heightadjust=object, framearound=object	The first style, boxed, emulates co-named style from the float package. The *width of object* equals to the width of main text (usually \textwidth), predefined \hsize, or the width in \floatbox's option; frame climbs out to the right and left sides (see example on the page 47). Frame separation and rule width equal to current \fboxsep and \fboxrule settings. (Default values are 3pt and .4pt consequently.)
Boxed[23]	style=boxed, framefit=yes	In capitalized form, Boxed, *the width of frame* around object fits the width of main text (usually \textwidth), predefined \hsize, or the width in \floatbox's option; the width of object is reduced to fit inside frame (see example on the page 47).
BOXED[23]	framestyle=fbox, framefit=yes, heightadjust=all, framearound=all	Uppercase form, BOXED, draws frame which fits to the width of main text (usually \textwidth), predefined \hsize, or the width in \floatbox's option, but around all float elements: caption, object and foot material (see example on the page 42).

Finished on next page

Table 11 (*Finished*)

Style	\floatsetup keys	Description
Offered by fr-fancy package. They also need fancybox package.		
shadowbox[4]	style=boxed, framestyle=shadowbox	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \shadowbox from fancybox package (see example on the page 76). Besides \fboxsep and \fboxrule, there is added parameter \shadowsize—the width of shadow, default is 4pt.
Shadowbox	style=Boxed, framestyle=shadowbox	
SHADOWBOX	style=BOXED, framestyle=shadowbox	
doublebox[4]	style=boxed, framestyle=doublebox	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \doublebox from fancybox package (see example on the page 84). The frame shape is controlled by \fboxrule and \fboxsep parameters.
Doublebox	style=Boxed, framestyle=doublebox	
DOUBLEBOX	style=BOXED, framestyle=doublebox	
Additional float styles. They also need fancybox package.		
wshadowbox[4]	style=boxed, framestyle=wshadowbox	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \wshadowbox, based on \shadowbox (but drops white shade from frame, or draws edges of "second copy") from fancybox package (see example on the page 83), you may use the same frame parameters like in shadowbox style.
Wshadowbox	style=Boxed, framestyle=wshadowbox	
WSHADOWBOX	style=BOXED, framestyle=wshadowbox	

When a float style is set with frame around object which is fitted to the box width (like Boxed), and \floatbox macro uses \FBwidth command as ⟨*width*⟩ option, which sets box width equal to float contents, the width of all other float elements in this case enlarged to get width of framed object (see figure 32 on the page 48).

[1] The styles co-named and analogous to float package styles.
[2] This style is used in the sample file frsmaple01.tex
[3] This style is used in the sample file frsmaple02.tex
[4] During usage of these styles in floatrow environment you ought to enlarge space between floats, using key floatrowsep.

3.1.2 Font Settings

font Defines font for float object contents. Option analogous to font= key in \captionsetup stuff.

Available font setting options:

scriptsize	Very small size
footnotesize	The size usually used for footnotes
small	Small size

`normalsize`	Normal size
`large`	Large size
`Large`	Even larger size
`up`	Upright shape
`it`	*Italic shape*
`sl`	*Slanted shape*
`sc`	SMALL CAPS SHAPE
`md`	Medium series
`bf`	**Bold series**
`rm`	Roman family
`sf`	Sans Serif family
`tt`	`Typewriter family`
`...`	You may create your own options with the `\DeclareFloatFont` command, see page 54.

You may set font for float object like

```
font=small
```

(which is used in current documentation), or

```
font={small,sf}
```

.

If you need to color text of your float object, you may use the mechanism, created by the version **3.1** of the caption package:

```
font={small,color={blue}}
```

.

`footfont` Defines font for legends or explications (defined by the `\floatfoot` command, see *section* 2.6). This macro uses `\captionsetup` mechanism (because `\floatfoot` macro uses caption package's mechanism and utilities). By default the font size of float foot text equals to footnote text: `footfont=footnotesize`.

Font Settings for longtable. If you use caption package version 3.0**q**, the font settings, loaded in `\floatsetup` in `longtable` environment, could expand to captions. In this case, when you write something like

```
\floatsetup{font={sf,scriptsize,it}...
```

or

```
\floatsetup[longtable]{font={sf,scriptsize,it}...
```

for floats (or for [long]tables only, option [longtable] of `\floatsetup`), you ought to restore correct font size, family, shape (here) and series for caption contents and write:

```
\captionsetup{font={rm,small,up}...
```

or

```
\captionsetup[longtable]{font={rm,small,up}...
```

The version **3.1** of caption package corrects that.

3.1.3 Position of Caption

`capposition`
Caption above table object

Defines position of captions. It is similar to `position=` key in caption package, but it has two additional options: 1) `TOP`, if you prefer to align captions above objects, in the case of beside floats (in `floatrow` environment), by the top line; 2) `beside` to put caption beside object (this option could be more popular in settings for one environment, see about `\thisfloatsetup` on the page 51):

`top`	caption above object;
`TOP`	caption above object and also aligned by top line in float row. For example the `Plaintop` style is the variant of `plaintop` where used `capposition=TOP` settings, see tables 14–15;
`bottom`	caption below object;
`beside`	caption beside object.

Floatrow note. The `auto` option does not used by the `capposition=` key.

Compare two examples:

`\floatsetup[table]{style=plain,capposition=top}%` ≡ `style=plaintop`

Table 12
The table I in the row with long, long, long, long, long, long caption

Left Column Head	Data	
	I	II
First row	1	2
Second row	3	4
Third row	6	8
Fourth row	10	16

Table 13
Table II in the row with caption

Column Head	Data		
	I	II	III
First row	1	2	1
Second row	3	4	6
Third row	6	8	28

`\floatsetup[table]{style=plain,capposition=TOP}%` ≡ `style=Plaintop`

Table 14
The table I in the row with long, long, long, long, long, long caption, aligned by the top line

Left Column Head	Data	
	I	II
First row	1	2
Second row	3	4
Third row	6	8
Fourth row	10	16

Table 15
Table II in the row with caption, aligned at the top line

Column Head	Data		
	I	II	III
First row	1	2	1
Second row	3	4	6
Third row	6	8	28

Note. The option `TOP` uses `\label`—`\ref` mechanism, so, to get necessary result with it, you need to run LaTeX twice (when you make changes in contents which could change number of lines, you get correct result also on the second run).

3.1.4 Position of Beside Caption

`capbesideposition` Defines position of beside captions: vertical and horizontal. For horizontal position there are defined four options:

`left`	caption is printed to the left side of object (the default option, see example above);
`right`	caption is printed to the right side of object;
`inside`	caption is printed in binding side of page if `twoside` option switched on in document class and key `facing=yes` is used; in `oneside` option of document (or key `facing=no` is used), caption is printed at the left side;
`outside`	least popular option: caption printed in outer side of page if `twoside` option switched on in document class and key `facing=yes` is used; in `oneside` option of document (or key `facing=no` is used), caption is printed at the right side; this option makes sense for the document with usage of outer margins.

For vertical position there are defined three options

`top`	caption aligned to the top of object;
`bottom`	caption aligned to the bottom of object;
`center`	caption aligned to the center of object.

You may define position of beside caption by following:

```
capbesideposition={top,outside}  .
```

```
⟨preamble⟩
  \floatsetup[widefigure]{margins=hangleft,capposition=beside,
      capbesideposition={top,left},floatwidth=\textwidth}
⟨preamble⟩

\begin{figure*}
  \includegraphics{BlackDog}
  \caption{...}\label{...}
\end{figure*}
```

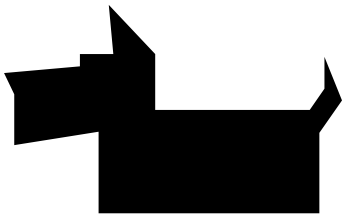


Figure 16. Wide figure with the settings of float box width `floatwidth=\textwidth`; caption beside object (on the margins), aligned by top of graphics

See examples in file `frsample02.tex` with all variants of position of captions beside float objects.

3.1.5 Defining The Width of Beside Caption

`capbesidewidth` Defines width of beside caption. This option could be more preferable in settings for one environment, see about `\thisfloatsetup` on the page 51. You may set:

```
capbesidewidth=4cm
```

.

(see figure 19). If you'll write `capbesidewidth=none` or `capbesidewidth=sidefil` (this is default key setting), the width of caption will be calculated by usual way, accordingly to float width (i.e. occupies the rest width of float box, see figure 4 on the page 12).

3.1.6 Defining Width of Object

`floatwidth` It is used for redefinition of width of objects. This key, similar to `\capbesidewidth=`:

```
floatwidth=.35\hsize
```

or

```
floatwidth=7cm
```

It could be used at first for settings of one floating environment (see page 51 about settings for current floating environment and `\thisfloatsetup`). Such settings anyway may be used for example for wide floats with the object width equal to main text width (`floatwidth=\textwidth`) and beside caption placed on the margins (see figure 56).

```
⟨preamble⟩
  \floatsetup[figure]{margins=raggedright}
⟨preamble⟩

\thisfloatsetup[figure]{floatwidth=.35\hsize}
\begin{figure}
  \includegraphics[width=\hsize]{Bear}
  \caption{...}\label{...}
\end{figure}
```

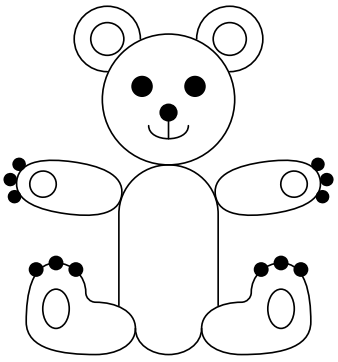


Figure 17. Graphics with settings `floatwidth=0.35\hsize` moved to the left margin

```
\thisfloatsetup{floatwidth=.35\hsize,capbesidewidth=sidefil,
      capposition=beside,capbesideposition=right}
...
```

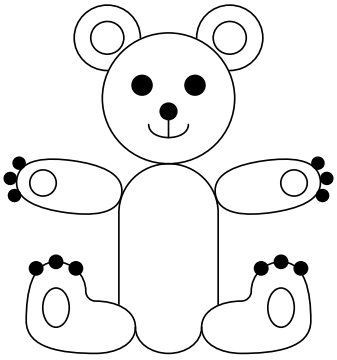


Figure 18. Caption beside graphics with the width settings `floatwidth=0.35\hsize`

(These examples you can write also using box commands with the width option: `\ffigbox[.35\hsize]` and `\fcapside[.35\hsize]` consequently.)

If you use option `floatwidth=sidefil` for objects with beside captions (in the case of key `capbesidewidth=`, uses absolute value, like `capbesidewidth=4cm`) the box with object contents (instead of caption's) occupies the rest space of float box (see figure 19 on the page 39 and appendix, figure 86 on the page 94).

3.1.7 Other Settings for Beside Captions

`capbesideframe` This boolean key declares whether the beside caption stays near the framed object (`capbesideframe=yes`) in this case caption lines will be aligned by top or bottom of frame; otherwise caption lines will be aligned with top or bottom of framed object's *contents* (`capbesideframe=no`).

```
⟨preamble⟩
  \floatsetup[figure]{style=Boxed,frameset={\fboxsep8pt},
    objectset=justified,capbesideposition={right,top},capbesideframe=yes}
  \captionsetup[figure]{...,strut=no}
⟨preamble⟩
```

```
\thisfloatsetup{capposition=beside,
      floatwidth=sidefil,capbesidewidth=4cm}
\begin{figure}
  ...
  \caption{...}\label{...}
\end{figure}
```

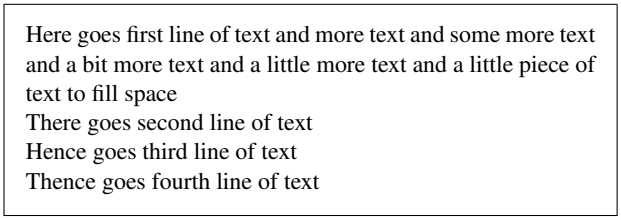


Figure 19. Caption beside framed object, (caption has width 4 cm), aligned by top of frame

```
⟨preamble⟩
  \floatsetup[figure]{...,capbesideframe=no}
  ...
⟨preamble⟩

\thisfloatsetup{capposition=beside,
      floatwidth=9cm,capbesidewidth=sidefil}
```

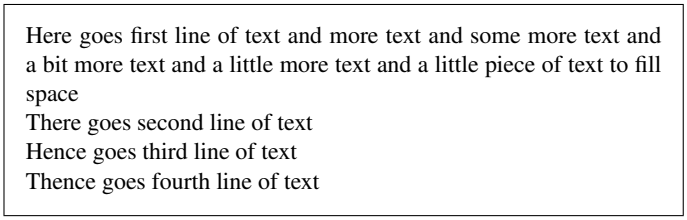


Figure 20. Caption beside framed object, (object has width 9cm), aligned by top of object contents

Floatrow note. For examples above the `\captionsetup{strut=no}` sentence also was used, which cancels struts at the beginning and end of caption (`\strut`: the rules with height and depth, which are set accordingly to current `\baselineskip`).

3.1.8 Defining Float Foot Position (Legends and Footnotes)

`footposition` Defines position of `\footnote`'s and `\floatfoot`'s in float box with above/below captions. (See examples in file `frsample01.tex`.)

`default`	if caption above float object foot material is placed below float object, otherwise below caption;
`caption`	always placed below caption;
`bottom`	always placed at the bottom of float box.

In the case of caption beside float object, footnotes and foot text are always placed below caption.

The next example shows the usage of the `caption` option of this key:

```
⟨preamble⟩
  \floatsetup{style=ruled,footposition=caption}
⟨preamble⟩

\begin{figure}
    ...
    \caption{...}\label{...}%
```

```
    \floatfoot{...}
\end{figure}
```

Figure 21 The `ruled` figure with explications which are placed under caption contents

The graphics demonstrate very pleasant muzzle of the very funny ginger cat with very fluffy fur. The cat has yellow eyes, big ears, a small pink wet nose, and thick white whiskers

3.1.9 Vertical Alignment of Float Elements

`heightadjust` Defines whether the common maximum height of objects or/and captions in the `floatrow` environment will be used for building of float row. It has following options

`all`	adjust both caption and object heights (e.g. for styles `ruled`, `Ruled` and `BOXED`);
`caption`	adjust caption heights (e.g. for `Plaintop` style);
`object`	adjust object heights (e.g. for `Boxed` style);
`none`	nothing to be adjusted (the `plain` style);
`nocaption`	no adjusting for captions;
`noobject`	no adjusting for objects;

You may define height adjustment even as followed:

```
heightadjust={caption,noobject}
```
.

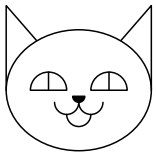
The following two examples show `ruled` and `Ruled` style. Both styles use `heightadjust=all` key option, but first style uses `capposition=top`, and second one—`capposition=TOP`.

```
⟨preamble⟩
  \floatsetup{style=ruled}
⟨preamble⟩

\begin{figure}
\begin{floatrow}
  \ffigbox
    {...}{\caption{The left ...}\label{...}}%
  \ffigbox
    {\caption{The beside ...}\label{...}}{...}
\end{floatrow}
\end{figure}
```

Figure 22 Left `ruled` figure

Figure 23 The beside figure at the right side uses settings of `ruled` layout

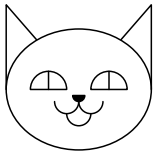


⟨*preamble*⟩
```
\floatsetup{style=Ruled}
```
⟨*preamble*⟩

```
...
```

Figure 24 Left `Ruled` figure

Figure 25 The beside figure at the right side uses settings of `Ruled` layout



`valign` Defines vertical alignment of float objects in `floatrow` if `heightadjust=all` or `heightadjust=object` keys were used, or `\floatbox` stuff uses ⟨*height*⟩ argument with value, which differs from the height of object. The options of this key are analogous to vertical alignment option in `minipage` environment and `\parbox` command. Default option is `c` (centered vertical alignment).

- `t` aligns objects by top line;
- `c` aligns objects by center line (this is default for all float styles which use `heightadjust=object` or `heightadjust=all` settings, see examples above);
- `b` aligns objects by bottom line;
- `s` stretches objects by full height (if it is possible).

Next example (figure 26) shows default vertical centered alignment for figure with changed height (remember that empty ⟨*width*⟩ option means `\hsize`).

⟨*preamble*⟩
```
\floatsetup{style=BOXED}
\usepackage{calc}
```
⟨*preamble*⟩

```
\begin{figure}
  \ffigbox[][\FBheight+2cm]
...
\end{figure}
```

Figure 26. The figure inside `\ffigbox` has ⟨*height*⟩ option, vertically centered

The example with figures 27 and 28 shows `BOXED` style, which uses `heightadjust=all` settings already, and also the `valign=t` option was added.

```
⟨preamble⟩
  \floatsetup{style=BOXED,valign=t}
  \usepackage{calc}
⟨preamble⟩

\begin{figure}
\begin{floatrow}
  \ffigbox[\FBwidth+2cm]
    {...}
    {\caption{Left ...}\label{...}}%
  \ffigbox[\FBwidth+2cm][2\FBheight]
    {\caption{The beside ...}\label{...}}
    {...}
\end{floatrow}
\end{figure}
```

Figure 27. The left beside figure uses settings for vertical top alignment

Figure 28. The beside figure at the right side in float row uses settings for vertical top alignment too

Please look at the ⟨*height*⟩ and ⟨*width*⟩ options of `\ffigbox` commands of the figure 26 and beside figures 27, 28 consequently: you may set the height and widths in this way with calc package. Right figure in the row has double height in the ⟨*height*⟩ argument of `\ffigbox`.

3.1.10 Facing Layout

`facing` This key defines whether facing layout is used for floats, if it is switched on, key options, which create different layout for even and odd pages are switched on. This key works if `twoside` option is switched on inside the document class line.

