

The floatrow package*

Olga Lapko
Lapko.O@g23.relcom.ru

2006/07/24

Abstract

This package was created like extension or addition of `float` package. The `floatrow` borrows core code from `float`¹ and `rotfloat`² packages, so you must not load these packages.

The `float` package has good mechanism for creation (and easy modification) of common layout for all floats without adding any repeated code in document; unfortunately it deals only with alone (plain) object—caption combinations.

Current package extends this mechanism and allows:

- to change width of float box;
- to set width for float box, which equals 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 put legend-like text;
- to create or modify special layout for each type of float and for different positioning of float and its components (two-column or rotated float).

The `floatrow` package is cooperated with `caption` package (version 3.0 and later). Also the `floatrow` package (like `caption` one) uses `keyval` package mechanism for layout settings.

I think I did my best to follow this idea and I hope that someone likes this idea: helps to maintain this idea anyway, or finds bugs and absurdities in this package or documentation.

Document terminology:

float could include *object*, *caption*, and *foot material*; *float* is created by `figure` or `table` environments (*plain float*), or in `\floatbox` command;
object `tabular` or `graphics`, as contents of table (`figure`) or figure (`table`) or other type of float;
caption text in `\caption`;
foot material explications, legends and/or footnotes inside *float* box (`\footnote`/`\mpfootnotemark`/`\footnotetext`, and `\floatfoot` macros).

*This file has version number v0.1j, last revised 2006/07/24.

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.

Contents

1	Introduction	5
1.1	Do not write that with floatrow package	6
2	Float Layout Settings	7
2.1	Floatsetup Keys	9
2.1.1	Float Style	9
2.1.2	Font settings	9
2.1.3	Position of Caption and of Other Elements	11
2.1.4	Defining Float Foot Position (Legends and Footnotes)	12
2.1.5	Defining Vertical Alignment	12
2.1.6	Facing Layout	13
2.1.7	Defining Float Margins and Object Settings	13
2.1.8	Defining Float Separators	14
2.1.9	Defining Float Rules/Skips	14
2.1.10	Defining Float Frames	15
2.1.11	Defining Float Skips	15
2.1.12	Defining Float Footnote Rule’s Style	16
2.1.13	Loading Style for Fancy Boxes	16
2.2	Settings for One Float	16
2.3	Clearing of Previous Float Type Settings	16
2.4	The Default Float Type Settings	16
2.5	Defining new options	17
2.5.1	Float Style	17
2.5.2	Defining Float Rules/Skips	18
2.5.3	Float Object Justification	18
2.5.4	Float Box Alignment (and Some Other Settings)	18
2.5.5	Defining Float Separators	19
2.5.6	Float font	20
2.5.7	Defining Float Footnote Rule’s Style	20
3	Creation of New Float Type	20
4	Borrowed code	21
4.1	float Package: Compatibility	21
4.1.1	The User Interface—New Floats [float]	21
4.1.2	How macros from float package work in floatrow	23
4.1.3	Printing of Float List [float]	23
4.1.4	The User Interface—[H] Placement Specifier [float]	24
4.1.5	The [H] Placement Specifier—Managing of Page Breaks	25
4.2	rotfloat Package	25
5	Macros for Building Floats	25
5.1	The Floatbox Macro	25
5.1.1	User Commands for Float Boxes	28
5.2	Building Float Row	28
5.3	Usage of Footnotes Inside Float Environment	30
5.3.1	The Legend-Like Macro	30

5.4	Fine Tuning of Vertical Spaces of Float	31
5.5	Running Raw L ^A T _E X Floats	31
6	Style Tandems	32
6.1	The <code>caption</code> Package	32
6.2	The <code>subfig</code> Package	33
6.3	The <code>wrapfig</code> Package	35
6.4	The <code>floatflt</code> package	36
6.5	The <code>picins</code> Package	36
6.6	The <code>rotating</code> Package and <code>sideways...</code> Environment	37
6.7	The <code>lscap</code> Package and <code>landscape</code> Environment	39
6.8	The <code>listings</code> Package	40
6.9	The <code>longtable</code> Package	40
6.10	The <code>hyperref</code> and <code>hypcap</code> Packages	40
6.11	The Incompatibilities	40
7	Limitations	41
8	Acknowledgements	42
9	Appendix	44
9.1	Miscellaneous	44
9.1.1	Mixed Row	44
9.1.2	Usage of <code>Captionsetup</code> and <code>Thisfloatsetup</code> Inside <code>Floatbox</code> Stuff	44
9.1.3	Predefined Beside Caption Width	46
9.1.4	Predefined Beside Caption Width with The Rest Space for Object	46
9.1.5	Caption Beside with Centered Object	47
9.1.6	Width Definition for Beside Caption—Object Box in Float Row	47
9.1.7	Caption Above/Below Float and Caption Beside Float at The One Float Row	48
9.1.8	Photo-Album-like Layouts	49
9.2	Sample files	51
10	Obsolete Commands	52

List of Figures

1	Beside caption (‘one-column’ width)	5
2	Left beside figure of simple row	6
3	Beside figure at the right side of simple figure row	6
4	Plain figure in <code>MyBoxed</code> style	17
5	The simple example of <code>\floatbox</code>	27
6	Beside caption	28
7	Beside figure I, “column” width	29
8	Beside figure II in <code>floatrow</code> , float box has the width of graphic	29
9	Beside figure III in <code>floatrow</code> , occupies the rest of row	29
10	A figure caption in raw L ^A T _E X	32
11	A figure caption in <code>\floatbox</code> in raw L ^A T _E X	32
12	Beside caption vertically top aligned with fancy <code>Shadowbox</code> layout	34

13	Wrapped plain figure (<code>wrapfig</code> package)	35
14	Wrapped figure in <code>\ffigbox</code> (<code>wrapfig</code> package)	35
15	Wrapped figure inside <code>floatingfigure</code> environment (<code>floatflt</code>)	36
16	Wrapped figure (<code>\parpic</code>); <code>\floatbox</code> with <code>\FBwidth</code> option	36
17	Figure inside <code>sidewaysfigure</code> environment	38
18	Beside figure I, top of object box	43
19	Beside figure II, bottom of object box	43
20	Beside figure III, center of object box	43
21	Beside figure IV	43
22	Figure (<code>Boxed</code> style) beside table (vertical centering)	45
23	Float box (<code>\fcapsideleft</code>) width of graphics	45
24	Float box (<code>\fcapsideright</code>) width of rest float row space	45
25	Beside caption, width equals to caption's text	46
27	Plain figure	47
30	Float box (<code>\fcapside</code>) with beside caption in float row width float with caption below	49
31	Float box (<code>\ffigbox</code>) width of rest float row space	49
32	Float box (<code>\ffigbox</code>) width of rest float row space	49
33	Float box with beside caption (<code>\fcapside</code>) in mirror float row width float with caption below	49
34	Float box in photo-album-like layout: alone in left column	50
35	Float box in photo-album-like layout: upper float in right column	50
36	Lower float in right column	50
37	Float box in photo-album-like layout: upper float in left column	51
38	Float box in photo-album-like layout: lower float in left column	51
39	Float box in photo-album-like layout: alone in right column	51

List of Tables

1	A table with caption above	5
2	Float layout styles	10
3	Beside table I long header	11
4	Beside table II with top aligned caption	11
5	Pascal's triangle. This is a re-styled <code>L^AT_EX</code> table.	23
6	Simple tabular with caption above	27
7	Table with additional foot material	31
8	A table caption in raw <code>L^AT_EX</code>	32
9	A table caption in <code>\floatbox</code> in raw <code>L^AT_EX</code>	32
10	Two <code>\subtable</code> 's (created with <code>subfig</code> package)	34
11	Table beside <code>Boxed</code> -styled figure (vertical centering)	45

1 Introduction

One of the first macros of package, created for float contents, is a macro which builds contents of table environment with caption above. The width of caption equals to the width of contents, e.g. of tabular (see table 1).