The most popular usage of `facing` key is printing of beside captions to the inner sides of pages with option `capbesideposition=inside` (the opposite option is `capbesideposition=outside`) works together with switched `facing=yes` key.

The figures 3 and 4 with beside captions in the Introduction illustrate these options—`facing=yes,capbesideposition=inside`.

3.1.11 Object Settings

`objectset justification` Defines justification of float object (float contents). Predefined options are similar to `justification=` key in `\captionsetup`.

`justified`	Blocks (in the case of a picture or text in parbox) moved to the left, the text aligned as a normal paragraph (without indentation).
`centering`	Blocks centered, each line of the object text will be centered. (This is the default.)
`raggedright`	Blocks moved to the left, each line of the text shoved to the left margin.
`RaggedRight`	As in previous item, each line of the text shoved to the left margin, too. But this time the command `\RaggedRight` of the `ragged2e` package will be used to achieve this. This difference is that this time the word breaking algorithm of TEX will work inside the text.
`raggedleft`	Blocks moved to the right, each line of the text shoved to the right margin.
`...`	You may also create your own settings with the `\DeclareObjectSet` command (see page 59)

3.1.12 Defining Float Margins

`margins` Defines margins (skips, rules or other margin material) of alone float boxes with captions above/below, of float boxes with beside captions, and of `floatrow` environments. It has following three predefined options:

`centering`	float box centered;
`raggedright`	float box flushed to the left (see figure 17);
`raggedleft`	float box flushed to the right;
`hangleft`	usually for wide floats: left edge of float boxes hangs to the margin space (there are used `\marginparwidth` and `\marginparsep` values; the `\leftskip` and `\rightskip` values are added, which have been taken from the settings of the `objectset=` key);
`hangright`	analogous to previous, right edge of floats boxes hangs to the margin space;
`hanginside`	analogous to previous, but in this option hangs inner edge for facing/twoside layout, or left margin for one side layout;
`hangoutside`	analogous to previous, but in this option hangs outer edge for facing/twoside layout, or right margin for one side layout;

`...`	You may create your own alignment settings with `\DeclareMarginSet` command, see page 59.

3.1.13 Defining Float Separators

`floatrowsep` Sets separation material between beside float boxes in one row inside `floatrow` environment (see page 18).

`capbesidesep` Sets separation material between object and beside caption (see page 11).

Both key settings work similarly to `labelsep=` key from `\captionsetup`. They use following predefined options:

`columnsep`	horizontal skip = `\columnsep` (default for both keys);
`quad`	horizontal skip = 1 em;
`qquad`	horizontal skip = 2 em;
`hfil`	horizontal skip = 1 fil (like `\hfil`);
`hfill`	horizontal skip = 1 fill (like `\hfill`);
`none`	empty separator.
`...`	You may also create your own settings with the `\DeclareFloatSeparators` command (see page 61)

This documentation uses settings `floatrowsep=qquad` for separation of beside floats and `capbesidesep=quad` for floats with beside captions.

The figure 29 uses tricky float style, which shows you layout, where the `capbesidewidth=` key with absolute value appears very useful.

```
⟨preamble⟩
  \DeclareFloatSeparators{mcapwidth}{\hskip-\FCwidth}
  \floatsetup[figure]
   {style=plain,objectset=centering,margins=centering,
    capbesidewidth=6cc,capbesideposition=left,capbesidesep=mcapwidth,
    floatwidth=sidefil}
  \captionsetup[capbesidefigure]{labelsep=newline,
    justification=raggedright}
⟨preamble⟩

\begin{figure}
\fcapside
\end{figure}
```

In this style all figures with beside captions centered accordingly to full text `\hsize`, because of the separator between float object and caption has negative value of caption width. Usage of such float layout supposes that all float objects with beside captions are narrower than `\hsize` (`\textwidth`) by at least 2 caption widths. Please note the `\FCwidth` command in the definition of `mcapwidth` key—later you may change the width of beside caption (loading e.g. `\thisfloatsetup{capbesidewidth=8cc}` settings), and, in spite of the value the separator also will be changed, picture will be anyway centered accordingly to full `\hsize`.

Figure 29
Plain figure

3.1.14 Defining Float Rules/Skips

`precode` Defines skip, rule or other analogous code above float box (see page 56).

`rowprecode` Defines skip, rule or other analogous code above alone float box, or, in the case of beside floats inside `floatrow` environment, above float row (see page 55 and 92).

`midcode` Defines skip, rule or other analogous code between caption above/below and float object.

`postcode` Defines skip, rule or other analogous code below float box (see page 56).

`rowpostcode` Defines skip, rule or other analogous code below alone float box, or, in the case of beside floats inside `floatrow` environment, below float row (see page 55 and 92).

For all these keys there are predefined following options (settings were taken from styles created in float package):

`none`	absent code (the default option for `precode=`, `rowprecode=`, `postcode=` and `rowpostcode=` keys); in `plain`, `plaintop`, `boxed`, and similar styles;
`thickrule`	thick rule (.8pt) with 2pt vertical skip below—rule above float box in `ruled` and `Ruled` styles which is used there by `precode=` key (see figures 22–25);
`rule`	rule of default thickness (.4pt), with 2pt vertical skips above and below—middle rule in `ruled` and `Ruled` styles is printed between object and caption, and used there by `midcode=` key;
`lowrule`	rule of default thickness (.4pt), with 2pt vertical skip above—rule below float box in `ruled` and `Ruled` styles, used there by `postcode=` key;
`captionskip`	vertical skip which uses the value, defined in `captionskip=` key; the default option for `midcode=` key: this option is used in `plain`, `plaintop`, `boxed`, and similar styles.
`...`	You may create your own options with the `\DeclareFloatVCode` command, see page 54.

The `rowprecode=` and `rowpostcode=` keys, in the case of unfilled row may occupy the whole width of the predefined size or get the natural width of row, depending to the defined settings of row contents (see description of the `rowfill` key, page 48).

3.1.15 Defining Float Frames

`framestyle` Defines type of frame; the floatrow package offers only two types of frames:

`fbox`	standard frame;
`colorbox`	colored frame, needs also color package; if not defined, the `\fbox` command is used instead.

`FRcolorbox` colored frame which allow to set additional material attached to its corners, needs also color package; if not defined, there is used `\fbox`;

`corners` the same as previous but without `\colorbox`—it puts the corner material only (current option doesn't need the frame definition); anyway it needs also color package.

There are options for additional frames, offered by fr-fancy package, installed with `floatrow`:

`doublebox` double frame, needs also fancybox package;
`shadowbox` frame with shadow, needs also fancybox package;
`wshadowbox` modified `shadowbox` frame (frame with "white shadow"), needs also fancybox package.

`frameset` The parameters for chosen frame; there are no predefined options for this key, just write something like:

```
frameset={\fboxrule1pt\fboxsep12pt}  .
```

The default settings for frame building with the `\fbox` command:

```
\fboxrule=.4pt \fboxsep=3pt  .
```

`framearound` Declares element of float box to be framed:

`none` no frames (usually not used);
`object` float object contents;
`all` full float box including object, caption, and any foot text;
`row` float row of beside floats, or alone float;
`none` nothing.

`framefit` Boolean which sets whether the *frame width* will be equal to current `\hsize`, predefined width or value of ⟨*width*⟩ option of float box (`framefit=yes`), in this case object size reduced (see figures 30 and 32); or the frame climbs out in the left and right sides, and *width of object* has current `\hsize`, predefined width or value of ⟨*width*⟩ option of float box (`framefit=no`, see figure 31).

```
⟨preamble⟩
  \floatsetup[figure]{framestyle=fbox,
        framearound=object,frameset={\fboxrule1pt\fboxsep10pt},
        framefit=yes}%≈ style=Boxed
⟨preamble⟩

\begin{figure}
\ffigbox[4cm]
   {...}{\caption{...}}
\end{figure}
```

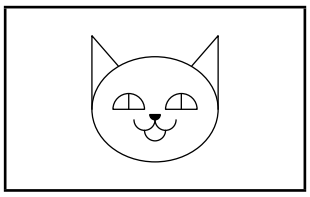


Figure 30. The frame around graphics fits to the width of float box (here: caption)

```
⟨preamble⟩
  \floatsetup[figure]{...,framefit=no}%≈ style=boxed
⟨preamble⟩

\begin{figure}
\ffigbox[4cm]
  {...}{\caption{...}}
\end{figure}
```

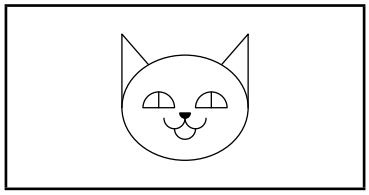


Figure 31. The frame around graphics climbs out to the right and left sides

Next follows an example with `framefit=yes` key in the case of `[\FBwidth]` option of `\ffigbox` command. In this case the width of float box (here: the width of caption) expanded to the width of framed object.

```
⟨preamble⟩
  \floatsetup[figure]{...,framefit=yes}%≈ style=Boxed
⟨preamble⟩

\begin{figure}
\ffigbox[\FBwidth]
   {...}{\caption{...}}
\end{figure}
```

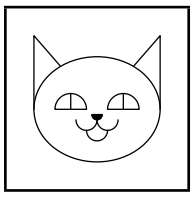


Figure 32. Framed object has natural width; the caption width expanded

`rowfill` Boolean key which in the case of `true` the material above and below float row (the `rowprecode=` and `rowpostcode=` keys) or row frames (`framestyle=row` option) will be expanded to full predefined width, otherwise the rule or frame material will have natural width of beside float boxes. (Unfilled row aligned using the `objectset=` settings.) Default value is `false`.

```
⟨preamble⟩
  \DeclareColorBox{yellowplate}{\colorbox{yellow}}
  \floatsetup{style=plain,framestyle=colorbox,
     framearound=row,colorframeset=yellowplate,frameset={\fboxrule0pt},
     framestyle=colorbox,framefit=yes,heightadjust=object,valign=c}
  \usepackage{calc}
⟨preamble⟩

\begin{figure}
\begin{floatrow}
  \ffigbox[\FBwidth+2cm]
    {...}
\end{floatrow}
\end{figure}
```

Figure 33. The left beside figure uses settings for vertical top alignment

Figure 34. The beside figure at the right side in float row uses settings for vertical top alignment too

The result you see in the row of figures 33, 34.

```
⟨preamble⟩
  \floatsetup{...rowfill=yes}
```

```
...
⟨preamble⟩
...
```

Figure 35. The left beside figure uses settings for vertical top alignment

Figure 36. The beside figure at the right side in float row uses settings for vertical top alignment too

The result you see in the row of figures 35, 36.

```
⟨preamble⟩
  \floatsetup[widefloat]{margins=hanfleft}
  \floatsetup{...}
  ...
⟨preamble⟩

...
```

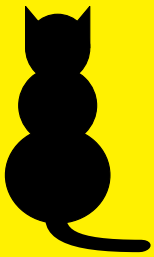


Figure 37. The left beside figure uses settings for vertical top alignment

Figure 38. The beside figure at the right side in float row uses settings for vertical top alignment too

The result you see in the row of figures 37, 38.

```
⟨preamble⟩
  ...

  \floatsetup{...rowfill=yes}
  ...
⟨preamble⟩

...
```

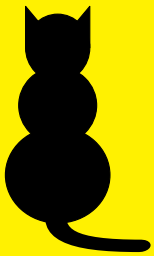


Figure 39. The left beside figure uses settings for vertical top alignment

Figure 40. The beside figure at the right side in float row uses settings for vertical top alignment too

The result you see in the row of figures 39, 40.

3.1.16 Settings for Colored Frames

`colorframeset`
`\DeclareColorBox`

This key (needs color package) defines a color box in the case of the `framestyle=colorbox` or `framestyle=FRcolorbox` settings are loaded (default is standard `\fbox`). There are not any predefined options for this key so you must define your color box option, using the `\DeclareColorBox` command like following:

```
\DeclareColorBox{mycolorbox}{\fcolorbox{red}{yellow}}
```

then use this option in `colorframeset=` key:

`colorframeset=`⟨*option*⟩ ,

for example:

```
\floatsetup{colorframeset=mycolorbox} .
```

`colorframecorners`
`\DeclareCBoxCorners`

This key defines material attached to the corners of the frame defined by the `framestyle=FRcolorbox` option. This key, as the previous one, has not predefined options; the needed material is set by the `\DeclareCBoxCorners` command (page 57).

3.1.17 Defining Float Skips

`captionskip`

Defines vertical space between caption and float object in case of `midcode` key defined as `midcode=captionskip`; or in case of usage of float styles (`style=` key) `plain`, `boxed` and similar to them:

```
captionskip=10pt .
```

The settings above are default and equal to LaTeX's settings (`\abovecaptionskip=10pt`). The settings of current documentation: `captionskip=5pt`.

`footskip`

Defines vertical space before foot material and footnotes. It can be defined like:

```
footskip=4pt ,
```

or

```
footskip=\skip\footins .
```

the last line shows default settings.

3.1.18 Defining Float Footnote Rule's Style

`footnoterule` Defines type of footnote rule for footnotes inside floating environment.

`normal`	standard LaTeX definition, the width of it equals to 0.4 of current width of text (`\columnwidth`);
`limited`	like previous one but max width of footnote rule equals to the value defined by `\frulemax` command, like: `\newcommand\frulemax{1in}`
`fullsize`	rule to full current text width.
`none`	Absent rule.
`...`	You may create your own options with the `\DeclareFloatFootnoterule` command, see page 61.

3.1.19 Managing Floats with `[H]` Placement Option

`doublefloataswide` This boolean key redefines starred floating environment *in onecolumn layout* like non-starred ones, but in this case they are still store layout settings, declared by `[wide...]` options of `\floatsetup` (page 28). This key is necessary for usage of the `[H]` option in starred environments in the same way as in non-starred.

`floatHaslist` This boolean key adds values of penalties before and after this "anchored" float like in the list environment and cancels paragraph indentation, if there is no blank line appears after environment (see also page 66).

3.2 Settings for Current Float Environment

`\thisfloatsetup` You may define some settings only for one float just before necessary environment. Command `\thisfloatsetup` could contain the same keys and options as in `\floatsetup`. It has only mandatory argument (the `\thisfloatsetup` is defined as abbreviation of the `\floatsetup[tmpset]` command).

3.3 Clearing of Settings for Current Float Type

`\clearfloatsetup` If you want to get rid of parameters marked for an automatic use within a particular environment you can use the command[1]:

`\clearfloatsetup{`⟨*float type*⟩`}` ,

where `{`⟨*float type*⟩`}`—types as `figure`, `widefloat` etc.

3.4 Temporary Clearing of All Float Settings

`\killfloatstyle` The first case when this command is needed: mixed rows of floats where figure stays beside table and you need to cancel layout of "foreign" float (see page 21). The

[1] Created as additional macro for `\clearcaptionsetup` macro, see also documentation of caption package about `\clearcaptionsetup` command

`\killfloatstyle` command is used before any command of `\floatbox` stuff (see section 2.1).

Another case—layout of floats with beside captions is quite different from other subtypes: `[figure]` option of floatsetup defined with `style=plain` and `[cabesidefigure]` must be defined with `style=boxed`. In this case you may define your command, based on predefined `\fcapside`:

```
⟨preamble⟩
  \newcommand\myfcapside{\killfloatstyle
       \floatsetup[figure]{style=Boxed,capbesideframe=yes}\fcapside}  .
⟨preamble⟩

\begin{figure}
\myfcapside[\FBwidth]
...
\end{figure}  .
```

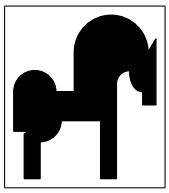


Figure 41. Figure with beside caption in `Boxed` style. The special command `\myfcapside` created to change layout for figures from plain in the case of captions below float to boxed in the case of caption beside

The option `[figure]` is necessary if you have defined settings for this option in the preamble.

Notes.

1) Please remember that such command with redefined settings can be placed only *inside an environment* or *group*.

2) Before creation of such risky command, please revise your layout settings: maybe the `[widefigure]` option never used in your documentation settings, so you can define necessary settings in `\floatsetup[widefigure]{style=Boxed,capposition=beside...}` and then use "starred" floats in following way:

```
⟨preamble⟩
  \floatsetup[widefigure]{style=Boxed,capposition=beside,
      capbesideframe=yes}
⟨preamble⟩

\begin{figure*}
\fcapside[\FBwidth]
...
\end{figure*}  .
```

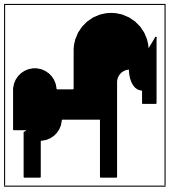


Figure 42. Figure with beside caption in `Boxed` style. The special settings for framed graphics were created in "starred" environment

3.5 The Default Float Type Settings

The following keys and options are switched on when the floatrow package loaded. They equal to `default` style:

```
font=normalsize
footfont=footnotesize
capposition=bottom
capbesideposition=left
capbesideframe=no
footposition=default
heightadjust=none
facing=no
margins=centering
objectset=centering (≡justification=centering, caption)
floatrowsep=columnsep
capbesidesep=columnsep
precode=none
rowprecode=none
postcode=none
rowpostcode=none
framearound=none
rowfill=no
midcode=captionskip
captionskip=10pt
```

3.6 Defining New Options

In the next few sections a list of commands is presented, which help to define additional key options for the `\floatsetup` command.

3.6.1 Float Style Option (`style=`)

`\DeclareFloatStyle` Defines new float style. Example shows definition of new float style `MyBoxed`. The figures 43, and some others in current documentation show result.

```
\DeclareFloatStyle{MyBoxed}{style=Boxed,captionskip=5pt,
      frameset={\fboxrule1pt\fboxsep12pt}}
\floatsetup[figure]{style=MyBoxed}
```

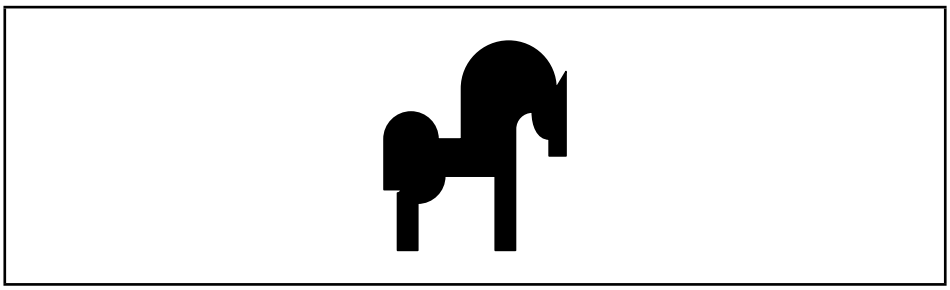


Figure 43. Plain figure in `MyBoxed` style

Much more, more and more and more and more and more and more and more and more text inside macro `\floatfoot`

The same result you get with:

```
\floatsetup[figure]{style=Boxed,captionskip=5pt,
      frameset={\fboxsep12pt\fboxrule1pt}}
```

3.6.2 Float Font Option (`font=`)

`\DeclareFloatFont`

With this macro you may define new option for font (`font=` key) of float contents. This macro works like `\DeclareCaptionFont` in caption package: you may also use key options declared by `\DeclareCaptionFont` command.

To get red color for text in the example with figure 57 on the page 61, you may define the red color by following way:

```
\DeclareFloatFont{red}{\color{red}}
```

and then write, for example

```
\floatsetup[figure]{font={small,red}}  .
```

The version **3.1** of the caption package offers special option inside `font=` key. Since the floatrow package uses the same mechanism for its `font=` key, the example above you can write as following:

```
\floatsetup[figure]{font={small,color={red}}}  .
```

3.6.3 Option for Float Rules/Skips (`precode=` etc.)

`\DeclareFloatVCode`

This command defines the skip, rule or other analogous code above and below full float box and between caption above/below and object. The defined option might be used in `rowprecode`, `precode`, `midcode`, `postcode`, and `rowpostcode` keys (page 45).

Compare two examples:

```
⟨preamble⟩
  \DeclareFloatVCode{grayruleabove}%
        {{\color{gray}\par\rule\hsize{2.8pt}\vskip4pt\par}}
  \DeclareFloatVCode{grayrulebelow}%
        {{\color{gray}\par\vskip4pt\rule\hsize{2.8pt}}}
  \floatsetup{...,heightadjust=all,valign=c,
        rowprecode=grayruleabove,rowpostcode=grayrulebelow}
⟨preamble⟩
```

```
\begin{figure}
\begin{floatrow}
  \ffigbox
    {...}{\caption{The left ...}\label{...}}%
  \ffigbox
    {...}{\caption{The beside ...}\label{...}}
 \end{floatrow}
\end{figure}
\begin{figure}
  ...
  \caption{Alone figure ...}\label{...}%
\end{figure}
```

Figure 44. The left beside figure inside float row with defined row rules above and below

Figure 45. The beside figure at the right inside float row with defined row rules above and below

Figure 46. Alone figure with defined row rules above and below

```
⟨preamble⟩
  ...
  \floatsetup{...,heightadjust=all,
     precode=grayruleabove,postcode=grayrulebelow}
⟨preamble⟩
```

...

Figure 47. The left beside figure inside float row with defined rules for float box

Figure 48. The beside figure at the right inside float row with defined rules for float box above and below

Figure 49. Alone figure with defined rules above and below for float box

Please note that for ruled styles defined for boxes, like for figures 47 and 48, which could be placed in one row, you need to set `heightadjust=all` key.

The examples with unfill rows.

```
⟨preamble⟩
  \floatsetup{...,heightadjust=all,valign=c,
        rowprecode=grayruleabove,rowpostcode=grayrulebelow}
⟨preamble⟩

\begin{figure}
\begin{floatrow}
  \ffigbox[\FBwidth]...
  \ffigbox[\FBwidth]...
 \end{floatrow}
\end{figure}
```

Figure 50. The left beside figure inside unfill float row with defined row rules above and below

Figure 51. The beside figure at the right inside unfill float row with defined row rules above and below

The same, but with `rowfill` option.

```
⟨preamble⟩
  \floatsetup{...,rowfill=yes}
⟨preamble⟩

...
```

Figure 52. The left beside figure inside unfill float row with defined row full size rules above and below

Figure 53. The beside figure at the right inside unfill float row with defined row full size rules above and below

3.6.4 Settings for Colored Frame (`colorframeset=`)

`\DeclareColorBox` Let's repeat the command for definition of colored box used by `colorframeset=` key (see also page 50). Here is defined frame for figure 56 below:

```
\DeclareColorBox{framedfigure}{\fcolorbox{gray}{white}}  .
```

The yellow plate for figure rows on the page 48:

```
\DeclareColorBox{yellowplate}{\colorbox{yellow}}  .
```

Please note, that for correct positioning of the color plate during usage of the `\colorbox` command you need set to zero value for the `\fboxrule` command in the `frameset` option:

```
frameset={\fboxrule0pt}  .
```

`\DeclareCBoxCorners` If you use the `FRcolorbox` option for the `framestyle` key (page 45), you may set additional material (rules or something), attached to four corners.

```
\DeclareCBoxCorners{⟨option⟩}{⟨llcorner⟩}{⟨lrcorner⟩}{⟨urcorner⟩}{⟨ulcorner⟩}
```

The {⟨*option*⟩} argument defines name of option of the `colorframecorners` key. The four others define material attached to four corners.

The order of corner material analogous to the order in the METAPOST's `bbox` box for the `label` command: first goes lower left corner ({⟨*llcorner*⟩}) then, counterclockwise, lower right corner ({⟨*lrcorner*⟩}), upper right corner ({⟨*urcorner*⟩}) and last goes upper left corner ({⟨*ulcorner*⟩}). There are used modified commands of `picture` environment inside these arguments: all lengths and coordinates must have units like points, millimeters etc., but here you may use usual length parameters like `\textwidth`. When

the color box is created the \FRcolorboxht, \FRcolorboxwd and \FRcolorboxdp parameters define height, width and depth of the box, you may use them inside settings of the \DeclareCBoxCorners xommand. You may use the \floatfacing command to create facing layout.

The example with material in all corners, which shows overlapping.

```
\DeclareCBoxCorners{angles}
   {{\color{green}%green llcorner
      \linethickness{10pt}\put(-5pt,-5pt)
      {{\put(0pt,0pt){\line(0,1){\FRcolorboxht}}}%
       {\put(-5pt,0pt){\line(1,0){\FRcolorboxwd}}}}%
   }}{{\color{red}%red lrcorner
      \linethickness{10pt}\put(0pt,0pt)
      {{\put(0pt,0pt){\line(0,1){\FRcolorboxht}}}%
       {\put(5pt,0pt){\line(-1,0){\FRcolorboxwd}}}}%
   }}{{\color{blue}%blue urcorner
      \linethickness{10pt}\put(5pt,-5pt)
      {{\put(0pt,0pt){\line(0,-1){\FRcolorboxht}}}%
       {\put(5pt,0pt){\line(-1,0){\FRcolorboxwd}}}}%
   }}{{\color{magenta}%magenta ulcorner
      \linethickness{10pt}\put(0pt,0pt)
      {{\put(0pt,0pt){\line(0,-1){\FRcolorboxht}}}%
       {\put(-5pt,0pt){\line(1,0){\FRcolorboxwd}}}}%
   }}
```

Please note, that this material has not any width and its values do not used during calculation of frame position and width. Please note also that material in the left lower and upper corners will be covered by frame, but right lower and upper corner material cover the frame (inside these “layers” the material from upper corners covers lower ones) the object contents appear in the upper layer.

```
\floatsetup{style=Boxed,framestyle=FRcolorbox,
  colorframeset=yellowplate,colorframecorners=angles,
  frameset={\fboxrule=0pt\fboxsep=2pt},framefit=yes,captionskip=15pt}
```

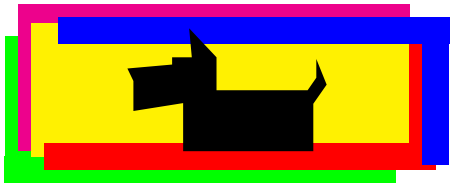


Figure 54. The picture on the color plate with multicolored corners

The same but without color plate.

```
\floatsetup{style=Boxed,framestyle=corners,colorframecorners=angles,
```

```
frameset={\fboxrule=0pt\fboxsep=2pt},framefit=yes,captionskip=15pt}
```

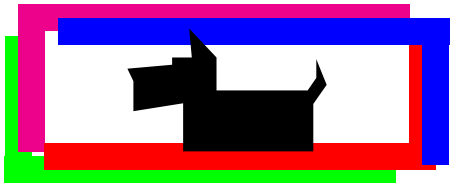


Figure 55. The picture on the "transparent" box with multicolored corners

3.6.5 Object Justification Option (`objectset=`)

`\DeclareObjectSet` You may define justification for `objectset=` key (page 43) like:

```
\DeclareObjectSet{centering}{\centering}
```

In option's definition you may try to include any regular commands (it could be the repeated head text also) which you need to put before each object contents in float environment. You may also use key options declared by `\DeclareCaptionJustification` command of caption package as options for `objectset=` key.

3.6.6 Option for Float Box Alignment/Settings (`margins=`)

`\DeclareMarginSet` You may define box alignment for float box (`margins=` key) like:

```
\DeclareMarginSet{center}{%
  \setfloatmargins{\hfil}{\hfil}}
```

or like (see also sample files):

```
\DeclareMarginSet{outside}{\setfloatmargins*{\hfil}{}}
```

The `\DeclareMarginSet` command uses the `\setfloatmargins` command, which defines fill code for each margin.

`\setfloatmargins` Non-starred form of `\setfloatmargins` defines left and right margin.

`\setfloatmargins{`⟨*left margin*⟩`}{`⟨*right margin*⟩`}`

Here goes rather complex example which was created as alternative float layout for one-column document. The starred, `figure*`, environment places caption on the left margin, beside object. Frame around object has default width of main text.

⟨*preamble*⟩

```
\makeatletter\@mparswitchfalse\makeatother

\DeclareMarginSet{hangleft}%
  {\setfloatmargins
    {\hskip-\marginparwidth\hskip-\marginparsep}{\hfil}}

\DeclareColorBox{framedfigure}{\fcolorbox{gray}{white}}

\DeclareFloatSeparators{marginparsep}{\hskip\marginparsep}
\floatsetup[widefigure]{margins=hangleft,floatwidth=\textwidth,
   capposition=beside,capbesideposition=left,capbesideframe=yes,
```

```
    capbesidewidth=\marginparwidth,capbesidesep=marginparsep,
    framestyle=colorbox,framefit=yes,
    colorframeset=framedfigure,frameset={\fboxrule3pt\fboxsep8pt}}
  \captionsetup[capbesidefigure]{justification=RaggedRight,
      font=small,labelfont={normalsize,sf,bf},labelsep=newline,strut=no}
⟨preamble⟩

\begin{figure*}
...
\end{figure*}
```

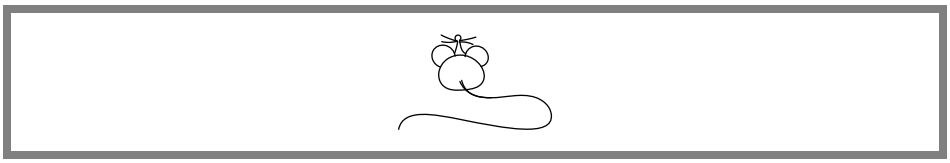


Figure 56
Figure with alternative layout ("starred" environment) caption placed on the left margin

Note. The row of figures 8–11 on the page 20 uses the same `margin=` settings of option `margins=`.

Starred form, `\setfloatmargins*`, defines facing layout: inside and outside margin.

`\setfloatmargins*{`⟨*inside margin*⟩`}{`⟨*outside margin*⟩`}`

You may even set much more complex definition:

```
\DeclareMarginSet{facingrule}{%
\setfloatmargins*{%
   \floatfacing{\hskip-12pt\vrule width4pt\hskip8pt\hfill}%
               {\hfill\hskip8pt\vrule width4pt\hskip-12pt}}{}}
```

`\floatfacing` the `\floatfacing` defines following

`\floatfacing{`⟨*odd page definition*⟩`}{`⟨*even page definition*⟩`}`

This macro has also starred form `\floatfacing*`, which you can use in key options for `\captionsetup` stuff and for floats with beside captions.

Note. Please remember that all options, which set different layout for facing pages need `facing=yes` key option.

`\floatboxmargins` `\floatrowmargins` `\floatcapbesidemargins`

The `\setfloatmargins` could be "separated" into the three macros which set margins for three main variants of float positions:

`\floatboxmargins` sets left/right margins around alone float box;
`\floatrowmargins` sets left/right margins around `floatrow` environment;
`\floatcapbesidemargins` sets left/right margins around alone float box with beside caption.

The grammar for using three mentioned commands is similar to `\setfloatmargins`. Again, the settings which use `\floatfacing` command work only in the case when key `facing=yes` is used.

Alignment Settings for longtable. The floatrow expands some settings of table layout to the `longtable` environment, so you may set `\LTleft` and `\LTright` parameters inside `\DeclareMarginSet` settings. For example, `centering` option was defined like:

```
\DeclareMarginSet{centering}{\setfloatmargins{\hfill}{\hfill}%
  \LTleft=\fill \LTright=\fill}
```

3.6.7 Float Separators Options (`floatrowsep=`, `capbesidesep=`)

`\DeclareFloatSeparators` You may define separator between float boxes, or between float object and beside caption:

```
\DeclareFloatSeparators{columnsep}{\hskip\columnsep}
```

.

Please remember, that you may use options defined with `\DeclareFloatSeparators` by both `floatrowsep=` and `capbesidesep=` keys. You may also use key options declared by `\DeclareCaptionLabelSeparator` command.

The next example uses more complex separator, which uses, the color package.

⟨*preamble*⟩
```
\DeclareObjectSet{colorred}{\parskip2pt\parindent15pt\color{red}}
\DeclareFloatSeparators{colorsep}%
  {\begingroup\color{blue}%
    \hskip8pt\vrule width4.8pt\hskip8pt\endgroup}
\floatsetup[widefigure]{margins=hangleft,capbesidesep=colorsep,
   objectset=colorred,floatwidth=\textwidth}
\captionsetup[figure]{justification=justified,
    labelfont={color={magenta},bf},textfont={color={green}},
    labelsep=newline}
```
⟨*preamble*⟩

```
\begin{figure*}
...
\end{figure*}
```

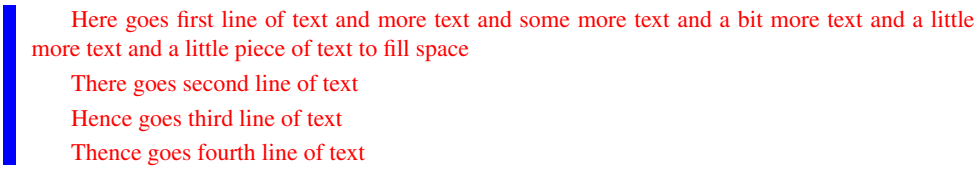


Figure 57
Multi-colored figure with beside caption. And A bit more text, and some more text

Note. The settings of color of caption font like `labelfont={color={magenta},bf,}` `textfont={color={green}}` were documented first time in the caption documentation version **3.1**.

3.6.8 Option for Footnote Rule's Style (`footnoterule=`)

`\DeclareFloatFootnoterule` You may define new footnoterule (`footnoterule=` key) like:

⟨*preamble*⟩
```
...
\usepackage{ifthen}
```

```
\renewcommand\frulemax{72pt}
\newcommand \Limitedrule{.33\columnwidth}
\DeclareFloatFootnoterule{Limited}{\kern-3pt
  \def\Limitedrule{.33\columnwidth}%
  \ifthenelse{\lengthtest{\frulemax<\Limitedrule}}%
         {\def\Limitedrule{\frulemax}}{}%
  \hrule width\Limitedrule\kern2.6pt}
```

⟨*preamble*⟩

Remember, that the summary vertical height for footnote rule must be equal to 0pt.

4 Creation of New Float Types

\DeclareNewFloatType For creation of new float type the \DeclareNewFloatType command was created which also uses ⟨*key*⟩ = ⟨*value*⟩ mechanism:

\DeclareNewFloatType{⟨*type*⟩}{⟨*options*⟩}

The ⟨*type*⟩ argument includes the new floating environment name.
The ⟨*options*⟩ could include the following keys:

placement The value of this key could contain any combination of the letters t, b, h, and p, which define the placement of current float type on the page in the case floating environment has no option argument. (As default is declared placement=tbp.)

name Defines the name of environment in the caption label. (As default for caption label is declared the name of environment.)

fileext Defines extension of the file in which gathered list of floats.

Note. In the version v0.2b, in the case of this key not defined, the captions of one type are gathered in the file with extension, co-named to current floating environment with perfix "lo". This new feature allows to create separate float lists by default.

within Declares the section head of document, by which current float resets its numbering to zero. If this key is absent, the float numbering increases during whole documentation.

relatedcapstyle In the float package the non-starred \newfloat/\restylefloat macros attach the related caption style for float styles (see section 5.1). If you use \DeclareNewFloatType mechanism and exists (you created it by \captionsetup[...]) co-named, i.e. related, caption style you may attach this style with key relatedcapstyle=yes.

Below is an example of the \DeclareNewFloatType command, which was used for definition of the Example environment demonstrated on page 104. It consists of following code:

```
\DeclareNewFloatType{Example}%
    {placement=t,within=section,fileext=loe}
```

4.1 How to replace \newfloat with \DeclareNewFloatType

The \newfloat command takes three required and one optional argument:

\newfloat{⟨*type*⟩}{⟨*placement*⟩}{⟨*ext*⟩}[⟨*within*⟩]

which could be replaced with

\DeclareNewFloatType{⟨*type*⟩}%
    {placement=⟨*placement*⟩,fileext=⟨*ext*⟩,widthin=⟨*within*⟩}

The float package offers also other commands of float type declaring: the \floatname command can be replaced by the name= key of \DeclareNewFloatType command; the \floatplacement—by the placement= key.