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

Table 1
A table with caption above

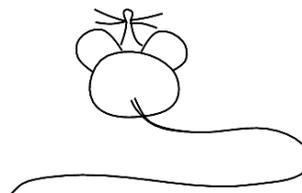
First data	Second data	Third data
A	B	C
D	E	F

The pair of this command, which creates figures—`\ffigbox`—puts caption below contents and default width of caption equals to the width of main text. The default layout of `\ffigbox` is similar to plain `figure` environment; this macro will be useful for creation of beside floats (see example with figures 2 and 3)

The third macro—`\fcapside` (figure 1)—puts caption beside:

```
\begin{figure}
\fcapside
{\caption[Beside caption]{Beside caption ...}\label{...}}
{...}
\end{figure}
```

Figure 1. 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 peace of text to fill space



These three commands abbreviate `\floatbox` command, which creates float boxes (see section 5.1).

The `floatrow` environment allows to put floats side by side, like figures 2 and 3. By default `floatrow` creates necessary number of “columns” (usually two) where floats are placed:

```
\begin{figure}
\begin{floatrow}
\ffigbox
```

```

    {...}\caption{...}\label{...}}

\ffigbox
  {...}\caption{...}\label{...}}
\end{floatrow}
\end{figure}

```



Figure 2. Left beside figure of simple row



Figure 3. Beside figure at the right side of simple figure row

Please note that inside `floatrow` you must use `\ffigbox` or `\ttabbox` macros for each float box. (See also sections 5.1 for more information about commands used in introduction and `\floatbox` command; and 5.2 about `floatrow` environment.)

1.1 Do not write that with `floatrow` package

The `floatrow` package creates some features or limitations of float layout building. If you'll write something like

```

\begin{figure}
\caption{A figure caption}
\centering \input{Mouse.picture}
\end{figure}

```

please do not expect that caption appears at the top of figure. If you want, e.g. to put figure captions above its contents, you must write in preamble `\floatsetup[figure]{style=plaintop}` (see also section 2.1 about `\floatsetup` mechanism).

Another example. If you put beside floats by following way:

```

\begin{figure}
\begin{minipage}{0.5\textwidth}
\centering \input{Mouse.picture}
\caption{A figure caption}
\end{minipage}
\begin{minipage}{0.5\textwidth}
\captionof{table}{A table caption}
\centering \begin{tabular}{cc} A & B \\ C & D \end{tabular}
\end{minipage}
\end{figure}

```

you also won't get expected result. Use `floatrow`, like in example with figures 2 and 3 to get correct result¹.

¹See also the page 44 about mixed float rows

2 Float Layout Settings

The idea of `floatrow` package is to avoid a lot of repeated code inside document for creation of desired layout for floats (which also lowers readability of document), and make layout modification easier if you ought to change book layout. In this case you could think only about *markup* of floats and their contents.

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

The common layout and its modification for captions and special settings for each type of float supported by `caption` package, version 3.x.

The layout settings of `floatrow` package are built similarly to `caption` ones and use `\floatsetup`¹ mechanism which is analog to `\captionsetup` one².

`\floatsetup` You may use layout settings stuff as `floatrow` option in `\usepackage` line.

```
\usepackage[<options>]{floatrow}
```

You may write

```
\usepackage[style=boxed,font=small]{floatrow}
```

The same result you get with

```
\floatsetup{style=boxed,font=small}
```

The lines above declare `boxed` style and `\small` font for float contents. These settings were loaded for *all* float types.

The `\floatsetup` command has following form:

```
\floatsetup[<float type>]{<options>}
```

For defining of a special style for float of one type use optional argument with name of necessary float type. The following command

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

sets a special float style for tables (captions are placed above float objects; in the case of beside floats inside `floatrow` environment captions aligned by top line)³.

There are also special options of `\floatsetup` for settings for floats in different positioning: two-column floats (starred environment, like `figure*`, you may also use, for example, for wide floats) rotated floats etc. The “strength” of options

¹Some 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 10).

²Look also at `caption` documentation (version 3.0 and later)

³The keys, used in special settings for float of one type, are always “stronger” than synonym keys in common setting. In other words, if you want after example above to redefine float style for all floats with `\floatsetup{style=<option>}` you won’t change tables—use `\floatsetup[table]{style=<option>}` again.

in the lists below decreases from previous item to next one (examples based on `figure` environment):

- Wide or two-column floats (e.g. `figure*`, you may load special settings for starred environments even in one-column document):
 - `\floatsetup[widefigure]`—the “strongest” settings; if they absent, package uses settings from next item;
 - `\floatsetup[widefloat]`—these settings “stronger” than settings from next item (`\floatsetup[figure]`); if they absent, package uses settings from next item;
 - `\floatsetup[figure]`; if they absent, package uses settings from optional argument in `\usepackage` line or `\floatsetup{...}` command; if they absent—default package settings (see page 16);
- Wrapped floats (`wrapfigure`):
 - `\floatsetup[wrapfigure]`;
 - `\floatsetup[wrapfloat]`;
 - `\floatsetup[figure]`;
- Rotated floats (in environments, like `sidewaysfigure`):
 - `\floatsetup[rotfigure]`;
 - `\floatsetup[rotfloat]`;
 - `\floatsetup[figure]`;
- Wide rotated floats (`sidewaysfigure*`)¹:
 - `\floatsetup[widerotfigure]`;
 - `\floatsetup[widerotfloat]`;
 - `\floatsetup[rotfigure]`;
 - `\floatsetup[rotfloat]`;
 - `\floatsetup[figure]`;
- Floats with beside captions:
 - `\floatsetup[capbesidefigure]`;
 - `\floatsetup[capbesidefloat]`;
 - setting of outer environments from previous items: `sidewaysfigure*`, `sidewaysfigure`, and `figure*`.

Please note, that usage of `style=` key (see beginning of next section) cancels settings for beside position of caption.

You can also create and change special settings for captions, using the same (*float type*) options in `\captionsetup` stuff.

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

¹The settings for wide float (`widefloat`, `widefigure`) removed—use settings for `widerotfloat` and—here—`widerotfigure`

2.1 Floatsetup Keys

2.1.1 Float Style

`style=` The *float style* is specified by following way:

```
style=(float style name) ,
```

the *(float style name)* you may take from table 2.