5 Borrowed Code

5.1 The float Package: Compatibility

The floatrow package includes some macros of float (version v1.3d, dated 2001/11/08) with necessary modifications. In the case of loaded float package *before* floatrow you'll get error message.

Note. In the case of some packages could call float package[1] the floatrow package loads code which emulates already loaded float package v1.3, so future requests for this package will be ignored. This will help to avoid strange error messages in the case of these packages loaded after floatrow. Please note that packages, which load float must be loaded *after* floatrow.

I hope that old documents, which use the float package, could work with floatrow. The first limitation or feature is—if you didn't use any `\restylefloat` command—all figures and tables appear in `plain` float style with bottomed captions. Another limitation—you ought to put all `\newfloat` and `\floatstyle` and `\restylefloat` commands in preamble, before `\begin{document}`. The commands `\restylefloat`, `\newfloat` and `\floatstyle` are obsolete but supported[2] (see section below).

The sections below explain how float commands and options work in floatrow. Sections, signed with "[float]" and typed with slanted font, were borrowed from float's documentation. The section which describes commands of layout settings of float package was moved in the section 11.3 (subsection 11.3.1, "The User Interface—New Floats [float]"), this section describes obsolete stuff.

5.1.1 How Settings From The float Package Work in floatrow

The combination of command `\floatstyle{`⟨*style*⟩`}` and one of commands

```
\floatstyle{⟨style⟩}
\newfloat{⟨float⟩}
```

or

```
\floatstyle{⟨style⟩}
\restylefloat{⟨float⟩}
```

in floatrow package set float layout in the following way:

```
\floatsetup[⟨float⟩]{style=⟨style⟩}
```

Please note that there is used `\floatsetup[`⟨*float*⟩`]{...}` settings for current type of float, but not `\floatsetup{...}`.

[1] I'm aware about algorithm package.

[2] The better way is to use `\floatsetup` macros. The floatrow package supports obsolete macros but there is no guarantee that they will work as expected.

5.1.2 Printing of Float List [float]

\listof

The \listof command produces a list of all the floats of a given class. Its syntax is

\listof{⟨type⟩}{⟨title⟩}

⟨type⟩ is the float type given in the \newfloat command. ⟨title⟩ is used for the title of the list as well as the headings if the current page style includes them. Otherwise, the \listof command is analogous to the built-in LaTeX commands \listoffigures and \listoftables.

5.1.3 The User Interface—[H] Placement Specifier [float]

Anchored float

Many people find LaTeX's float placement specifiers too restrictive. A Commonly Uttered Complaint (CUC) calls for a way to place a float exactly at the spot where it occurs in the input file, i.e., to not have it float at all. It seems that the [h] specifier should do that, but in fact it only suggests to LaTeX something along the lines of "put the float here if it's OK with you". As it turns out, LaTeX hardly ever feels inclined to actually do that. This situation can be improved by judicious manipulation of float style parameters.

The same effect can be achieved by changing the actual method of placing floats. David Carlisle's here option introduces a new float placement specifier, namely [H], which, when added to a float, tells LaTeX to "put it HERE, period". If there isn't enough space left on the page, the float is carried over to the next page together with whatever follows, even though there might still be room left for some of that. This style option provides the [H] specifier for newly defined classes of floats as well as the predefined figures and tables, thereby superseding here. David suggests that the here option be withdrawn from the archives in due course.

The [H] specifier may simply be added to the float as an optional argument, like all the other specifiers. It may not be used in conjunction with any other placement specifiers, so [Hhtbp] is illegal. Neither may it be used as the default placement specifier for a whole class of floats. The following table is defined like this:

```
\begin{table}
\begin{tabular}{cl}
\tt t & Top of the page\\
... more stuff ...

\end{tabular}
```

(It seems that I have to add some extraneous chatter here just so that the float actually comes out right in the middle of a printed page. When I LaTeXed the documentation[1] just now it turned out that there was a page break that fell exactly between the "So now" line

[1] For float package.

and the float. This wouldn't Prove Anything. Bother.) So now we have the following float placement specifiers:

`t`	Top of the page
`b`	Bottom of the page
`p`	Page of floats
`h`	Here, if possible
`H`	Here, definitely

Floatrow note. Please don't mix meaning of `[H]` and `[h]` options. Float with `[h]` and `[!h]` option, if succeed, appears *after completing line* of text, where it was appeared in the source file. That could be visible if you put floating environment within a paragraph (and at the middle of line also).

The `[H]` option places the float just *at the point* where it appeared in the source file, it is used (*but that strongly not recommended when typesetting books*!) for floats after text like "... shown in this **figure:**", i.e. the `[H]` float, almost like math formulas, continues the current paragraph.

5.1.4 The [H] Placement Specifier—Managing of Page Breaks

The strange phrase at the end of previous paragraph, "almost like math formulas" means, that "anchored" floats have no management of page breaking, and also the text, typed without blank line after float, always gets `\parindent`.

To follow the idea of `\allowdisplaybreaks` command from amsmath package there is created a *beta-temp*[1] version of listpen package (it can be used separately). It offers commands, which manage the penalty values in the list environments:

`\allowprelistbreaks` sets penalty before lists (and also "anchored" floats);
`\allowpostlistbreaks` sets penalty after lists;
`\allowitembreaks` sets penalty between list items (surely, this command not for floats!).

All of them can be set globally, inside groups, and inside environments. These penalties are set accordingly to digits from `[-4]` (never break) to `[4]` (always break). The positive values of optional argument in these commands analogous to values of optional arguments in `\pagebreak` command. The negative ones—to optional arguments `[1]`–`[4]` in `\nopagebreak` command. The default value of all three commands is `[-1]` which equal to settings of standard LaTeX classes: book, article etc. (`[-1]` option equal to `\@lowpenalty` value).

`floatHaslist` The key, if true, uses list penalties, otherwise anchored float works without any penalty, i.e. like defined in float.

Also (added in version 0.1k with current key): Since list environments do *not make indentation* in the paragraphs next to them, in the case of *no blank* line after environment, the "anchored" floating environment does the same, if this option is true. Default value of `floatHaslist` is `false` (for backward compatibility with previous version 0.1j).

[1] I hope that such support sooner or later could appear in paralist package and think it is better to follow grammar of master-package for similar situations.

\floatHpenalties This macro, defined with \renewcommand can include settings for list penalties around anchored floats. If you define

```
⟨preamble⟩
  \makeatletter
  \renewcommand\floatHpenalties{\@beginparpenalty\@M}
  \makeatother
⟨preamble⟩
```

or, with listpen package

```
⟨preamble⟩
  \renewcommand\floatHpenalties{\allowprelistbreaks[-4]}  ,
⟨preamble⟩
```

you'll never get page breaks before anchored floats.

\RestoreSpaces \RemoveSpaces The commands-aliases of the \if@nobreak flag were added. The first is equal to \@nobreakfalse. The main (and most visible) usage of this flag is for managing vertical spaces: The true value in the case of two sectioning commands cancels usage of the space before next \..section command of the pair; in the case of spaces around list environments it cancels usage of the space before list just after sectioning command. Usually the \@nobreakfalse flag toggles at the next paragraph (or \par command), but in some cases this "toggling" cannot be happen in necessary point. The \RestoreSpaces command would help. Opposite command \RemoveSpaces equals to \@nobreaktrue.

5.2 The rotfloat Package

Code of rotfloat package was also borrowed by floatrow package. This package originally allows to expand settings of float package to rotated environments like sidewaysfigure and sidewaystable. This mechanism was borrowed to expand the floatrow's settings in the similar way.

In the case of loaded rotfloat package *before* floatrow you will get error message.

The floatrow package loads code which pretends that rotfloat is already loaded, so next loads are ignored. The rotfloat allowed in the \usepackage line with rotating package, which could have options. It is necessary to delete rotfloat package from \usepackage line where also rotating package loaded with options: otherwise you may get an 'option clash' error message.

6 The floatrow Package and The caption Package

Tested (and compatible) with caption version from v3.0q to v3.1j.

The caption package has strong mechanism for creation of caption layout, so floatrow addresses the creation of new caption styles to this package (see documentation for caption package[1]).

The floatrow package adds a possibility to create variations of caption layouts for floats in different positions or float layouts (e.g. like wide or two-column floats, rotated floats, wrapped floats) in the same time when `\floatsetup` settings were loaded, using the same optional argument in `\captionsetup` settings.

For example you want to create a special caption layout for wide or two-column floats. In this case you may use

```
\captionsetup[widefloat]{⟨options⟩}
```

or for wide or two-column figures:

```
\captionsetup[widefigure]{⟨options⟩}
```

The priority of `\captionsetup` optional arguments is similar to `\floatsetup` ones: in current examples `\captionsetup[widefigure]` will be stronger than `\captionsetup[widefloat]`—the priority and usage of "⟨*float subtypes*⟩" in optional arguments see on page 29.

6.1 Managing of Float Parts With the `subfloatrow` Environment

`\subcaption`

The version 3.1 of caption package offers possibility for creation of subcaptions, using the `subtype` settings (and `\DeclareCaptionSubType` command, see caption documentation), which allow to create captions for parts of floats.

In this section you may see some examples with building of rows of beside parts of floats.

Subcaption above subtable

The example with subtables 16, *a* and 16, *b* (table 16).

```
⟨preamble⟩
  ...
  \DeclareCaptionSubType[alph]{table}
  \captionsetup[subtable]{labelformat=brace,textfont=md,labelfont=up}
  \floatsetup[subtable]{style=Plaintop}%
⟨preamble⟩

\begin{table}
\ttabbox[\FBwidth]
{\begin{subfloatrow}
  \ttabbox
    {\caption{First subtable}\Flabel{...}%
     \begin{tabular}{..}...
  \ttabbox...
\end{subfloatrow}}
```

[1] The English documentation is ⟨*texmf folder*⟩`/doc/latex/caption/caption-eng.pdf`.

```
{\caption{Two ...}\Flabel{...}}
\end{table}
```

Table 16
Two subtables (captions for parts of float created with \caption command)

a) First subtable

Column Head	Data	
	I	II
First row	1	2
Second row	3	4
Third row	6	8
Fourth row	10	16

b) Second subtable inside of \ttabbox and floatrow environment

Column Head	Data	
	I	II
First row	1	2
Second row	3	4
Third row	6	8

Please note that for the labels of table parts the special option brace of the labelformat key was used.

subfloatrow The subfloatrow is analogous to the floatrow environment[1]. The usage is similar to floatrow, you may write for example:

```
\begin{subfloatrow}[⟨number of beside parts of floats⟩]
\floatbox...
\floatbox...
...
\end{subfloatrow}
```

i.e. by default there are allowed two parts of floats. For other number of parts you ought to put number in the optional argument. This environment puts horizontal separator, defined by subfloatrowsep= key. This key uses the same options as floatrowsep= and capbesidesep= keys (options of these keys defined by the \DeclareFloatSeparators command).

Inside the subfloatrow environment you may use the \caption command, which this time creates the label for parts of float. This is because of setting

```
\captionsetup{subtype}
```

at the very beginning of this environment.

Note: With the floatrow package you may use also \captionsetup[subfloat] settings, the caption package offers the \captionsetup[subtype]{...} settings which will be stronger than previous, to say nothing about \captionsetup[subfigure]{...} for parts of figure, which are strongest. (Please note that in caption terms word "subtype" means part of float.)

Next follows an example with beside main caption (figure 58).

[1] It skips some features of "parent" environment, (e.g. margins or margin material this environment build box and follows objectset= option).

⟨*preamble*⟩
```
  ...
  \DeclareCaptionSubType[alph]{figure}
  \captionsetup[subfigure]{labelformat=brace,justification=centerlast}

  \floatsetup[figure]{style=Shadowbox,capbesidesep=columnsep,%
      capbesideframe=yes,capbesideposition={left,bottom}}
  \floatsetup[subfigure]{style=plain,heightadjust=object}
```
⟨*preamble*⟩

```
\begin{figure}
\fcapside[\FBwidth]
  {\begin{subfloatrow}
    \ffigbox[\FBwidth]{\caption{One funny cat}\Flabel{...}...}{}

    \ffigbox[\FBwidth]{\caption{Another pleasant cat}\Flabel{...}...}{}%
   \end{subfloatrow}}
  {\caption{... \Fref{...} and \Fref{...}}\label{...}}
\end{figure}
```

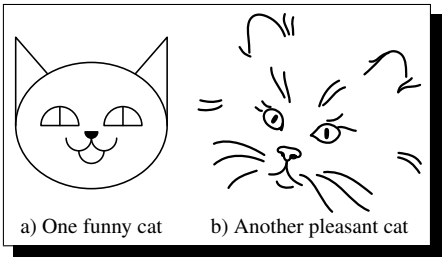


Figure 58. Beside figure caption vertically bottom aligned; fancy `Shadowbox` layout. There are two parts: 58, *a* and 58, *b*

In the next example the main caption will be placed below, but labels of figure parts were printed beside (see figure 59). For this reason the `\useFCwidth` command was used, which creates the width of caption box equal to natural caption width.

⟨*preamble*⟩
```
  ...
  \captionsetup[subfigure]{labelformat=brace,list=off}

  \floatsetup[subfigure]{style=plain,capbesideposition=left,
        capbesidesep=space,heightadjust=object}
```
⟨*preamble*⟩

```
\begin{figure}[H]
  \ffigbox[\FBwidth]
    {\begin{subfloatrow}\useFCwidth
       \fcapside[\FBwidth]{\caption{}\Flabel{...}...}{}
       \fcapside[\FBwidth]{\caption{}\Flabel{...}...}{}
     \end{subfloatrow}}
    {\caption[...]{...}\label{...}}
\end{figure}
```

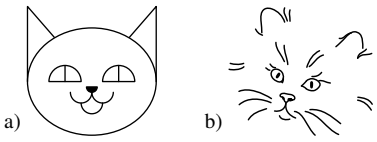


Figure 59. Two parts of figure in a row with labels beside. Main caption below. There are two subfigures: 59, *a* and 59, *b*

In the next example the difference from previous layout settings is in usage of the top vertical alignment. The height of the right graphics was enlarged by 1cm just to show how the alignment for parts (here is default centering alignment) and the top alignment for their captions (they are aligned by top) works.

⟨*preamble*⟩
```
  ...
  \floatsetup[subfigure]{style=plain,heightadjust=object,
        capbesideposition={left,top},capbesidesep=space}
```
⟨*preamble*⟩

```
\begin{figure}[H]
  \ffigbox[\FBwidth]
    {\begin{subfloatrow}\useFCwidth
       \fcapside[\FBwidth]{\caption{}\Flabel{...}...}{}
       \fcapside[\FBwidth][\FBheight+1cm]{\caption{}\Flabel{...}...}{}
     \end{subfloatrow}}
{\caption[...]{...}\label{...}}
\end{figure}
```

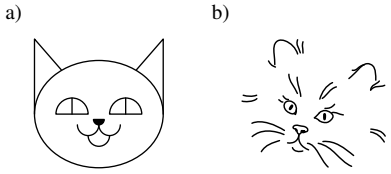


Figure 60. Two parts of figure centered vertically; beside labels aligned by top. Main caption below. There are two subfigures: 60, *a* and 60, *b*

Another example (61) demonstrates, that you may not only use the option `style=plain` for parts of float, and there can not only be labels for beside subcaptions.

⟨*preamble*⟩
```
  \captionsetup[subfigure]{labelformat=brace,justification=rightlast,
        format=hang}

  \floatsetup[figure]{style=plain}
```

```
  \floatsetup[subfigure]{style=BOXED,capbesideposition={left,top}}
⟨preamble⟩

\begin{figure}
\ffigbox
  {\begin{subfloatrow}
    \fcapside[1.1\FBwidth]{\caption{One ...}\Flabel{...}...}{}

    \fcapside[1.1\FBwidth]{\caption{Another ...}\Flabel{...}...}{}%
   \end{subfloatrow}}
  {\caption{... \Fref{...} and \Fref{...}}\label{...}}
\end{figure}
```

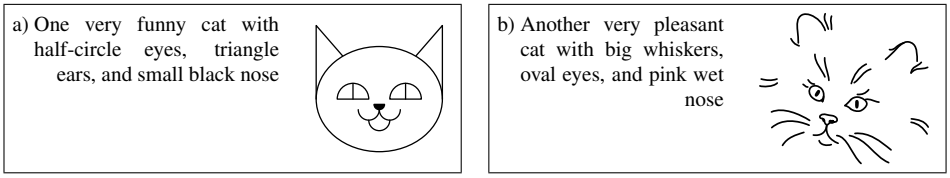


Figure 61. Beside subcaptions vertically top aligned. There are two subfigures: 61, *a* and 61, *b*

`\captionlabel`
`\subcaptionlabel`

The last example demonstrates new command `\subcaptionlabel` for caption labels, which can be used inside, e.g., `picture` environment or as replacing text in `psfrag` command of psfrag package. Unlike the `\caption` and `\subcaption` commands, the `\subcaptionlabel` will not be saved in special box register when the float box is building, and will be typed like caption label, which follows settings of caption layout. This command is based on `\subcaption` command but with changed internal command of caption package. There is also the `\captionlabel` command.

```
⟨preamble⟩
  ...
  \floatsetup[figure]{style=plain}
⟨preamble⟩

{\begin{picture}(82,28)(0,0)
\put(0,0){\framebox(40,28)[bl]{}}
\put(2,2){\makebox(0,0)[bl]{\relax\hbox{\subcaptionlabel{}\Flabel{scap:I}}}}
...
\put(42,0){
\put(0,0){\framebox(40,28)[bl]{}}
\put(2,2){\makebox(0,0)[bl]{\hbox{\subcaptionlabel{}\Flabel{scap:II}}}}
...}
\end{picture}}
{\caption{Here are two simple subfigures.
\textit{Left} shows cat's eyes (\Fref{scap:I});
\textit{right}---cat's ears (\relax\Fref{scap:II})%
}\label{figcap:label}}
\end{figure}
```

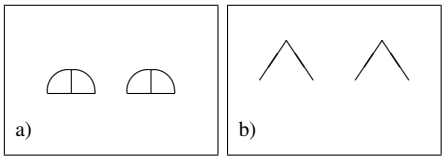


Figure 62. Here are two simple subfigures. *Left* shows cat's eyes (61, *a*); *right*—cat's ears (61, *b*)

6.2 Support of The Label–Sublabel References

In the examples above of the current section the `\Flabel` and `\Fref` commands were used for cross referencing (you may see these commands in the code examples). The `\Flabel` gets a modified format of current label of subfloat number: In these definitions the float and subfloat separators are divided by a special separator command, which by default has no effect. The label command `\Flabel` can be defined like following:

```
⟨preamble⟩
  \newseparatedlabel\Flabel{figure}{subfigure}
⟨preamble⟩
```

or, for all floats:

```
⟨preamble⟩
  \makeatletter
  \newseparatedlabel\Flabel{\@captype}{sub\@captype}
  \makeatother
⟨preamble⟩
```

Next command, `\Fref`, redefines this separator, and defines, if necessary, the font emphasize (or other command which uses one argument) of following part of label, and prints reference with standard `\ref` command. It was defined in this documentation like following:

```
⟨preamble⟩
  \newseparatedref\Fref{,\,\textit}  .
⟨preamble⟩
```

Thus, labels, which use `\Flabel` command can be referenced by usual way with `\ref` command and with `\Fref` command. The labels in current section and in the section, which describes the subfig package, use the `\Flabel`. You may see the result of this command in all `\Fref`erences to these parts of figures.

The last command, `\makelabelseparator`, defines label separator globally:

```
⟨preamble⟩
  \makelabelseparator{,\,\textit}  .
⟨preamble⟩
```

In this case both `\Fref` and `\ref` commands give the same result with `\Flabel`ed elements.

6.2.1 The `\RawCaption` with Parts of Figure

`\RawCaption` The example with usage of `\subcaption` and `\RawCaption` command. The layout of figure float is modified `BOXED` style. The idea behind this example is to place caption in

the free right lower corner of graphics. The \RawCaption allows to put the caption in necessary place without disturbing the float layout.

subfloatrow* The starred form loads settings for creation captions of float parts, but in this environment the \caption command restores its meaning. Thus, you need the \subcaption command for typesetting sub-captions. You may define it by yourself:

```
\newcommand*\subcaption{\captionsetup{subtype*}\caption}
```

or use the additional package called subcaption which on top of everything defines the \subcaption command.

⟨*preamble*⟩
```
\DeclareColorBox{framedfigure}{\fcolorbox{gray}{white}}
\floatsetup[figure]{style=BOXED,heightadjust=object,
    colorframeset=framedfigure,
    framestyle=colorbox,frameset={\fboxrule3pt\fboxsep8pt}}
\floatsetup[subfigure]{style=plain,capbesideposition={left,top},
    heightadjust=object}
```
⟨*preamble*⟩

```
\begin{figure}[H]
\ffigbox{}{\begin{subfloatrow*}
\fcapside[1.1\FBwidth]{\subcaption{...}\Flabel{...}...}{}
\fcapside[1.1\FBwidth]{\subcaption{...}\Flabel{...}...}{}%
\end{subfloatrow*}%
\renewlengthtocommand\settowidth\Mylen{\subfloatrowsep}\vskip\Mylen
\BottomFloatBoxes\floatsetup[subfigure]{heightadjust=none}
\begin{subfloatrow*}
\fcapside[1.1\FBwidth]{\subcaption{...}\Flabel{...}...}{}
\ffigbox[][][b]{}{\RawCaption{\caption[...}\label{...}}}
\end{subfloatrow*}}
\end{figure}
```

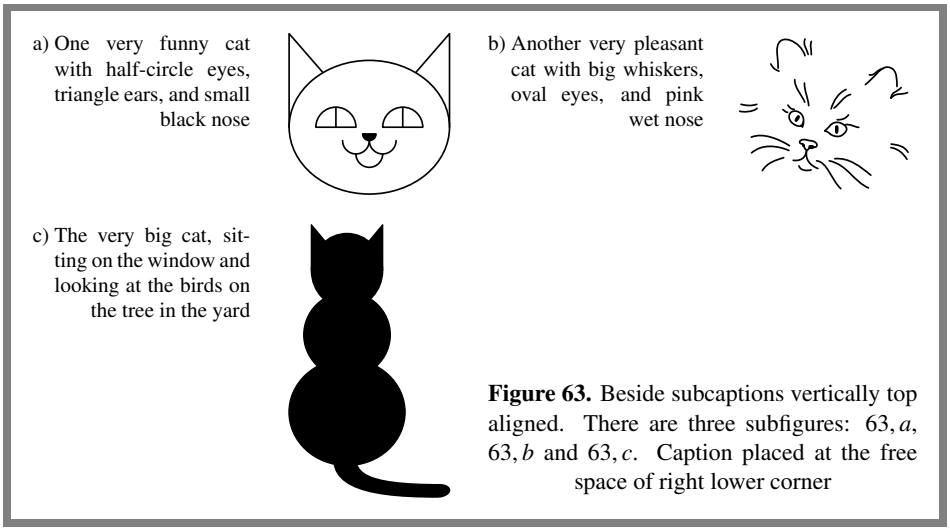


Figure 63. Beside subcaptions vertically top aligned. There are three subfigures: 63, *a*, 63, *b* and 63, *c*. Caption placed at the free space of right lower corner

7 Style Tandems

The next few sections show examples and explain some noticed features with usage of floatrow and other packages. There is no full list of style compatibilities. You may succeed with other versions of mentioned packages, and maybe with not mentioned packages too.

7.1 The subfig Package

Tested (and compatible) with version 1.3, dated 2005/06/28[1]. For the subfig package there are additional macros in floatrow which put subcaption label beside contents of subfloat and put alone subcaption label.

7.1.1 Additions in floatrow

Subcaption above subtable

The example with `\subfloat`'s (table 17). The setting command in preamble `\floatsetup[table]{style=Plaintop}` includes also settings for subcaption positions used with the subfig package (like `\captionsetup[table]{position=top}` in caption package):

```
\begin{table}\setlength\extrarowheight{1pt}
  \floatbox{table}[\FBwidth]
   {\caption{Two ...}\label{...}}
   {\begin{subfloatrow}
     \subfloat[First subtable]
      {\begin{tabular}{|l|c|c|}
        ...\end{tabular}}
     \subfloat[Second subtable...]
      {\begin{tabular}{|l|c|c|}
        ...\end{tabular}}%
    \end{subfloatrow}}
\end{table}
```

Table 17

Two `\subtable`'s (created with subfig package)

(a) First subtable

Column Head	Data	
	I	II
First row	1	2
Second row	3	4
Third row	6	8
Fourth row	10	16

(b) Second subtable with long long long subcaption

Column Head	Data	
	I	II
First row	1	2
Second row	3	4
Third row	6	8

[1] The English documentation is ⟨*texmf folder*⟩`/doc/latex/subfig/subfig.pdf`.

The `subfloatrow` is analogous to the `floatrow` environment. The usage is similar to `floatrow`:

```
\begin{subfloatrow}[⟨number of beside floats⟩]
\subfloat...
\subfloat...
...
\end{subfloatrow}
```

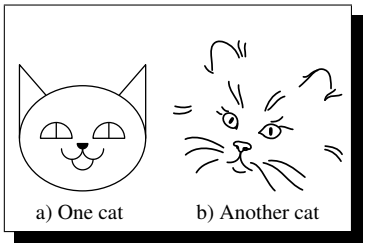
i.e. by default two subfloats are allowed. For other number of subfloats you ought to put number in optional argument. This environment puts a horizontal separator between subfloats, defined by `subfloatrowsep=` key instead of `floatrowsep=`. This key uses the same options as `floatrowsep=` and `capbesidesep=` keys (options of these keys defined by `\DeclareFloatSeparators` command, page 61).

Next follows an example with beside caption (see figure 67).

```
⟨preamble⟩
  \floatsetup[figure]{style=Shadowbox,capbesidesep=columnsep,
      capbesideframe=yes,capbesideposition={left,top}}
  \floatsetup[subfigure]{style=plain}
  \captionsetup[subfigure]{labelformat=brace,justification=centerlast,
        strut=no}
⟨preamble⟩

\fcapside[\FBwidth]
   {\begin{subfloatrow}
   \subfloat[...\label{...}]{...}
   \subfloat[...\label{...}]{...}
   \end{subfloatrow}}
{\caption{...}}
```

Figure 67. Beside caption vertically top aligned; fancy `Shadowbox` layout. There are two subfigures: 67, *a* and 67, *b*



`\sidesubfloat` Another addition in floatrow for subfloats is the command,which puts subcaption label beside subfloat. The subcaption label always appears on the left side. The key `subcapbesideposition` `subcapbesideposition=` sets vertical alignment of beside subcaption and subfloat. The options are analogous to the ones for `capbesideposition=` key:

`top`	subcaption label aligned to the top of object;
`bottom`	subcaption label aligned to the bottom of object;
`center`	subcaption label aligned to the center of float contents.

Subcaption beside subfloat The figure 71 shows layout with subfloat labels beside.

```
⟨preamble⟩
  ...
  \floatsetup[figure]{style=plain,subcapbesideposition=top}
⟨preamble⟩

\begin{figure}[H]
  \ffigbox[\FBwidth]
    {\begin{subfloatrow}
       \sidesubfloat[]{...\label{...}}%

       \sidesubfloat[]{...\label{...}}%
     \end{subfloatrow}}
{\caption[...]{...}\label{...}}
\end{figure}
```

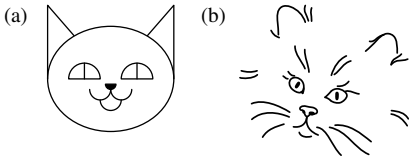


Figure 71. Beside caption vertically centered. There are two subfigures: 71, *a* and 71, *b*

`\subfloatlabel`

There are cases when usage of something like `\subfloat[]{\label{..}}` is needed. The first case shows the figure 72—the funny `picture` environment where subfloat labels were `\put` as a part of subfigures. Other—when you use mechanism of psfrag package and replace text entries from PostScript file with LaTeX ones. Unfortunately, the subfig package creates unnecessary spaces around alone subfloat label in the `\subfloat[]{\label{..}}` combination. The fr-subfig tries to fix this problem.

This command is based on `\subfloat[]{\label{..}}` sentence and puts alone subcaption label with necessary number. The full variant of `\subfloatlabel`

`\subfloatlabel[`⟨*subfloat number*⟩`][`⟨*label entry*⟩`]`

is the abbreviation of the following:

```
\setcounter{⟨sub\@captype⟩}{⟨subfloat number-1⟩}
\subfloat[]{\label{⟨label entry⟩}}
```

Another example:

```
⟨preamble⟩
  ...
  \floatsetup[figure]{style=plain}
⟨preamble⟩

\begin{figure}[h]
\fcapside[\FBwidth]
    {\unitlength2\unitlength\fboxsep-.4pt
       \begin{picture}(90,30)(0,0)
```

```
            \put(0,0){\framebox(40,30)[bl]{}}
            \put(2,2){\makebox(0,0)[bl]{\subfloat[]{\Flabel{subfig:wII}}}}
...
            \put(50,0){\framebox(40,30)[bl]{}}
            \put(52,2){\makebox(0,0)[bl]{\subfloatlabel[3][subfig:bII]{}}}%...
        \end{picture}}
{\caption{...}\label{...}}\end{figure}
```

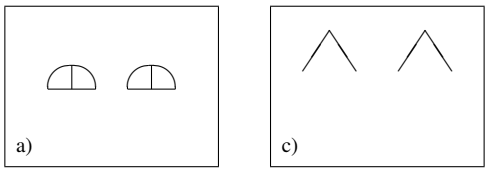


Figure 72. Here are two simple subfigures. Left one shows cat's eyes (72, *a*), labeled with `\subfloat[]{}` macro; with `\subfloatlabel[3][subfig:bII]` sentence were labeled the cat's ears (72, *c*)

In the examples of current section the `\Flabel` and `\Fref` commands for cross referencing of the subfloats were used (you may see these commands in the code examples). As described in section 6.2 these commands allow to create combined references which consist of the parent and current labels separated by predefined punctuation sign.

Some explanation. Previous versions of documentation used the `listofformat=` key; the necessary option was defined by `\DeclareCaptionListOfFormat` command:

```
\DeclareCaptionListOfFormat{comma-separated}{#1,\,#2}
```

This format is used, in particular, by `\subref` command. But usage of this key changes output of subfloat numbers in the lists (list of tables and list of figures etc.), which could be undesirable (see numbers of subfigures 72, a and 72, c in the List of Figures).

See examples with `subfloatrow` environments in sample files `frsample03.tex`, `frsample05.tex`; and also `frsample10.tex`–`frsample12.tex` where aligned contents of beside subfloats are used in different layouts.

7.2 The longtable Package

Tested with version v4.11, dated 2004/02/01.[1]

Please note that almost all settings in the `\floatsetup`'s argument do not work inside `longtable` environments, except settings for caption width (see below) and plain horizontal alignment in the `margins=` key. So, during building of `\floatsetup` settings for the tables, be aware that you may use only something like `style=plaintop` or `style=Plaintop`, to place caption above, also you may use options of the `margins=` key, which use only spacing commands, like defined ones in this package (page 43), and do not forget settings for `\LTleft` and `\LTright` margins, which set the alignment of `longtable` environment.

Please see the caption documentation about how to build necessary caption layout when `longtable` environment is used.

7.2.1 Additions in The floatrow Package

A patch was added to the longtable package[2]: this patch adds the same font settings as for `table` environments, and adds code which helps to get the width of `longtable` caption equal to the width of table. For settings of the caption width the special key was created.

`LTcapwidth`
Caption width equals to longtable's

This key could have any value, like `5cm` or `\hsize`. The key value will be sent to the `\LTcapwidth` command. If you'll write `LTcapwidth=table` or `LTcapwidth=contents`, you will get the caption width equal to the width of table. In this case settings for width of caption use information from the `aux`-file, so you'll get correct caption width at the time when the width of full table *become stable*.

The `longtable` environment uses layout settings from `\floatsetup[table]` and `\floatsetup[longtable]` contents. The `\floatsetup[longtable]` will be "strongest" in this pair.

The addition with version 0.1k. A beta-temp[3] package fr-longtable with additions is added, which allows creation of special head for the last page of longtable environment and special foot for pages before last (the table 11 uses these commands for head and foot settings).

`\endlasthead`
`\endprelastfoot`

The `\endlasthead` command defined for last head of longtable; second command, `\endprelastfoot`, defined for foot on the page before last. Since these names of commands "intrude" in the longtable naming territory they get defined if they are still unknown, i.e. the main, longtable, package didn't defined them. The syntax is also analogous as for commands `\endhead`, `\endfirsthead` etc. (See examples and additional explanation in the sample file `sample-longtable.tex` file.)

Note. Please remember that the footnote stuff inside `longtable` works like in main text and puts the text of footnotes at the bottom of page[4].

[1] The English documentation is ⟨*texmf folder*⟩`/doc/latex/tools/longtable.dvi`.

[2] Thanks to A. Sommerfeldt for help to make this code compact.

[3] Again, like with listpen package, I hope that such support sooner or later could appear in longtable and think it is better to follow grammar of master-package for similar situations. Also it is necessary to say that command names from fr-longtable package "intrude" in the longtable's naming space.

[4] See also `longtable` documentation.

The floatrow package's command for legends or explications, \floatfoot, in current version has emulation mode inside longtable, and needs stuff, similar to \noalign{\floatfoot{...}}. Since the default font definition for explications (\floatfoot) is also set to \footnotesize, like for footnotes, you may put footnotes-emulations at the end of table, inside this explication block, using \mpfootnotemark commands inside table contents and at the beginning of each text of footnote.

The fragments from the longtable 11 on the page 31, which describes float styles, will be the resumé for this section.

```
⟨preamble⟩
  \DeclareCaptionLabelFormat{continued}{\rightline
                {\bothIfFirst{#1}{ }#2 (\emph{Continued})}}
  \DeclareCaptionLabelFormat{finished}{\rightline
                {\bothIfFirst{#1}{ }#2 (\emph{Finished})}}
⟨preamble⟩

\def\LongtableHead{
   \hfil\thead{Style} &
   \hfil\thead{\cmd{\floatsetup} keys} &
   \hfil\thead{Description}
   }
\begin{longtable}{⟨tabular preamble⟩}
\caption{Float layout styles}\label{tab:floatlayouts}\\
\hline
\LongtableHead
\\ \hline
\endfirsthead% end of standard box of longtable package
\captionsetup{labelformat=continued}%   caption settings for continued page
\caption[]{}\\
\hline
\LongtableHead
\\ \hline
\endhead% end of standard box of longtable package
\captionsetup{labelformat=finished}%   caption settings for finished page
\caption[]{}\\
\hline
\LongtableHead
\\ \hline
\endlasthead% end of box offered by fr-longtable package
\hline
\multicolumn{3}{r@{}}{\topstrut\emph{Continued on next page}}
\endfoot% end of standard box of longtable package
\hline
\multicolumn{3}{r@{}}{\topstrut\emph{Finished on next page}}
\endprelastfoot% end of box offered by fr-longtable package
\endlastfoot% end of standard box of longtable package
⟨Contents of long table⟩
⟨Contents of long table⟩\mpfootnotemark[1]
⟨Contents of long table⟩
\\ \hline
```

```
\noalign{\floatfoot*{⟨Text of foot material⟩.\vspace{-3pt}\par
\rule{1in}{.4pt}\vspace{2pt}% Emulation of footnote rule
\parindent15pt
% emulations of footnote texts
\mpfootnotemark[1]⟨Text of footnote⟩
...
}}
\end{longtable}
```

Note. The usage of settings `\captionsetup{labelformat=continued}` inside `longtable` environment was documented in the caption package 3.1.