The caption's `ruled` style is the only one from `float` package, which was pre-defined in `caption` package. To use it, write

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

2.1.2 Font settings

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

Available font setting options:

<code>scriptsize</code>	Very small size
<code>footnotesize</code>	The size usually used for footnotes
<code>small</code>	Small size
<code>normalsize</code>	Normal size
<code>large</code>	Large size
<code>Large</code>	Even larger size
<code>up</code>	Upright shape
<code>it</code>	<i>Italic shape</i>
<code>sl</code>	<i>Slanted shape</i>
<code>sc</code>	SMALL CAPS SHAPE
<code>md</code>	Medium series
<code>bf</code>	Bold series
<code>rm</code>	Roman family
<code>sf</code>	Sans Serif family
<code>tt</code>	Typewriter family

You may set font for float object like

```
font=small
```

or

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

`footfont=` Defines font for legends or explications. This macro uses `\captionsetup` mechanism (because `\floatfoot` macro uses `caption` mechanism). By default the font size of float foot text equals to footnote text: `footfont=footnotesize`.

Table 2

Float layout styles

Style	\floatsetup keys	Description
		Offered by floatrow package
		Standard L ^A T _E X's layout
plain ^a	<code>(none)</code>	
plaintop ^a	<code>caption=top</code>	The same as plain but puts caption at the top—analogue of float package style. Capitalized form aligns captions above object by top line in floatrow environment
plaintop ^b	<code>caption=TOP</code>	
ruled ^a	<code>caption=top,precode=thickrule, midcode=rule,postcode=lowrule, heightadjust=all</code>	The first one emulates float package's co-named style. Capitalized form aligns captions above object by top line in floatrow environment
Ruled	<code>style=ruled,caption=TOP</code>	
boxed ^{a,b}	<code>caption=2pt,framestyle=fbox, heightadjust=object, framearound=object</code>	The first one emulates float package's style: object's width equals to \hspace or predefined width of float box; frame climbs out to the right and left sides. In capitalized form frame fits the \hspace or predefined width; width of object reduced to fit inside frame. Uppercase form draws frame which fits to current \hspace or predefined width, but around full float: caption, object and foot material.
Boxed	<code>style=boxed,framefit=yes</code>	
BOXED	<code>framestyle=fbox,framefit=yes, heightadjust=all,framearound=all</code>	
		Offered by fr-fancy package. They also need fancybox package.
shadowbox ^b	<code>style=boxed,framestyle=shadowbox</code>	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \shadowbox from fancybox package.
Shadowbox	<code>style=Boxed,framestyle=shadowbox</code>	
SHADOWBOX	<code>style=BOXED,framestyle=shadowbox</code>	
doublebox ^b	<code>style=boxed,framestyle=doublebox</code>	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \doublebox from fancybox package.
Doublebox	<code>style=Boxed,framestyle=doublebox</code>	
DOUBLEBOX	<code>style=BOXED,framestyle=doublebox</code>	
		Additional float style. It also needs fancybox package.
wshadowbox ^b	<code>style=boxed,framestyle=wshadowbox</code>	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.
Wshadowbox	<code>style=Boxed,framestyle=wshadowbox</code>	
WSHADOWBOX	<code>style=BOXED,framestyle=wshadowbox</code>	

When there is set float style with frame around object of fitted to the text width (like Boxed, etc.), and \Floatbox macro uses \FBwidth command as (*width*) option, which sets box width equal to float object's width (see page 25), the width of all float other elements in this case enlarged to get width of framed object.

^aThe styles analogous to float package style.

^bDuring usage of these styles in floatrow environment you ought to enlarge space between floats, using key floatrowsep).

2.1.3 Position of Caption and of Other Elements

`capposition=` Defines position of captions. It is similar to `position=` key in `caption` package, but it has two additional options¹: `TOP`, if you prefer to align captions above objects, in the case of beside floats (in `floatrow` environment), by the top line; and `beside` to put caption beside object (this option could be more preferable in settings for one environment, see about `\thisfloatsetup` on the page 16):

`top` caption above object (equals to `\captop` command);
`TOP` caption above object and also aligned by top line in float row (equals to `\CAPTOP` command). For example the `Plaintop` style is the variant of `plaintop` where used `capposition=TOP` settings, see tables 3–4;
`bottom` caption below object (equals to `\capbot` command);
`beside` caption beside object (equals to `\capbeside` command).

Table 3
Beside table I with long long 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 4
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

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

`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;
`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;
`outside` least popular option: caption printed in outer side of page if `twoside` option switched on in document class.

¹The option `auto` isn't used by `capposition=`.

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 float contents.

You may define position of beside caption by following:

`capbesideposition={top,outside}` .

capbesidewidth= Defines width of beside caption (this option could be more preferable in settings for one environment, see about `\thisfloatsetup` on the page 16). You may set:

`capbesidewidth=4cm` .

If you'll write `capbesidewidth=none` or `capbesidewidth=sidefill` (this is default key setting), the width of caption will be counted by usual way, accordingly to float width (i.e. occupies the rest width of float box, see figure 6 on the page 28).

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

2.1.4 Defining Float Foot Position (Legends and Footnotes)

footposition= Defines position of `\footnote`'s and `\floatfoot`'s in float box with above/below captions.

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.

See examples in file `frsample01.tex`. In the case of caption beside float object, footnotes and foot text always placed below caption.

2.1.5 Defining Vertical Alignment

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

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

You may define height adjustment even as followed:

`heightadjust={caption,noobject}` .

valign= Defines vertical alignment of float objects in `floatrow` if `heightadjust=all` or `heightadjust=object`, or `\floatbox` stuff uses $\langle height \rangle$ argument with value, which differs from the height of object. The options of this key 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;
`b` aligns objects by bottom line;
`s` stretches objects by full height (if it is possible).

2.1.6 Facing Layout

facing= This key defines whether facing layout used for floats. This key works if `twoside` option switched on in document class.

2.1.7 Defining Float Margins and Object Settings

margins= Defines margins 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 flushed to the left;
`raggedleft` float flushed to the right.

You may create your own alignment settings, see page 18.

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

`justified` Typesets the object text as a normal paragraph. (This is the default.)
`centering` Each line of the object text will be centered.
`raggedright` Each line of the object text will be moved to the left margin.
`RaggedRight` Each line of the object text will be moved 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 object text.
`raggedleft` Each line of the object text will be moved to the right margin.
`RaggedLeft` Analogous to `RaggedRight`.

You may also create your own settings (see page 18)

floatwidth= You may use this key for redefinition of width of floats below. You may use

`floatwidth=.6\hsize`

or

`floatwidth=7cm`

This key could be useful in settings for one float environment (see about `\thisfloatsetup` on the page 16).

If you use option `floatwidth=sidefill` for objects with beside captions (in the case of key `capbesidewidth=` uses absolute value, like `4cm`) the object (instead of caption) occupies the rest space of float box (see appendix, figure 26 on the page 47)

2.1.8 Defining Float Separators

`floatrowsep=` Sets separation material between beside float boxes (defined by `\floatbox` macro etc., see page 25) inside `floatrow` environment (see page 28).

`capbesidesep=` Sets separation material between object and beside caption.

Both key settings work similarly to `labelsep=` key from `\captionsetup`.