7.3 The wrapfig Package

Fig. 73 Wrapped plain figure (wrapfig package)

Plain figure fails with package version 3.3

Tested with version 3.3 dated 1999/10/12 (style from ltxmisc bundle) and 3.6 dated 2003/01/31 (the separate LaTeX package)[1].

Options for environment (text borrowed from package comments):

```
\begin{wrapfigure}%
  [⟨number⟩]{⟨placement⟩}%
  [⟨overhang⟩]{⟨width of figure⟩}
...
\end{wrapfigure}
```

⟨Placement⟩ is one of `r`, `l`, `i`, `o`, `R`, `L`, `I`, `O`, *for right, left, inside, outside. Lowercase letters set unfloated positioning, uppercase—floated variant. The figure sticks into the margin by ⟨overhang⟩, if given, or by the length* `\wrapoverhang`*, which is normally zero. The ⟨number⟩ of wrapped text lines is normally calculated from the height of the figure, but may be specified manually, e.g.*

```
\begin{wrapfigure}[10]{r}[34pt]{5cm}
⟨figure⟩
\end{wrapfigure}
```

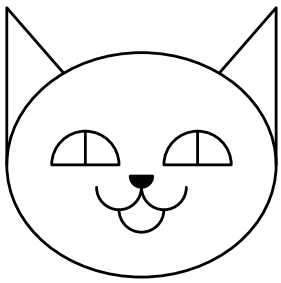


Fig. 74. Wrapped figure in `\ffigbox` (wrapfig package)

Notes. 1) For figure, contents in e.g. in `wrapfigure` environment you set width in mandatory argument. If you'll write `0mm` as {*⟨width of figure⟩*} argument, the wrapfig package will calculate a natural width of float contents. If you use the `\floatbox` command, put `\FBwidth` option to use natural object width.

2) Sometimes above (below) float box in `wrap...` environment appears unwanted space. To correct vertical position, use `\FBaskip` (`\FBbskip`) commands (see section 2.7) and optional argument *⟨number⟩* of `wrap...` environments.

3) Please note that the label of wrapped floats changed to 'Fig. *⟨number⟩*'. This happened because of the following settings:

```
⟨preamble⟩
  \DeclareCaptionLabelFormat{thinspace}{\bothIfFirst{#1}{\,}#2}
⟨preamble⟩

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

In preamble was added special format `thinspace` with smallest space between 'Fig.' and number which we use in the `wrapfig` settings. See also caption documentation.

Special settings.

[1] The English documentation is *⟨texmf folder⟩*`/doc/latex/wrapfig/wrapfig.pdf`.

You may create settings for `wrap...` environment, there are following priorities. (Please note that you can also create special caption settings with `\captionsetup` stuff.):

- if exists `\floatsetup[wrap`⟨*captype*⟩`]{...}` floatrow uses these settings—they are the "strongest" settings; if they are absent—uses settings of next item;
- if exists `\floatsetup[wrapfloat]{...}` floatrow uses these settings—these settings are "stronger" than next ones; if they are absent—settings of current float

  `\floatsetup[`⟨*captype*⟩`]{...}`;

  if they are absent—uses `\floatsetup{...}` settings, package settings inside `\usepackage` command or default settings of package (page 53).

Founded limitations.

1) The usage of plain floating environment in version 3.3 will not succeed with floatrow—use `\floatbox` stuff. The version 3.6 allows usage of plain `wrap...` environment with `plain` (or `ruled`) styles, but the framed styles, like `Boxed` (which use key `framefit=yes`, where text inside frames changes its `\hsize` to fit frames, fitted to defined `\hsize`) could work only with `\floatbox` macro, otherwise you'll get incorrect widths and layout.

2) The `wrap...` environments could fail inside list ones. You ought be careful with grouping around wrapping environment (float can sail away or disappear). Tests show that you may set `wrap...` environment at the very beginning of list, in the case of you created faked or empty paragraph just before list (i.e. between `wrap...` and list) with compensate negative spacing, like following: `\noindent\strut\par\nobreak \vskip-\baselineskip`.

7.4 The floatflt package

Tested with version v1.3 dated 1996/02/27.

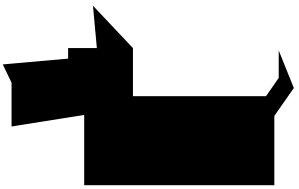


Fig. 75. Wrapped figure inside floatingfigure environment (floatflt)

Founded limitations. 1) There is not support for creation of new `floating...` environment. Since `floatflt` environments need usage of `\floatbox` in any case, you can use either `floatingfigure` or `floatingtable` and put necessary float type in `\floatbox` argument (or use necessary macro abbreviation, like `\ffigbox`). For these wrapped floats the `\usepackage` option can be used or `\floatsetup{...}` settings and main settings for float types like `\floatsetup[figure]{...}` settings.

The next limitations could not tied with floatrow package.

2) If you put a `floatingfigure` environment just after `\...section` command you need (if you do not indentation after heads) to put `\noindent` for the first paragraph.

3) The `floatflt` environments could fail with list environments.

4) The special caption settings were created for figure label.

```
\captionsetup[floatingfigure]{name=Fig.,labelformat=thinspace}
```

7.5 The picins Package

Tested with version v 3.0 dated 1999/10/12.

This package produces pictures inside paragraphs. This package supports usage of captions with command `\piccaption`. It also allows the caption package settings.

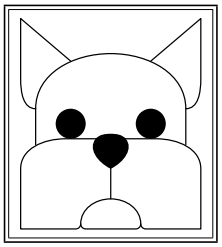


Fig. 76. Wrapped figure (`\parpic`)

The `\parpic` macro usually allows usage of `\floatbox` macro inside of its mandatory argument. In this case the `\floatsetup{...}` settings and main settings of for float types like `\floatsetup[figure]{...}` settings are used (but, unfortunately, they are the only here).

Founded limitations.

1) In `\parpic` argument you ought to to define the width of contents. If you put `\hsize0pt` before the `\floatbox` command, you will get box width equals to `\parpic` contents. (Compare with usage of `0mm` value inside the {⟨*width of figure*⟩} option in the `wrapfigure` environment.)

The next limitations could not tied with floatrow package.

2) If you put `\parpic` just after `\...section` command you need (if you do not indentation after heads) to put `\noindent` for the first paragraph.

3) It seems that the `\parpic` command cancels nonbreaking mechanism between section command and text in the case of appearance at the very beginning of the first paragraph (this situation appeared during testing of current documentation).

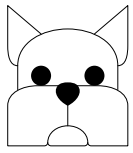


Figure 77. Wrapped figure (`\parpic`)

4) You may try to use `\parpic` inside list environment, but sometimes usage of this command in this environment could create wrong layout. (Tests show that paragraph(s) where the `\parpic` is used must be placed in group—compare it with the wrapfig package, which does not like grouping.)

5) This package has not options ⟨*outside*⟩ or ⟨*inside*⟩, like previous two packages (the option `[o]` means oval box around picture), so you ought to set horizontal position manually. Or you may create command:

```
⟨preamble⟩
  \usepackage{ifthen}
  \newcommand\oparpic{\ifthenelse{\isodd{\value{page}}}%
        {\def\next{\parpic[r]}}{\def\next{\parpic[l]}}\next}
⟨preamble⟩
```

6) The special caption settings were created for figure label

```
\captionsetup[parpic]{name=Fig.,labelformat=thinspace}
```

If you use `\piccaption` command these settings are switched on. In the first picture in this section the `\piccaption` co-operates with the `\ffigbox` command:

```
\piccaption{...\label{...}}%
\parpic[l]{\hsize0pt\ffigbox[\FBwidth]{}{...}}
```

Second picture uses the \caption command inside \ffigbox, so the \captionsetup [parpic]{...} settings do not work:

```
\parpic[r]{\hsize36mm\def\FBaskip{6pt}
   \ffigbox[\hsize]{}{...\caption{...}\label{fig:parpic:BcatII}}
```

You may see that label of the second figure was printed as 'Figure' number.

7.6 The rotating Package and sideways... Environment

Tested with version v2.13 dated Sep. 1992.

There is example (figure 78) with rotated float, using sidewaysfigure.

⟨*preamble*⟩
```
  \usepackage[figuresright]{rotating}
  \floatsetup[rotfigure]{style=WSHADOWBOX}
```
⟨*preamble*⟩

```
\begin{sidewaysfigure}\emptyfloatpage
\ffigbox[\FBwidth]
  {...}
  {\caption{Figure ...}%
  \label{...}}
\end{sidewaysfigure}%
```

Special settings.

You may create special settings for all rotated floats, which use sideways... environment (see page 29).

For one-column rotated float

- if exists \floatsetup[rot⟨*captype*⟩]{...} package uses these settings—the "strongest" settings; if they are absent—uses settings from next item, the same for each item of the list;
- \floatsetup[rotfloat]{...};
- \floatsetup[⟨*captype*⟩]{...};
- if all settings absent—the settings inside \floatsetup{...} and \usepackage commands, and, at last, package default settings are used.

For two-column or wide rotated float (starred environment)

- if exists \floatsetup[widerot⟨*captype*⟩]{...} package uses these settings—the "strongest" settings; if they are absent—uses settings of next item, the same for each item of the list;
- \floatsetup[widerotfloat]{...};
- \floatsetup[rot⟨*captype*⟩]{...};
- \floatsetup[rotfloat]{...};
- \floatsetup{⟨*captype*⟩}{...};
- if all settings absent—the settings inside \floatsetup{...} and \usepackage commands, and, at last, the package default settings are used.

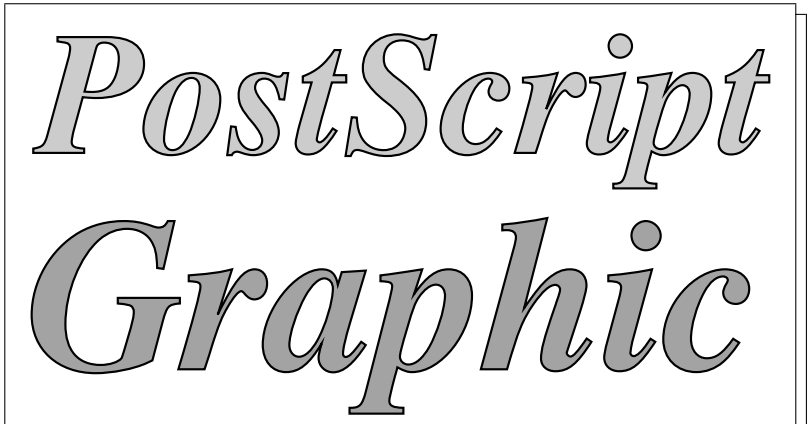


Figure 78. Figure inside `sidewaysfigure` environment

7.6.1 Special Page Style for Float Page

Empty page style for rotated floats

In example with figure 78 you may see the command \emptyfloatpage. It is offered by floatpagestyle package, (installed with floatrow package, can be used separately). The macro \emptyfloatpage is an abbreviation of \floatpagestyle{empty}. The last macro redefines the page style for the page where *current* floating environment appears in the way, analogous to \thispagestyle command.

The version 0.1h patches the core LaTeX macro \@outputpage[1] and I hope that it could work.[2] Since this package uses \label—\ref mechanism, the \floatpagestyle command works after *second* LaTeX run.

7.6.2 Rotated Floats on the Facing Pages

Continued rotated floats

1) If you place two continued rotated floats on facing pages, the better way is to gather them to binder margin, using \buildFBBOX command (see page 21). For this reason you may define

```
⟨preamble⟩
  \usepackage[figuresright]{rotating}
  \newlengthtocommand\setlength\rottextwidth{\textwidth}
⟨preamble⟩

\begin{sidewaysfigure}
\buildFBBOX{\vbox to\rottextwidth\bgroup\vss}{\egroup}
\ffigbox{}{⟨contents of first figure⟩}
\end{sidewaysfigure}
\begin{sidewaysfigure}
\buildFBBOX{\vbox to\rottextwidth\bgroup}{\vss\egroup}
\ffigbox{}{⟨contents of second figure⟩}
\end{sidewaysfigure}
```

2) In the example above (and also in the example with figure 78) the rotating package has [figuresright] option; in this case all sideways... floats on even and odd pages will be rotated by 90° counterclockwise.

7.6.3 Commands instead of lengths

The \rottextwidth command in the example above stores value of the \textwidth of the document; the \columnwidth and \textwidth inside sideways... environment are redefined and equal to \textheight. If a) you are limited in creation of the new length or dimension command (for example you use the pictex package[3]), or b) the width/height or the space values, defined with the \newcommand (like the \headrulewidth command from fancyhdr package) need complex calculation with us-

age of the calc package, or get the width of some text—the floatrow package provides commands

\newlengthtocommand
\renewlengthtocommand

```
\newlengthtocommand   or
\renewlengthtocommand
```

which are placed just before standard LaTeX commands like `\setlength` or `\settowidth` and save the *absolute* value from their arguments; here the usual code like

```
⟨preamble⟩
  \usepackage{calc}
⟨preamble⟩

\newlength\fulltextwidth
\setlength\rottextwidth{\textwidth+\marginparsep+\marginparwidth}
```

changed to

```
⟨preamble⟩
  \usepackage{calc}
⟨preamble⟩

\newlengthtocommand\setlength
\fulltextwidth{\textwidth+\marginparsep+\marginparwidth}  .
```

Please note than the usage of calculation inside `\setlength` command (and its analogs) can be used only with the calc package.

7.7 The lscape Package and `landscape` Environment

Tested with version v3.0a dated 1999/02/16.

The example with usage of `landscape` environment from lscape package on the page 92, figures 79–82):

```
⟨preamble⟩
  \DeclareFloatVCode{lowthickrule}{\kern2pt\rule{\hsize}{.8pt}}
  \floatsetup[figure]{style=ruled,rowprecode=thickrule,
    rowpostcode=lowthickrule,capposition=TOP}
⟨preamble⟩

\begin{landscape}
\begin{figure}\emptyfloatpage
...
```

`\floatsetup` code sets `ruled` float style, then settings for above and below material are redefined: `rowprecode=` and `rowpostcode=` keys define thick rules but for floatrow as a whole (the 'individual' `\hrule`'s above/below float boxes are absent).

[1] At the start of document floatpagestyle package puts additional code at the very beginning of this output routine.

[2] If you know more honest way to get the same result—the redefinition of *alone float* page style (in the case when this page can *float* inside document)—please let me know.

[3] The e-TeX engine could solve this problem.

The `landscape` environment creates a new page. It would be useful 1) for rotation of multipage rotated float (in this case it is better to put this float in a separate file, and to start from necessary page, in this case you need the afterpage package and its `\afterpage` command) 2) and also to start new section of document, e.g., appendix. (In current document the `landscape` environment was placed just before appendix)

7.8 The listings Package

Tested with version v1.3 dated 2004/09/07.

This package has its own strong layout mechanism for creation of floating algorithms itself. The usage of `\lstset` command (see package documentation) and caption package settings gives you necessary result[1] for algorithm type of float.

For the cases of appearance of listings inside of other float environments, which get settings from floatrow package, there is a limitation: you can't put `lstlisting` inside `\floatbox` contents. The plain float environment is still allowed. Also you are still free with settings for float type, used `lstlisting` inside: you may still use the `BOXED`, `Boxed` and other unusual styles: the float width will be recalculated for mentioned two styles and similar ones and then will be used necessary setting. If you need to change box width—use `\thisfloatsetup` settings.

7.9 The hyperref and hypcap Packages

There were tested versions v6.77i (hyperref) and v1.7 (hypcap).

The floatrow package tries not to expand its code to `\caption` stuff. I hope that environments supported by floatrow won't make harm to caption—hyperref/hypcap tandem.

7.10 The setspace Package

There was bug during usage of setspace package—this package redefines font size to `\normalsize`. The version 0.2d of floatrow tries to fix it. The default stretch is equal to 1. The version 3.1 of caption package offers special font settings (see caption documentation) for captions. You may try the same for the float font:

```
\floatsetup{font=onehalfspacing}
```

or

```
\floatsetup{font={stretch=⟨amount⟩}}
```

.

8 The Incompatibilities

At first the incompatibilities or rules of co-operation with other packages could follow the caption 3.x package. *Please look first in the caption package documentation to know the newest rules.*

[1] Please note and read caption documentation: the co-operation of caption3.x and listings succeeds with version of last one not older than 1.2.

The known incompatibilities of floatrow package itself: 1) sidecap package[1]: the floatrow package doesn't expands its layouts to `SCfigure` and `SCtable` environments; 2) ctable package; if you used to use ctable's tools, e.g. for tables, please set `\RawFloats[table]` in the preamble, and remember that commands like `\ttabbox` won't loose its strength (see also section 2.4).

9 Limitations

There are known limitations, which were found during usage of floatrow:

- You cannot use `\floatbox` stuff for floats with `verbatim` environment and/or `\verb`. But you still can use plain float environments. If you need to change width of float box, you may change it with `\thisfloatsetup` settings. The usage of `verbatim` and `\verb` do not create limitations for layout: you may still use the `BOXED`, `Boxed` and other unusual styles: the float width will be recalculated for mentioned two styles and similar ones and then will be used necessary setting.
- The `tabbing` environment in current version creates incorrect layout for float box which must occupy whole text width: it recalculates the width of object box to the natural width of its contents. The problem will be solved with the `minipage` environment and width option `\hsize`: you'll get necessary layout with full width and for the styles like `BOXED` and `Boxed` the width of contents will be recalculated.
- Be careful with minipages inside `floatrow` environment—there could be wrong alignment. Use `heightadjust=` key for this case. (Fortunately I cannot imagine good readability of two beside `tabbing`s.)
- This limitation was mentioned above: some tools of the package use `\label`—`\ref` mechanism, thus, if you use float layout which demands common height of objects and/or captions in float row, you'll get correct result after second or more runs. If you change contents of float which change its height you must run LaTeX twice or more times too.

  Beside captions and other facing layout will appears correctly only after second LaTeX's run (sometimes you need to run more times).
- The caption and floatrow packages do not support an optional argument *after* caption "title" (the float package's stuff). You may use `\floatfoot` macro after main caption argument.
- Do not use the `\FBwidth` option for complex float contents (which you could not put inside one `\hbox`). But you are allowed to use `\vspace` macro at the very end/very beginning of object contents for fine vertical tuning for them.
- The `floatrow` environment allows spaces (and even empty lines, which sometimes create better and correct result!) between `\floatbox`'es, but if you add some code between them you must put `%` after this command.

[1] Despite that I'm trying to follow all offered layouts of this package. Great thanks for Rolf Niepraschk and Hubert Gäßlein for package with rich implementation of such float layouts.

- This is a common rule—be careful with spaces at the end of lines inside float contents (see `CTAN:/info/epslatex.ps` for more explanations).

  When you build plain floating environments the better way is to separate `\caption` and object contents (and also `\floatfoot/\footnotetext` contents) each by empty lines or (if not empty lines) end each part (and arguments of mentioned commands) by percent sign. In this case you'll avoid unwanted spaces/lines at the end of contents of each part, or wrong justification of float components.

- If you use `tabularx` or `tabular*` environments inside `\floatbox` stuff (or any other) with `\hsize` command inside ⟨*width*⟩ argument, you must repeat the `\hsize` argument in ⟨*width*⟩ argument of `\floatbox` macro.

  If you want to set width of `tabularx` or `tabular*` environments (or any other) like `.8\hsize` (or `1.2\hsize`) and these environments placed inside any `\floatbox` macro, load `.8\hsize` in ⟨*width*⟩ argument of `\floatbox` macro, and in ⟨*width*⟩ argument of `tabularx` or `tabular*` load only `\hsize` macro (see also sample file `frsample03.tex`).

  In other cases (especially in fancy layout or settings) be careful with usage of `\hsize` as ⟨*width*⟩ option of `\floatbox`.

10 Acknowledgements

Thanks for Steven Cochran and Axel Sommerfeldt for all their advices and spirit. Special thanks for Axel for the patient answering, code, finding and showing bugs, and help in *all* my questions and problems in floatrow package. All good text pieces in this documentation are filled with Axel's advices and great help.

Thanks for *all* involuntary (LA)TEX teachers, who teaches me with their program code all these years.

Thanks for Keith Reckdahl, author of `epslatex`, which documentation, at last, encouraged me to create the CTAN version of this package.

Thanks for all *authors of second edition of LaTeX Companion for this book.*

Figure 79 Figure in the row I, top of object box[a]

[a]This picture was created with `\qbezier` macros

Figure 80 Figure in the row II, bottom of object box

There are all `\qbezier` macros and only two vertical lines

[b]Look at funny footnotemark!

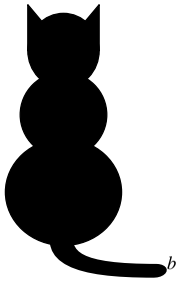


Figure 81 Figure in the row III, center of object box

The image of cat

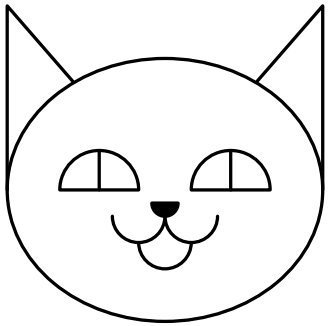
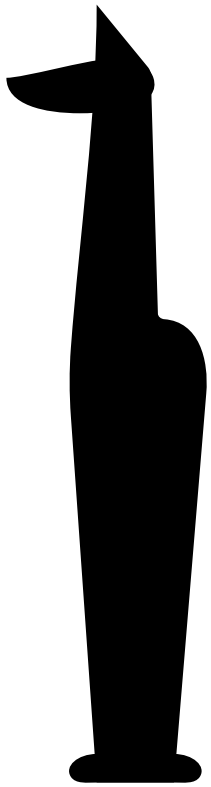


Figure 82 Figure in the row IV



11 Appendix

11.1 Miscellaneous

11.1.1 Usage of Captionsetup and Thisfloatsetup Inside Floatbox Stuff

Example of figures in row (figures 83 and 84). There predefined float commands `\fcapsideleft` and `\fcapsideright` with were used additional `\captionsetup` and `\thisfloatsetup` settings:

```
⟨preamble⟩
  \newfloatcommand{fcapsideleft}{figure}[{\capbeside
    \captionsetup[capbesidefigure]{labelsep=newline,
     justification=raggedleft}%
     \thisfloatsetup{capbesideposition=left}}][\FBwidth]
  \newfloatcommand{fcapsideright}{figure}[{\capbeside
    \captionsetup[capbesidefigure]{labelsep=newline,
     justification=raggedright}%
     \thisfloatsetup{capbesideposition=right}}][\FBwidth]
  \floatsetup[figure]
   {style=Boxed,objectset=centering,margins=centering,
    capposition=beside,capbesidesep=cicero,capbesideframe=yes}
⟨preamble⟩

\begin{figure}
\begin{floatrow}
  \fcapsideleft{...}{...}
  \fcapsideright[\hsize]{...}{...}
\end{floatrow}
\end{figure}
```

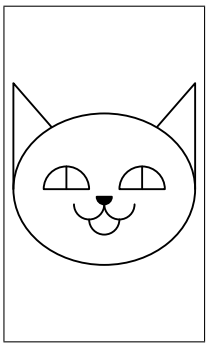


Figure 83
Float box (`\fcapsideleft`) width of graphics

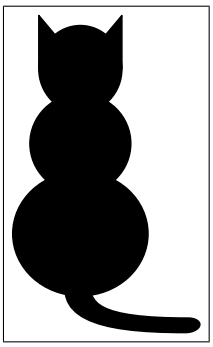


Figure 84
Float box (`\fcapsideright`) width of rest float row space

Since the key `heightadjust=object` is used in the `Boxed` float style, both objects have the same height.

11.1.2 Predefined Beside Caption Width

This example includes the `\useFCwidth` command which switches on usage of previously defined caption width with `capbesidewidth=` key (in command `\thisfloatsetup`

before \floatbox macro) or, if you didn't set caption width (like in current example), macro calculates natural width of caption contents (see figure 85). In this case the object—caption box is aligned using alignment settings from margins key (its options are defined by \setfloatmargins or \floatcapbesidemargins macro). In this documentation they are centered (see page 59).

⟨*preamble*⟩
```
  \floatsetup[figure]{style=plain}
```
⟨*preamble*⟩

```
\begin{figure}
\floatbox[\capbeside\useFCwidth]{figure}[\FBwidth]
 ...
\end{figure}
```

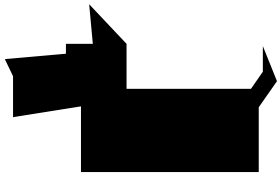


Figure 85

Please note that inside \floatbox you may not set predefined width of caption, but remember that you *must* define width of caption in case of usage of plain floating environment.

11.1.3 Predefined Beside Caption Width with The Rest Space for Object

The figure 86 uses the following float style:

⟨*preamble*⟩
```
  \renewlengthtocommand\settowidth\Mylen{\captionfont\captionlabelfont
     \figurename\ \thefigure}
```
⟨*preamble*⟩

```
\floatsetup[figure]
 {style=Boxed,capposition=beside,objectset=centering,
  capbesidewidth=\Mylen,capbesideposition=left,capbesidesep=cicero,
  margins=centering,capbesideframe=yes,
  floatwidth=sidefil}
```

The \Mylen dimension was defined as width of caption label.

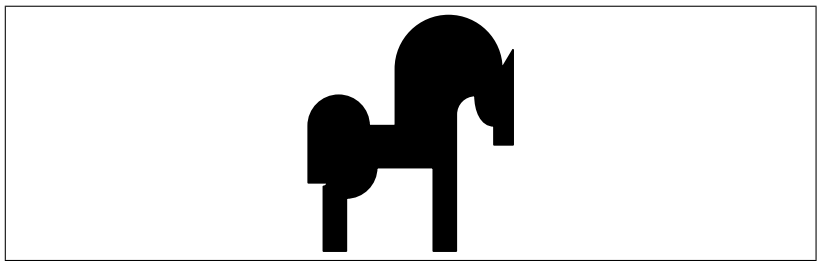


Figure 86

11.1.4 Width Definition for Beside Caption—Object Box in Float Row

The float row with predefined width boxes "beside object—caption" (figures 87 and 88): just define before `\fcapside` command something like:

```
⟨preamble⟩
  \floatsetup[figure]
   {style=plain,objectset=centering,margins=centering,
    capbesideposition=left,capbesidesep=enskip,
    floatwidth=sidefil}
⟨preamble⟩

\begin{figure}\useFCwidth
 \begin{floatrow}
   \setlength\hsize{1.2\hsize}%
   \fcapside...
   \setlength\hsize\Xhsize
   \fcapside...
 \end{floatrow}
\end{figure}
```

(please remember that option of `\fcapside` command defines the width of object contents but not the full box object—caption).

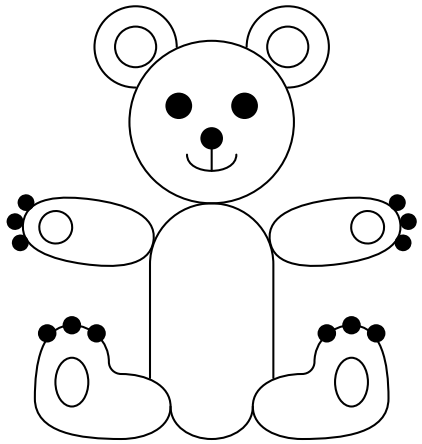


Figure 87

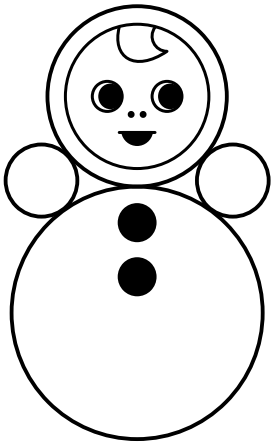


Figure 88

11.1.5 Caption Above/Below and Caption Beside at The Float Row

The float row with object and beside caption combined with object and caption below (figures 89 and 90). There we ought to use `\TopFloatBoxes`, `\CenterFloatBoxes`, or `\BottomFloatBoxes` commands to get correct layout—since the ⟨*height*⟩ argument in both float boxes has the same value, you may use each of these three commands. Unfortunately you must set the height of such beside floats by hand (the `heightadjust=` key works here incorrectly). The lines which create the described float row:

```
⟨preamble⟩
  \floatsetup[figure]
   {style=Boxed,frameset={\fboxsep4pt},captionskip=5pt,
```

```
    capposition=bottom,objectset=centering,capbesidewidth=sidefil,
    capbesideposition=inside,capbesidesep=enskip,margins=centering,
    capbesideframe=yes}
⟨preamble⟩

\begin{figure}\CenterFloatBoxes
\begin{floatrow}
\hsize1.098\hsize
    \fcapside[\FBwidth][3.6cm]
      ...

    \ffigbox[\Xhsize][3.6cm]
      ...
\end{floatrow}%
\end{figure}
```

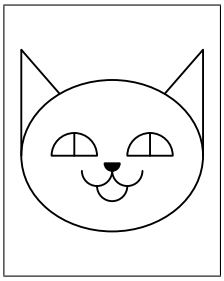


Figure 89. Float box (`\fcapside`) with beside caption in float row width float with caption below

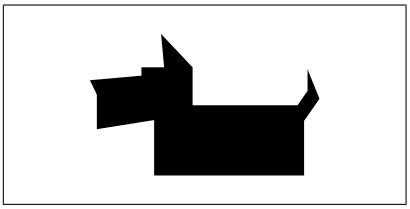


Figure 90. Float box (`\ffigbox`) width of rest float row space

The code for "mirror" layout (but not identical) looks like:

```
\begin{figure}\CenterFloatBoxes
\begin{floatrow}
   \ffigbox[1.28\FBwidth][3.6cm]
      ...

   \hsize\Xhsize
   \fcapside[\FBwidth][3.6cm]
      ...
\end{floatrow}%
\end{figure}
```

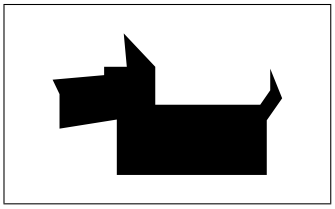


Figure 91. Float box (`\ffigbox`) in mirror float row

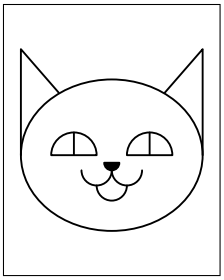


Figure 92. Float box with beside caption (`\fcapside`) in mirror float row width float with caption below

11.1.6 Photo-Album-Like Layouts

Another example of miscellaneous float row (figures 93–95, and, “mirror layout”—96–98) were created by following lines:

```
\begin{figure}\BottomFloatBoxes
\begin{floatrow}
\hsize1.2\hsize \ffigbox[][6.7cm]
  ...

\vbox to6.7cm
 {\floatsetup[figure]{floatrowsep=none}\killfloatstyle
    \ffigbox[.8\hsize]
      ...
    \vss
    \ffigbox[.8\hsize]
      ...%
 }%
\end{floatrow}%
\end{figure}
```

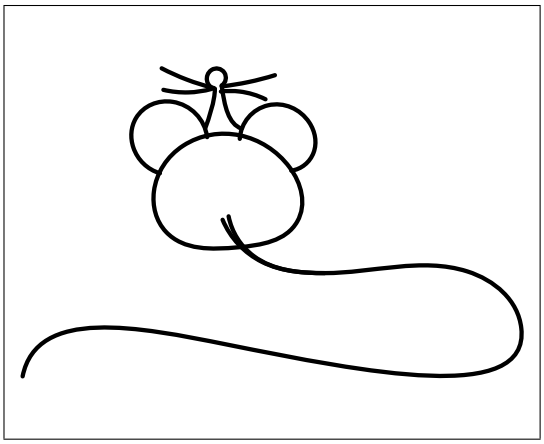


Figure 93. Float box in photo-album-like layout: alone in left column

Figure 94. Float box in photo-album-like layout: upper float in right column

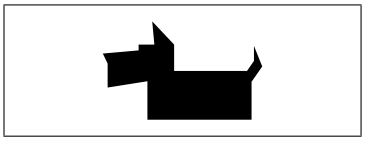


Figure 95. Lower float in right column

The “mirror” layout created by following commands:

```
\begin{figure}[t]\TopFloatBoxes
\begin{floatrow}
\vtop to7cm
 {\floatsetup[figure]{floatrowsep=none}\killfloatstyle
    \ffigbox[.8\hsize]
      ...
    \vss
    \ffigbox[.8\hsize]
      ...%
```

Figure 96. Float box in photo-album-like layout: upper float in left column

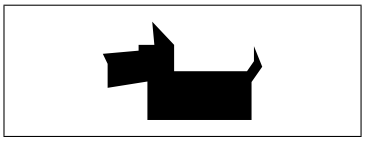


Figure 97. Float box in photo-album-like layout: lower float in the left column

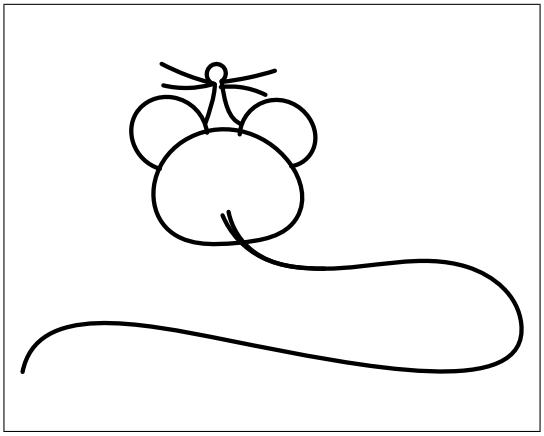


Figure 98. Float box in photo-album-like layout: alone in right column

```
 \vskip0pt}\floatrowsep

\ffigbox[\Xhsize][7cm-11pt]
  ...
\end{floatrow}%
\end{figure}
```

Note that in second example with "mirror" layout the trick with ⟨*height*⟩ definition was used—caption of float in the left column is one line longer, so for the right column height of float was reduced by 11pt—`\baselineskip` for `\small` size (here the calc package possibilities were used). The `\vtop` of left column ends with `\vskip0pt`, otherwise you get fanny unwanted layout.

In both examples for two floats one above another was cancelled `\floatrowsep` code inside `\vbox`/`\vtop`.

Note that these examples are rather specific—you may try with other combinations (e.g. more-"columned"), but maybe these layouts need more care with usage of `\Xhsize` and/or `\floatrowsep`.

I suppose that last two examples could conflict with "motto" of this package—to reduce and remove layout code from document; but photo-album-like layout is rather rare in technical literature (It isn't?).

11.1.7 Photo-Album-Like Layouts: Common Height for Beside Photos in Filled Row

This section shows example which allows to set common height for rectangular graphics, i.e. photos and fill full width of this row. To emulate the rectangular photos here, each graphic was loaded inside `\fbox` with zeroed `\fboxsep`. (See also file `frsample06.tex`.)

The code of example uses the \includegraphics command (graphicx package). You load the \CommonHeightRow command:

\CommonHeightRow[⟨*supposed height*⟩]{⟨*floatrow environment*⟩}

with supposed value of height in the optional argument, which could be near the necessary common height. The default value is controlled by the \DefaultCommonHeight command. It was defined

```
\newcommand\DefaultCommonHeight{25pt}
```

It seems that \DefaultCommonHeight could differ from one documentation to another, but inside one documentation the value in this command which once succeed in the row will gives the same correct result in other rows also.

The second argument—the contents of the floatrow environment. *All* float boxes in this row must use the [\FBwidth] option.