They use following predefined options:

<code>columnsep</code>	horizontal skip = <code>\columnsep</code> ;
<code>quad</code>	horizontal skip = 1 em;
<code>qqquad</code>	horizontal skip = 2 em;
<code>hfil</code>	horizontal skip = 1 fil (like <code>\hfil</code>);
<code>hfill</code>	horizontal skip = 1 fill (like <code>\hfill</code>);
<code>none</code>	empty separator.

2.1.9 Defining Float Rules/Skips

`precode=` Defines skip, rule or other analogous code above float box.

`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 43).

`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.

`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 43).

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

<code>none</code>	absent code;
<code>thickrule</code>	thick rule (.8pt) with 2pt vertical skip below—rule above float box in <code>ruled</code> style;
<code>rule</code>	rule of default thickness (.4pt), with 2pt vertical skips above and below—middle rule in <code>ruled</code> style;
<code>lowrule</code>	rule of default thickness (.4pt), with 2pt vertical skip above—rule below float box in <code>ruled</code> style;
<code>captionskip</code>	vertical skip equal to value, defined in <code>captionskip=</code> key.

You may create your own options, see page 18.

2.1.10 Defining Float Frames

`framestyle=` Defines type of frame; the `floatrow` package offers only one type of frame:

`fbox` Standard frame.

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

`colorbox` colored frame, needs also `color` package; if not defined, there is used `\fbox`;

`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;

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

`object` float object contents;

`all` full float box including object, caption, and any foot text.

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

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

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

`colorframeset=` Offered by `fr-fancy` package¹: defines concrete color box (default is standard `\fbox`). In current version you may use only `\fcolorbox` as color box. You may define your color box like

```
colorframeset={\fcolorbox{white}{yellow}} .
```

2.1.11 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`, `plaintop` or `Plaintop`:

```
captionskip=10pt .
```

`footskip=` Defines vertical space before foot material and footnotes.

```
footskip=4pt ,
```

or

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

¹Use `fancyboxes=yes` option in `\usepackage` line.

2.1.12 Defining Float Footnote Rule's Style

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

<code>normal</code>	standard L ^A T _E X definition, used in <code>minipage</code> environments, the width of it equals to 0.4 of current width of text (<code>\columnwidth</code>);
<code>limited</code>	like previous one but max width of footnote rule equals to the value defined by <code>\frulemax</code> command, like: <pre>\newcommand\frulemax{1in}</pre>
<code>fullsize</code>	rule to full current text width.
<code>none</code>	Absent rule.

2.1.13 Loading Style for Fancy Boxes

`fancyboxes=` This boolean key loads `fr-fancy` package. This key you may use only in optional argument in `\usepackage` line.

2.2 Settings for One Float

`\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 argument¹.

2.3 Clearing of Previous Float Type Settings

`\clearfloatsetup` If you want to get rid of parameters marked for an automatic use within a particular environment you can use the command²:

```
\clearfloatsetup{<Typ>} .
```

2.4 The Default Float Type Settings

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

```
font=normalsize
footfont=footnotesize
capposition=bottom
capbesideposition=inside
capbesideframe=no
footposition=default
heightadjust=none
facing=no
margins=centering
```

¹The `\thisfloatsetup` defined as abbreviation of `\floatsetup[tmpset]` command.

²See also documentation of `caption` package about `\clearcaptionsetup` command

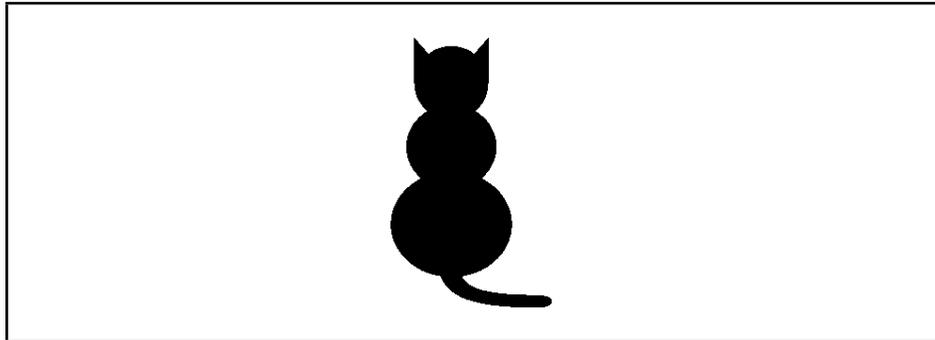


Figure 4. Plain figure in MyBoxed style

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

```
objectset=centering
floatrowsep=columnsep
capbesidesep=columnsep
precode=none
rawprecode=none
postcode=none
rawpostcode=none
midcode=captionskip
captionskip=10pt
```

2.5 Defining new options

In the next few sections goes a small list of commands, which define key options for `\floatsetup`. The examples of option definitions were taken mainly from `floatrow` definitions.

2.5.1 Float Style

`\DeclareFloatStyle` Defines new float style. Example shows definition of new float style `MyBoxed`. The figures 4, and some others in current documentation show result.

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

The same result you get with:

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

2.5.2 Defining Float Rules/Skips

`\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 you may use in `rowprecode`, `precode`, `midcode`, `postcode`, and `rowpostcode` keys.

```
\DeclareFloatVCode{thickrule}{\hrule height.8ptdepth0pt%
\kern2pt}
```

2.5.3 Float Object Justification

`\DeclareObjectSet` You may define justification for `objectset=` key like:

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

In option's definition you may try to include any regular commands which you need to put before each object contents in float environment¹.

2.5.4 Float Box Alignment (and Some Other Settings)

`\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}{}}
```

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

```
\setfloatmargins{\left margin}{\right margin}
```

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

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

You may set even much more complex definition:

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

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.

¹You may also use key options declared by `\DeclareCaptionJustification` command of caption package as options for `objectset=` key.

The `\setfloatmargins` consists of 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`. The settings which use `\floatfacing` command work only in the case when key `facing=yes` used.

Notes. 1) The `floatrow` expands some settings of table layout to the `longtable` environment, so you may set `\LTleft` and `\LTright` parameters inside `\setfloatmargins` command. For example, `centering` option was defined like:

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

2) The font settings loaded in `\floatsetup` in `longtable` environment 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), you must restore correct font size, family, and shape for caption contents and write:

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

or

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

2.5.5 Defining Float Separators

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

```
\DeclareFloatSeparators{columnsep}{\hspace\columnsep}
```

or more complex, using color package, and `\floatfacing` macro (do not forget the `facing=yes` key):

```
\DeclareFloatSeparators{colorsep}%
{\begingroup\color{blue}%
\floatfacing{\hspace14pt\vrule width1.8pt\hspace2pt}%
{\hspace2pt\vrule width1.8pt\hspace14pt}%
\endgroup}
```

You may use defined option by both `floatrowsep=` and `capbesidesep=` keys¹.

¹You may use key options declared by `\DeclareCaptionLabelSeparator` command.

2.5.6 Float 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.)

2.5.7 Defining Float Footnote Rule's Style

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

```
\DeclareFloatFootnoterule{fullsize}{%
  \kern-3\p@
  \hrule \@width\hsize\kern 2.6\p@}
```

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

3 Creation of New Float Type

`\DeclareNewFloatType` For creation of new float type there was created `\DeclareNewFloatType` command which also uses $\langle key \rangle = \langle value \rangle$ mechanism:

```
\DeclareNewFloatType{<type>}{<options>}
```

The $\langle type \rangle$ argument includes the new float environment name.

The $\langle options \rangle$ could include following keys:

- `placement=` The option of this key could have any combination of letters `t`, `b`, `h`, and `p`, which define placement of current float type on the page in the case float environment has no any 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 the file extension from which gathered list of floats. (As default, captions are gathered in `lox`-file).
- `within=` Declares the section head of document, by which current float resets its numbering to zero. If this key absent, float increases during whole documentation.
- `relatedcapstyle=` In `float` package the non-starred `\newfloat`/`\restylefloat` macros attach related caption style for float styles (see section 4.1). 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`.

The `\DeclareNewFloatType` was used for defining of `Example` environment (see page 22). It consists of following code:

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

4 Borrowed code

4.1 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 package¹ the floatrow package loads code which emulates already loaded float package v1.3, so next requests for float are 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 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 supported². See also section 4.1.2 about translation of these commands into `\floatsetup` command.

Sections, signed with “[float]” were borrowed from float's documentation.

4.1.1 The User Interface—New Floats [float]

`\newfloat` The most important command in float is the `\newfloat` command³. It is patterned on `\newtheorem`. The `\newfloat` command takes three required and one optional argument; it is of the form

$$\text{\newfloat}\langle type \rangle \langle placement \rangle \langle ext \rangle [\langle within \rangle]$$

- $\langle type \rangle$ 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.
- $\langle placement \rangle$ gives the default placement parameters for this class of floats. The placement parameters are the same as in standard L^AT_EX, i.e., `t`, `b`, `p` and `h` for ‘top’, ‘bottom’, ‘page’ and ‘here’, respectively.
- $\langle ext \rangle$ When L^AT_EX writes the captions to an auxiliary file for the list of figures (or whatever), it'll use the job name followed by $\langle ext \rangle$ as a file name.
- $[\langle within \rangle]$ Finally, the optional argument $\langle within \rangle$ determines whether floats of this class will be numbered within some sectional unit of the document. For example, if $[\langle within \rangle] = \text{chapter}$, the floats will be numbered within

¹I'm aware about `algorithm` package.

²The better way is to use `\floatsetup` macros. The floatrow package supports obsolete macros but there is no guarantee.

³It doubles the `\DeclareNewFloatType` command.

Program 4.1 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;
}
```

chapters. (In standard L^AT_EX, this happens with figures and tables in the report and book document styles.) As an example, Program 4.1 was created by a command sequence similar to that shown in the following Example¹.

Floatrow note. There is also created a `\newfloat*` pair, 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 4.1. 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.)

- | | |
|------------------------------|--|
| <code>\floatstyle</code> | The <code>\floatstyle</code> command sets a default float style. This float style will be used for all the floats that are subsequently defined using <code>\newfloat</code> , until another <code>\floatstyle</code> command appears. The <code>\floatstyle</code> command takes one argument, the name of a float style. For instance, <code>\floatstyle{ruled}</code> . Specifying a string that does not name a valid float style is an error. |
| <code>\floatname</code> | The <code>\floatname</code> command lets you define the <i>float name</i> that L ^A T _E X uses in the caption of a float, i.e., 'Figure' for a figure and so on. For example, <code>\floatname{program}{Program}</code> . The <code>\newfloat</code> command sets the float name to its argument (<i>type</i>) if no other name has been specified before. |
| <code>\floatplacement</code> | The <code>\floatplacement</code> command resets the default placement specifier of a class of floats. E.g., <code>\floatplacement{figure}{tp}</code> . |
| <code>\restylefloat</code> | The <code>\restylefloat</code> command is necessary to change styles for the standard |

¹Settings for Example float environment were created by `\DeclareNewFloatType` macro stuff.

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 5: Pascal’s triangle. This is a re-styled L^AT_EX table.

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 5 for an example¹. 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.

4.1.2 How macros from float package work in floatrow

The combination of `\floatstyle{<style>}` and any of `\newfloat{<float>}` or `\restylefloat{<float>}` commands in floatrow package set float layout in the way:

```
\floatsetup[<float>]{style=<style>}
```

Please note that co-named keys from `\floatsetup[<float>]{...}` will overwrite the `\floatsetup{...}` settings. That means that if you set float layout with these obsolete commands, the next layout tuning you ought to do with `\floatsetup[<float>]{...}` stuff only.

4.1.3 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

¹The 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.

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`.

4.1.4 The User Interface—[H] Placement Specifier [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}[H]
\begin{tabular}{c1}
\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 \LaTeX ed the documentation¹ just now it turned out that there was a page break that fell exactly between the “So now” line 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

¹For float package.

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

During usage of [H] option I found that it is necessary to set common penalties before and after “anchored” floats.

The rules for page breaking around such floats similar to page breaking of lists in the book.

To follow the idea of `\allowdisplaybreaks` command from `amsmath` package there is created a beta-temp¹ version of `listpen` package, which offers following commands:

```
\allowprelistbreaks sets penalty before lists (and also “anchored” floats);  
\allowpostlistbreaks sets penalty after lists;  
\allowitembreaks sets penalty between list items.
```

All they set globally, inside group or inside environments the penalties 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 L^AT_EX classes: `book`, `article` etc. ([-1] option equal to `-\@lowpenalty` value).

4.2 rotfloat Package

Code of `rotfloat` package was also borrowed by `floatrow` package. In the case of loaded `rotfloat` package before `floatrow` you will get error message.

The `floatrow` package loads code which pretends that `rotfloat` 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 ‘option clash’ error message.

5 Macros for Building Floats

5.1 The Floatbox Macro

This package offers macro, which creates necessary layout of float object contents and caption (see also the introduction). The main macro looks like:

```
\floatbox[⟨preamble⟩]{⟨capttype⟩}[⟨width⟩][⟨height⟩][⟨vert pos⟩]  
  {⟨caption⟩}{⟨object⟩}2
```

¹I suppose that there will be created such support in `paralist` package and think it is better to follow one grammar for similar situations.

²The order of mandatory arguments and their contents makes not difference during building of float box. This macro historically needs two mandatory arguments, but they could be filled freely.

where:

- <preamble>*** there could be `\capbeside` command which places caption beside float contents; `\nocapbeside` (to put caption above/below, accordingly to float type's style); or another systematic command (even with usage of `\captionsetup` and `\thisfloatsetup`—see examples in documentation and appendix).
- <captype>*** the type of float this command created for;
- <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 command `\FBwidth` in the *<width>* argument allows usage of natural width of float contents.
Note. If you use `\FBwidth` in *<width>* argument be sure that object contents can be placed in `\hbox` (you only allowed to use `\vspace` command at the very beginning and very end of object contents for fine tuning of vertical spaces);
- <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);
- <vert pos>*** vertical alignment of object contents in object's box in case of *<height>* argument has larger value than height of object contents, float placed inside `floatrow`.
- <caption>*** text of caption and legends (you may use `\floatfoot` command (see page 30); you may also use `\footnote`/`\mpfootnotemark`/`\footnotetext` stuff for footnotes inside float, or `\floatfoot` command;
- <object>*** Contents of float; you may use `\footnote`/`\mpfootnotemark`/`\footnotetext` stuff or `\floatfoot` command.