```
⟨preamble⟩
  \usepackage{graphicx}
  \floatsetup[figure]{style=plain}\floatsetup[widefloat]{margins=hangleft}
⟨preamble⟩

\begin{figure*}\fboxsep-.4pt
\CommonHeightRow{\begin{floatrow}[4]
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\end{floatrow}}
\end{figure*}%
```

Here you may see the result.

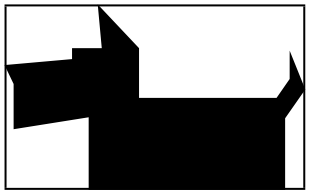


Figure 99. Figure I in the row with common heights

Figure 100. Figure II in the row with common heights

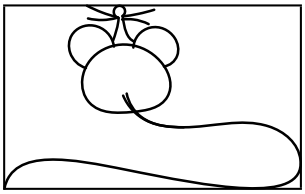


Figure 101. Figure III in the row with common heights

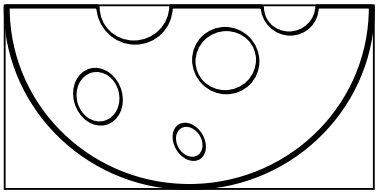


Figure 102. Figure IV in the row with common heights

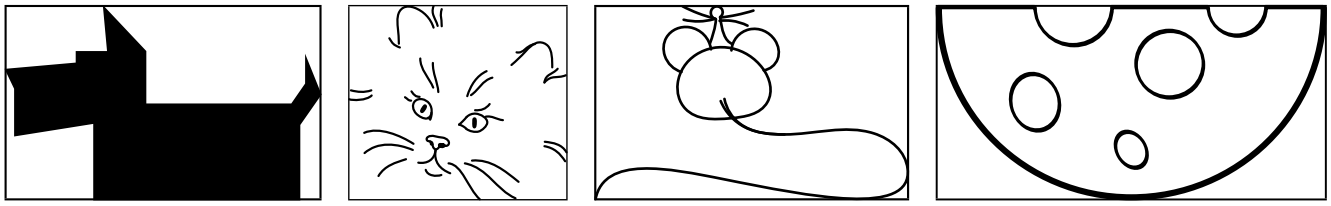
The next example is a variation of previous one. The command \CommonHeightRow here was used for the subfloatrow environment.

```
⟨preamble⟩
  \usepackage{graphicx}
```

```
  \floatsetup[figure]{style=plain}\floatsetup[widefloat]{margins=hangleft}
⟨preamble⟩

\begin{figure*}\fboxsep-.4pt
\ffigbox{}{\CommonHeightRow{\begin{subfloatrow}[4]
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\ffigbox[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{...}}
\end{subfloatrow}}\caption{Figure with a row of parts with common height}}
\end{figure*}%
```

Here you may see the result.



a) Part I in the row with common heights

b) Part II in the row with common heights

c) Part III in the row with common heights

d) Part IV in the row with common heights

Figure 103. Figure with a row of parts with common height

The last example load labels of parts of figures beside graphics.

```
⟨preamble⟩
  \usepackage{graphicx}
  \floatsetup[figure]{style=plain}\floatsetup[widefloat]{margins=hangleft}
  \floatsetup[subfigure]{capbesideposition=left}
⟨preamble⟩

\begin{figure*}\fboxsep-.4pt
\ffigbox{}{\CommonHeightRow{\begin{subfloatrow}[4]\useFCwidth
\fcapside[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{}}
\fcapside[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{}}
\fcapside[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{}}
\fcapside[\FBwidth]
{\includegraphics[height=\CommonHeight]{...}}{\caption{}}
\end{subfloatrow}}\caption{Figure with a row of parts with common height}}
\end{figure*}%
```

Here you may see the result.

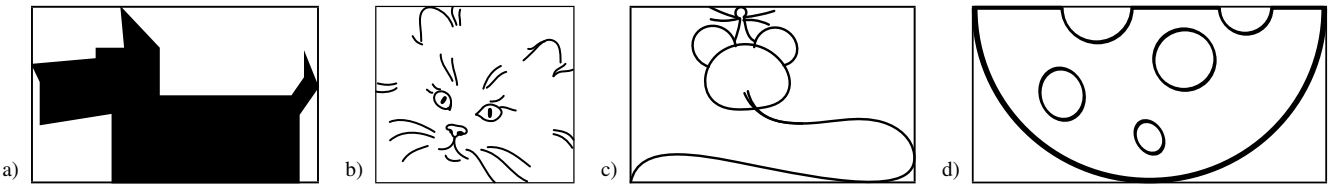


Figure 104. Figure with a row of parts with common height (labels beside)

The examples with beside figures which also include labeled parts.
The row with labels beside.

```
\floatsetup[subfigure]{capbesideposition=left}
\begin{figure*}[H]
\CommonHeightRow*%
{\begin{floatrow}
\ffigbox[\FBwidth]{}%
{\begin{subfloatrow}\useFCwidth
\fcapside[\FBwidth]{}{\caption{}\label{...}...}
\fcapside[\FBwidth]{}{\caption{}\label{...}...}
\end{subfloatrow}\caption{Common caption~I}}
\ffigbox[\FBwidth]{}%
{\begin{subfloatrow}\useFCwidth
\fcapside[\FBwidth]{}{\caption{}\label{...}...}
\fcapside[\FBwidth]{}{\caption{}\label{...}...}
\end{subfloatrow}\caption{Common caption~II...}}
\end{floatrow}}%
\end{figure*}
```

.

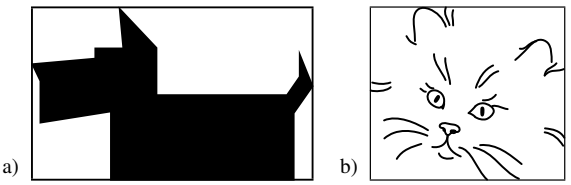


Figure 105. Common caption I in a multilevel row with common height of graphics

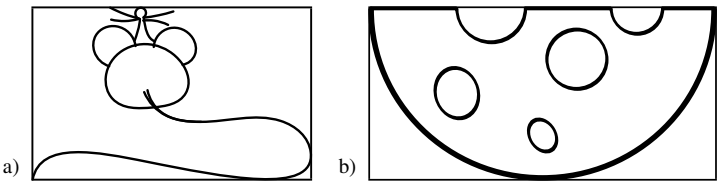


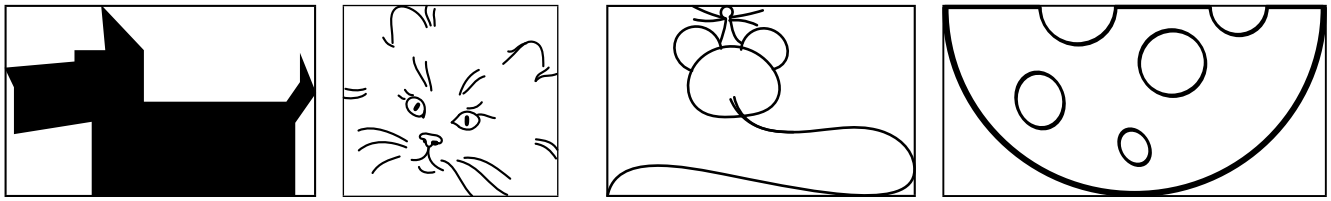
Figure 106. Common caption II in a multilevel row with common height of graphics

The row with labels below.

```
\floatsetup[subfigure]{capbesideposition=left}
\begin{figure*}[H]
\CommonHeightRow*%
{\begin{floatrow}
\ffigbox[\FBwidth]{}%
{\begin{subfloatrow}
\ffigbox[\FBwidth]{}{\caption{}\label{...}...}
\ffigbox[\FBwidth]{}{\caption{}\label{...}...}
\end{subfloatrow}\caption{Common caption~I}}
```

```
\ffigbox[\FBwidth]{}%
{\begin{subfloatrow}
\ffigbox[\FBwidth]{}{\caption{}\label{...}...}
\ffigbox[\FBwidth]{}{\caption{}\label{...}...}
\end{subfloatrow}\caption{Common caption~II...}}
\end{floatrow}}%
\end{figure*}
```

Here you may see the result.



a) Part I in the row with common heights

b) Part II in the row with common heights

a) Part III in the row with common heights

b) Part IV in the row with common heights

Figure 107. Common caption I in a row with common height of graphics

Figure 108. Common caption II in a row with common height of graphics

The row with labels beside.

```
\floatsetup[subfigure]{capbesideposition=left}
\begin{figure*}[H]
\CommonHeightRow*%
{\begin{floatrow}
\ffigbox[\FBwidth]{}
{\begin{subfloatrow}[3]\useFCwidth
\fcapside[\FBwidth]{}{\caption{}\label{...}...}
\fcapside[\FBwidth]{}{\caption{}\label{...}...}
\fcapside[\FBwidth]{}{\caption{}\label{...}...}
\end{subfloatrow}\caption{Common caption~I}}
\ffigbox[\FBwidth]{}{\caption{Caption~II...}\label{...}...}
\end{floatrow}}%
\end{figure*}
```

Here you may see the result.

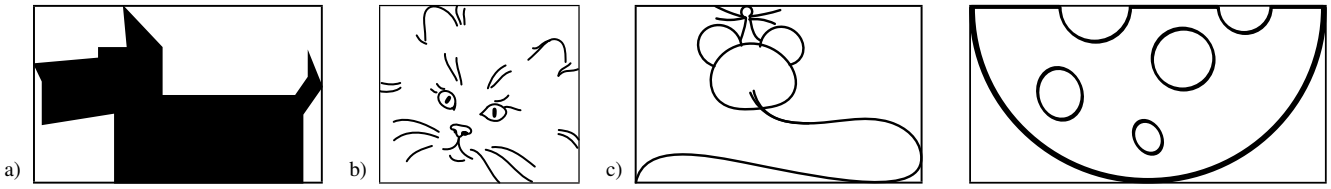


Figure 109. Common caption I

Figure 110

11.2 Sample Files

The floatrow package distribution offers a few files with examples, which show settings, not covered by current document (some of them are bit exotic for technical literature). The samples have no aim to create perfect layout, but to show easy modification for all float types, and show goals and drawbacks in combinations of chosen layout with different float types and their contents.

Note. All miscellaneous float styles (i.e. almost all sample files) need at least two LaTeX runs.

The list of samples:

frsample01.tex all possible combinations of predefined floatrow styles for captions above/below floats with foot material; the plain floating environments and floatrows were created, also the boxes with alone objects and alone captions;

frsample02.tex all possible combinations of predefined floatrow styles for beside captions and all possible caption positions;

frsample03.tex various tests with tables;

frsample04.tex sample with fancy layout with usage of beside captions;

frsample05.tex one-column facing layout; miscellaneous caption settings.

frsample06.tex examples of attempts to get common height for rectangular graphics (photos) in the filled row of floats or parts of floats. Also the examples of usage of the \Xhsize command in the mixed-level rows were added.

The next bundle of samples is a few file-headers with various preambles which run the same file with various float layouts. For these examples a new float type of float textbox was created. It includes text in its object contents.

frsample10.tex one column non-facing layout; figures printed in plain style; text boxes use miscellaneous ruled style;

frsample11.tex one-column non-facing layout with elements hanged on left margin (e.g. wide floats, in starred environments, like figure*);

frsample12.tex two-column layout with attempts of colored float styles.

Also added sample file sample-longtable.tex was added which uses *beta-temp* package-patch fr-longtable with defined commands \endlasthead and \endprelastfoot which defines captions for continued and last pages of long table in three possible ways.

11.3 Obsolete Commands

11.3.1 The User Interface—New Floats [float]

\newfloat *The most important command in float is the* \newfloat *command*[1]*. It is patterned on* \newtheorem*. The* \newfloat *command takes three required and one optional argument; it is of the form*

\newfloat{⟨*type*⟩}{⟨*placement*⟩}{⟨*ext*⟩}[⟨*within*⟩]

- {⟨*type*⟩} *is the 'type' of the new class of floats, like* program *or* algorithm*. After the appropriate* \newfloat*, commands like* \begin{program} *or* \end{algorithm*} *will be available.*
- {⟨*placement*⟩} *gives the default placement parameters for this class of floats. The placement parameters are the same as in standard LaTeX, i.e.,* t*,* b*,* p *and* h *for 'top', 'bottom', 'page' and 'here', respectively.*
- {⟨*ext*⟩} *When LaTeX writes the captions to an auxiliary file for the list of figures (or whatever), it'll use the job name followed by* {⟨*ext*⟩} *as a file name.*
- [⟨*within*⟩] *Finally, the optional argument* ⟨*within*⟩ *determines whether floats of this class will be numbered within some sectional unit of the document. For example, if* [⟨*within*⟩] = chapter*, the floats will be numbered within chapters. (In standard LaTeX, this happens with figures and tables in the* report *and* book *document styles.) As an example, Program 11.1 was created by a command sequence similar to that shown in the following Example*[2]*.*

Floatrow note. Also a \newfloat* pair was created which works similar to \restylefloat* command (see below).

```
\floatstyle{ruled}
\newfloat{Program}{tbp}{lop}[section]
... loads o' stuff ...
\begin{Program}
\begin{verbatim}
... program text ...
\end{verbatim}
\caption{... caption ...}
\end{Program}
```

Example 11.1. This is another silly floating Example. Except that this one doesn't actually float because it uses the [H] optional parameter to appear **Here**. (Gotcha.)

\floatstyle *The* \floatstyle *command sets a default float style. This float style will be used for all the floats that are subsequently defined using* \newfloat*, until another* \floatstyle *command appears. The* \floatstyle *command takes one argument, the name of a float style. For instance,* \floatstyle{ruled}*. Specifying a string that does not name a valid float style is an error.*

\floatname *The* \floatname *command lets you define the* float name *that LaTeX uses in the caption of a float, i.e., 'Figure' for a figure and so on. For example,* \floatname{program}{Program}*. The* \newfloat *command sets the float name to its argument* ⟨*type*⟩ *if no other name has been specified before.*

Program 11.1 The first program. This hasn't got anything to do with the package but is included as an example. Note the `ruled` float style.

```
#include <stdio.h>


int main(int argc, char **argv) {
       int i;
       for (i = 0; i < argc; ++i)
               printf("argv[%d] = %s\n", i, argv[i]);
       return 0;
}
```

`\floatplacement` *The* `\floatplacement` *command resets the default placement specifier of a class of floats. E.g.,* `\floatplacement{figure}{tp}`.

`\restylefloat` *The* `\restylefloat` *command is necessary to change styles for the standard float types* `figure` *and* `table`. *Since these aren't usually defined via* `\newfloat`, *they don't have a style associated with them. Thus you have to say, for example,*

```
\floatstyle{ruled}
\restylefloat{table}
```

to have tables come out `ruled`. *The command also lets you change style for floats that you define via* `\newfloat`, *although this is, typographically speaking, not a good idea. See table 18 for an example*[1]. *There is a* `\restylefloat*` *command which will restyle an existing float type but will keep the new float style from taking over the* `\caption` *command. In this case the user is responsible for handling their own captions.*

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$
0	1							
1	1	1						
2	1	2	1					
3	1	3	3	1				
4	1	4	6	4	1			
5	1	5	10	10	5	1		
6	1	6	15	20	15	6	1	
7	1	7	21	35	35	21	7	1

Table 18: Pascal's triangle. This is a re-styled LaTeX `table`.

[1] It doubles the `\DeclareNewFloatType` command.

[2] Settings for Example float environment were created by `\DeclareNewFloatType` macro stuff.

[1] The float package created special caption style with bold label for `boxed` style. Please note that `plain` and `boxed` float styles have not any special settings in caption 3.x package. To emulate `boxed` style from float documentation there were: cleared all special caption settings for tables, and restored default colon separator after label.

11.3.2 The \floatsetup Keys, Renamed or Deleted After Version 0.1b

Removed or changed commands

Command	Changed to
\renewfloatstyle, \newfloatstyle, \definefloatstyle	\DeclareFloatStyle—this command uses \floatsetup mechanism
\restorerestylefloat	removed
\captionskip	command, not a skip
\floatfootskip	command, not a skip

Commands, replaced by keys

Deleted Command	Key Analog
\floatobjectset	in current version *do not use for definition of object settings*, use key objectset=
\alignsidecaption	capbesideframe=yes
\capbesidecenter, \capbesidetop, \capbesidebottom, \capbesideinside, \capbesideoutside, \capbesideleft, \capbesideright	capbesideposition=center capbesideposition=top capbesideposition=bottom capbesideposition=inside capbesideposition=outside capbesideposition=left capbesideposition=right
\floatrowsep, \floatcapbesidesep	in current version *do not use for definition of separation material*, use keys floatrowsep= capbesidesep=
\FBcenter, \FBleft, \FBright, \FBnormal	margins=center, margins=raggedright, margins=raggedleft, margins=center,
\setfloatstyle	style=
\Setframe \setframe	use framestyle= and frameset= keys
\setrules	use precode=, postcode=, midcode= (also rowpercode and rowpostcode) keys

Renamed keys

Key	Changed to
attachedcapstyle=	relatedcapstyle=
floatstyle=	style=
floatfont=	font=
putcaptionbeside=	capposition=beside
besidecapposition=	capbesideposition=
besidecapwidth=	capbesidewidth=
besidecapframe=	capbesideframe=
floatmarginsset=	margins=

Renamed keys

Key	Changed to
besidecapsep=	capbesidesep=
Precode=	rowprecode=
Postcode=	rowpostcode=
framereduce=	framefit=
options of objectset= and margins=	options of objectset= and margins= (for unification with analogous key options in caption package)
flushleft,	raggedright,
flushright,	raggedleft,
center	centering