First simplest example:

```
\begin{figure}
\floatbox{figure}[4cm]
{\caption{The simple example ...}%
\label{fig:WcatI}}
{\input{TheCat.picture}}
\end{figure}
```

created the picture in the figure 5.

The more preferable way for usage of `\floatbox` is to put table caption *above* table contents and set caption width equal to table width. Caption position was defined in document preamble by command `\floatsetup[table]{style=Plaintop}`

```
\begin{table}
\floatbox{table}[\FBwidth] {\caption{Simple tabular in
\cmd{\floatbox}
with caption above}\label{tab:tabII}}
{\begin{tabular}{|l|c|c|}
```



Figure 5. The simple example of `\floatbox`

Table 6

Simple tabular with caption above

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

```

\hline
Column head & Data I & Data II \\
\hline
First row & 1 & 2 \\
...
\hline
\end{tabular}}
\end{table}

```

as shown in table 6. To fit size of caption (i.e full float box) to size of `tabular` environment there was used `\FBwidth` command in $\langle width \rangle$ argument.

The next example shows `figure` environment with beside caption. In this example only `\capbeside` command used in $\langle preamble \rangle$ argument, and `\FBwidth` in $\langle width \rangle$ argument. The caption occupies the rest space of float box's width, `\hsize` (in current example the width float box equals to main text width—`\textwidth`).

```

\begin{figure}
\floatbox[\capbeside]{figure}[\FBwidth]
{\caption[Beside caption]{Beside caption ...}\label{...}}
{...}
\end{figure}

```

The default width of beside object and caption are equal to one “column” width, i.e. half of current `\hsize` (figure 1) width excluding float margins and separation material between float and caption. (See more examples with float and beside captions in appendix, page 46.)

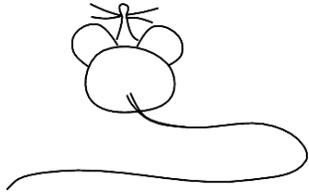


Figure 6. Beside caption and more text and some more text and a bit more text and a little more text and a little peace of text to fill space

5.1.1 User Commands for Float Boxes

The usage of plain `\floatbox` command sometimes is rather complex. You may define commands-abbreviations for your own purposes and define there some additional style settings.

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 float environment this command is created for;

<preamble> you may use commands, mentioned in page 26 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> here you put any dimension like `6cm` or `.75\textwidth`¹; if you put `\FBwidth` command the width of object—caption box (in case of caption above or below object—there could be defined `\nocapbeside` in *<preamble>*) or width of object box (if caption stays beside object—`\capbeside` in *<preamble>*) will be equal to width of object.

There are three already defined commands-abbreviations:

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

First two defined for figures and last one—for table. You may redefine existing macros using `\renewfloatcommand` command (it uses the same arguments as `\newfloatcommand` one).

5.2 Building Float Row

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

¹There is also exists possibility to set common width for current float type of float using `floatwidth` key in `\floatsetup` stuff.



Figure 7. Beside figure I, “column” width



Figure 8. Beside figure II in floatrow, float box has the width of graphic

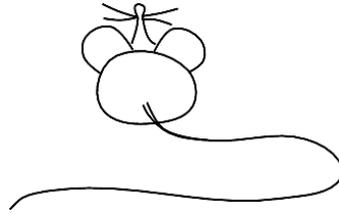


Figure 9. Beside figure III in floatrow, occupies the rest of row

This environment looks like:

```
\begin{floatrow}[\langle number of beside floats \rangle]
\floatbox...
...
\floatbox...
...
\end{floatrow}
```

And here is example. The first float (used `\ffigbox`) has default width equal to “column” width, the next uses width of included graphic (uses command `\FBwidth` in optional argument $\langle width \rangle$), third one occupies the rest horizontal space of page which was counted with command `\Xhsize`.

Note. Usually the command `\Xhsize` is used for the last float box. 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, or set something in $\langle width \rangle$ argument like `\Xhsize/2` and then `\Xhsize` for two last `\floatbox`’es, see sample files `frsample10–frsample12`, and also sample file `fr-sample.tex`.

```
\begin{floatrow}[3]
\ffigbox
{\caption{Beside figure~I...}...}
{...}

\ffigbox[\FBwidth]
{\caption{Beside figure~II...}...}
{...}

\ffigbox[\Xhsize]
{\caption{Beside figure~III...}...}
{...}
\end{floatrow}
\end{figure}
```

The result you see in the row of figures 7–9. The vertical alignment of floats lays on the bottom of upper part (here: objects) of float and the top of lower part (captions).

5.3 Usage of Footnotes Inside Float Environment

Sometimes table or figure contents have material, which authors mark and then write some explanation like footnotes. Current package has mechanism which allows to put footnotes inside float environment, in the same way as is in `minipage` environment.

In the case of few elements have the same footnote, we cannot use standard `\footnotemark—\footnotetext` combination, because of `\footnotemark` creates the sign of main text footnote. For these cases current package offers `\mpfootnotemark` macro instead of `\footnotemark`¹.

The `\floatbox` macro uses special own definition of footnote rule (the `\FBfootnoterule` command) and skip before footnotes and explications or legends (`footskip=`). (See also page 16 for variants of footnote rule.)

5.3.1 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 `\floatfoot` command. The `\floatfoot` is build by usage of `\caption` stuff and uses caption's justification, but star form (`\floatfoot*`) prints its contents as plain unindented paragraph (see table 7).

For defining of explication font use `footfont=` key (page 9) in `\floatsetup`. You may try to define special settings for float foot using `\captionsetup[floatfoot]` setup.

```
\begin{table}
\ttabbox
{\caption{Table with additional foot material}%
\label{tab:floatfoot}}
{\extrarowheight1pt
\tabcolsep2\tabcolsep
\begin{tabular}{|l|c|c|}
\hline
...
\end{tabular}%
\floatfoot*{'Data I' column ...}}
\end{table}
```

Notes. 1) The `float` package defines additional optional argument after main caption text. 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 `\footnotemark/\footnotetext` stuffs instead.

2) If you use both `\floatfoot` command 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

¹The same macro also is defined in `footmisc` package, version 4.10, dated 2003/01/20.

Table 7

Table with additional foot material

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

float object). See page 12 and sample file `frsample01.tex`. In case of caption beside float object, footnotes and foot text always placed below caption.

5.4 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 there you may use two simple commands `\FBaskip` and `\FBbskip`. For example define

```
\renewcommand\FBaskip{-4pt}
```

to move up float box up (or reduce space above) by 4pt. In current document the `\FBaskip` command was necessary for moving up some of wrapped figures.

Use `\vspace` command for vertical space correction around float object¹.

5.5 Running Raw L^AT_EX Floats

The `floatrow` package redefines float 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. If you still need a raw behavior of float environment, you may input a `\RawFloats` command *inside* environment:

```
\begingroup
\captionsetup[table]{position=top}
\begin{figure}\RawFloats
\begin{minipage}{0.45\textwidth}
\centering ...
\caption{...}\label{...}
\end{minipage}
\begin{minipage}{0.45\textwidth}
\captionof{table}{...}\label{...}
\centering ...
\end{minipage}
\end{figure}
```

¹The plain float 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.



Figure 10. A figure caption in row \LaTeX

Table 8
A table caption in row \LaTeX

A	B
C	D

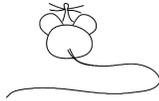


Figure 11. A figure caption in \floatbox in row \LaTeX

Table 9
A table caption in \floatbox in row \LaTeX

A	B
C	D

`\endgroup`

And you'll get figure 10 and table 8¹.

You may see that `floatrow` environment and `\floatbox` commands still work after `\RawFloats` (figure 11 and table 9)

```
\begin{figure}\RawFloats\CenterFloatBoxes
\begin{floatrow}
\floatbox{figure}
{...}{\caption{...}\label{...}}

\floatbox[table][\FBwidth]
{...}{\caption{...}\label{...}}
\end{floatrow}
\end{figure}
```

See also page 44 about how to set and align different float types like figure 11 and table 9 in one row (`floatrow` environment).

6 Style Tandems

The next few sections show examples and explain some noticed features with usage of `floatrow` and another package. There is not the full list of style compatibilities. You may succeed with other versions of mentioned packages, and maybe with not mentioned packages too.

6.1 The caption Package

Tested with versions v3.0a–v3.0i.

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

¹That could work correct if you do not use any fancy float layout.

The `floatrow` adds possibility to create a variations of captions for floats in different positions or layouts (e.g. like wide or two-column floats or rotated floats) using optional argument of `\captionsetup`.

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

```
\captionsetup[widelfloat]{\langle options \rangle}
```

or for wide or two-column figures:

```
\captionsetup[widefigure]{\langle options \rangle}
```

The priority of `\captionsetup` optional arguments is similar to `\floatsetup` ones. In current examples `\captionsetup[widefigure]` will be stronger than `\captionsetup[widelfloat]`. The priority and usage of “*float subtypes*” in optional arguments see on the page 8).

6.2 The `subfig` Package

Tested with version 1.2 dated 2004/01/28 and 1.3. dated 2005/06/28.

The example with `\subfloat`'s (table 10). The setting command in preamble `\floatsetup[table]{style=Plaintop}` includes also settings for subcaption positions used with `subfloat` package (like `\captionsetup[table]{position=top}` in `caption` package):

```
\begin{table}\extrarowheight1pt
\floatbox[table][\FBwidth]
{\caption{Two ...}\label{...}}
{\begin{subfloatrow}
\subfloat[First subtable]
{\begin{tabular}{...}
...
\end{tabular}}

\subfloat[Second subtable with long long long subcaption]
{\begin{tabular}{...}
...
\end{tabular}}}%
\end{subfloatrow}}
\end{table}
```

Note. There was used `subfloatrow` environment in current example. This is simplified `floatrow` environment, which expands vertical alignment of float contents to subfloats and puts horizontal separator, defined by `subfloatrowsep=` key. This key and uses the same options as `floatrowsep=` and `capbesidesep=` keys (options of these keys defined by `\DeclareFloatSeparators` command, page 19).

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

Table 10

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

The next goes example with beside caption (see figure 12).¹

```
\floatsetup{style=Shadowbox,capbesidesep=columnsep,
capbesideframe=yes,capbesideposition={top,inside}}
\fcapside[\FBwidth]
{\begin{subfloatrow}
\subfloat[White cat\label{subfig:w}]{%
\input{TheCat.picture}}

\subfloat[Black cat\label{subfig:b}]{\input{TheBlackCat.picture}}%
\end{subfloatrow}}
{\caption{...}}
```

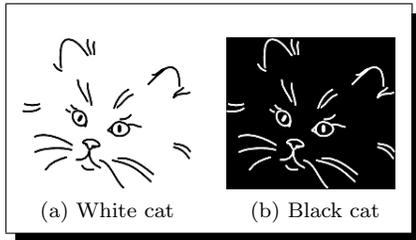


Figure 12. Beside caption vertically top aligned with fancy `Shadowbox` layout

¹In version 1.2 the beside boxed and non-boxed subfloat contents (like in figure 12) could not get aligned bottom (top) subcaptions. In this example I ought to use following lines for subfigure 12b, when `pstricks` package is loaded:

```
\raisebox{\depth}{\input{TheBlackCat.picture}}
```

6.3 The wrapfig Package

Tested with version 3.3 dated 1999/10/12 (style from ltxmisc bundle) and 3.6 dated 2003/01/31 (the separate L^AT_EX package).

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 unfloatoed positioning, uppercase—floatoed 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}
```

Floatrow note. For figure, placed in wrap... environment you must set width in mandatory argument. If you'll write 0mm as {⟨width of figure⟩} argument, wrapfig package will count natural width of float contents.

Sometimes above (below) float box in wrap... environment appears unwanted space. To correct vertical position, use \FBaskip (\FBbskip) commands (see section 5.4).

Special settings.

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

- if exists \floatsetup[wrap⟨captype⟩]{...} floatrow uses these settings—they are the “strongest” settings; if they absent—uses settings of next item;
 - if exists \floatsetup[wrapfloat]{...} floatrow uses these settings—these settings “stronger” than next ones; if they absent—settings of current float
- \floatsetup[⟨captype⟩]{...};
- if they absent—uses default settings (\floatsetup{...} or package settings).

Figure 13 Wrapped plain figure (wrapfig package)

Plain figure fails with package version 3.3

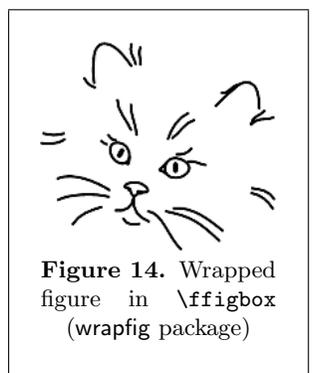
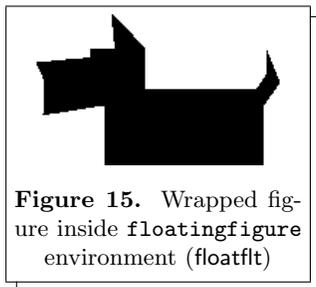


Figure 14. Wrapped figure in \ffigbox (wrapfig package)

Founded limitations.

The usage of plain float environment in version 3.3 will not succeed with current version of 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` (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.

6.4 The floatflt package



Tested with version v.1.3 dated 1996/02/27.

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).

The next limitations could not tied with `floatrow` package.

2) If you put `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) To get correct space before next `\...section` head you may need to put following commands `\makeatletter\@nobreakfalse\makeatother` (as was made in just before next section).

6.5 The picins Package

Figure 16 Wrapped figure (`\parpic`); `\floatbox` with `\FBwidth` option



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

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

The `\parpic` macro usually allows usage of `\floatbox` macro inside of its mandatory argument.

The next limitations could not tied with `floatrow` package.

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

2) You may try to use `\parpic` inside list environment, but in some cases usage of `\parpic` in the list gets bad or wrong layout.

3) 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.

6.6 The rotating Package and sideways... Environment

Tested with version v2.13 dated Sep. 1992.

There is example (figure 17) with rotated float, using `sidewaysfigure`.

```
\floatsetup[figure]{style=WSHADOWBOX}
\begin{sidewaysfigure}\emptyfloatpage
\ffigbox[\FBwidth]
{\includegraphics[width=4in]{pslearn.eps}}
{\caption{Figure inside \env{sidewaysfigure} environment}}%
\label{fig:rot}}
\end{sidewaysfigure}%
```

Special settings.

You may create special settings for all rotated floats, which use `sideways...environment` (see page 8).

For one-column rotated float

- if exists `\floatsetup[rot<captype>]{...}` package uses these settings—the “strongest” settings; if they absent—uses settings from next item;
- if exists `\floatsetup[rotfloat]{...}` package uses these settings—these settings “stronger” than next ones; if they absent—settings for current float (`\floatsetup[<captype>]{...}`); if they absent—uses default settings (`\floatsetup{...}` or package settings).

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

- if exists `\floatsetup[widerot<captype>]{...}` package uses these settings—the “strongest” settings; if they absent—uses settings of next item;
- if exists `\floatsetup[widerotfloat]{...}` package uses these settings—these settings “stronger” than settings in next item; if they absent—uses settings of next item;
- if exists `\floatsetup[rot<captype>]{...}` package uses these settings—these settings “stronger” than settings in next item; if they absent—uses settings of next item;
- if exists `\floatsetup[rotfloat]{...}` package uses these settings—these settings “stronger” than next ones; if they absent—uses settings for current float (`\floatsetup[<captype>]{...}`); if they absent—uses default settings (`\floatsetup{...}` or package settings).

PostScript
Graphic

Figure 17. Figure inside `sidewaysfigure` environment

Notes.

1) If you place two continued rotated floats on facing pages, the better way is to gather them to binder margin. For this reason you may define¹ before first float

```
\buildFBBBOX{\vbox to\rotextwidth\bgroup\vss}{\egroup}
```

and before second one

```
\buildFBBBOX{\vbox to\rotextwidth\bgroup}{\vss\egroup}
```

The `\rotextwidth` dimension—here is saved `\textwidth` of document—the `\columnwidth` and `\textwidth` inside `sideways...` environment are redefined and equal to `\textheight`.

2) In current example you may see command `\emptyfloatpage`. It is offered by `floatpagestyle` package, (installed with `floatrow` package). Macro `\emptyfloatpage` is abbreviation of `\floatpagestyle{empty}`. The last macro redefines page style for the page where *current* float environment appears.

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

6.7 The `lscap` Package and `landscape` Environment

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

The example with usage of `landscape` environment from `lscap` package on the page 43, figures 18–21):

```
\DeclareFloatVCode{lowthickrule}{\kern2pt\hrule height.8ptdepth0pt}
\floatsetup[figure]{style=ruled,rowprecode=thickrule,
  rowpostcode=lowthickrule,capposition=TOP}
\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 whole (the ‘individual’ `\hrule`’s above/below float boxes absent).

The `landscape` environment creates a new page. It would be useful to start new section of document, e.g., appendix. (In current document it was placed just before appendix)

¹In case `rotating` package has `[figuresright]` option, see also files `frsample10.tex–frsample12.tex` and code in `fr-sample.tex`.

²If 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.

6.8 The listings Package

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

This package has its strong layout mechanism for creation for floating algorithms itself. The usage of `\lstset` command and `caption` package settings gives you necessary result¹ 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.

6.9 The longtable Package

There was added a patch to the `\LT@array` macro (`longtable` package)²: this patch adds the same font settings as for `table` environments, and adds code which helps to get width of `longtable` caption equal to the width of table. For settings of caption width there was created a key.

`LTcapwidth=` This key could have any value, like `5cm` or `\hsize`. If you'll write `LTcapwidth=table` or `LTcapwidth=contents`, you will get caption width equal to the width of table. In this case settings for width of caption use information from `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]` macros. The `\floatsetup[longtable]` will be "strongest" in this pair.

6.10 The hyperref and hypcap Packages

There were tested versions v6.74i (`hyperref`) and v1.3 (`hypcap`).

The `floatrow` 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.

6.11 The Incompatibilities

The incompatibilities at first could follow the incompatibilities of `caption 3.x` package:

`ccaption`, `hvfloat`, `nonfloat`

The known incompatibilities of `floatrow` package itself is `sidecap` package³: the `floatrow` package doesn't expands its layouts to `SCfigure` and `SCtable` environments.

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

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

³Despite that I'm trying to follow offered layouts of this package. Great thanks for Rolf Niepraschk and Hubert Gäßlein for package with rich implementation of such float layouts.

7 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`.
- This limitation was mentioned above: the package uses `\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 version 3.0a (and newer) of `caption` package and `floatrow` package do not support optional argument *after* caption “title”. You may use `\floatfoot` macro after main caption argument.
- Do not use `\FBwidth` option for complex float contents (which you could not put inside `\hbox`). (But you allowed to use `\vspace` macro at the very end/very beginning of object contents for fine vertical tuning.)
- 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.
- This is 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 float environment 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 inside `\floatbox` stuff `tabularx` or `tabular*` environments (or any other) with `\hsize` command inside $\langle width \rangle$ argument, you must repeat the `\hsize` argument in $\langle width \rangle$ 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 $\langle width \rangle$ argument of `\floatbox` macro, and in $\langle width \rangle$ 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 $\langle width \rangle$ option of `\floatbox`.

8 Acknowledgements

Thanks for Steven Cochran and Axel Sommerfeldt for all their advices and spirit. Special thanks for Axel for the patient answering, code, and help in *all* my questions and problems in `floatrow` package.

Thanks for *all* involuntary (L^A)T_EX 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 L^AT_EX Companion for this book.

Figure 18 Beside figure I,
top of object box^a

^aThe picture was created with
`\qbezier` macro



Figure 19 Beside figure II, bot-
tom of object box

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

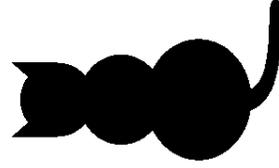


Figure 20 Beside figure III, center
of object box

The Mouse-animal image

^bLook at funny footnote mark!

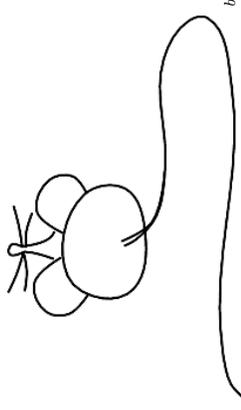


Figure 21 Be-
side figure IV



9 Appendix

9.1 Miscellaneous

9.1.1 Mixed Row

Sometimes author (or designer) wants to put different types of floats, e.g., table and figure beside in one row. This situation creates some problems.

The first problem is that you must put different types of float in one float environment, which sets its own layout for included float box(es). For creation of right layouts write `\killfloatstyle` command at the beginning of `floatrow` environment or just before first “foreign” `\floatbox` macro (in current example—before `\ttabbox`; the `\ffigbox` macro is “native” for `figure` environment).

The second 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 undesirable result. For such situations you may use one of following commands:

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

which align *full* float boxes by center, top or bottom lines. These macros were created by `\buildFBBOX` macro. There is also `\PlainFloatBoxes` which restores standard behavior of `\floatbox`’es.

The usage of `\CenterFloatBoxes` and `\killfloatstyle` you’ll see in mixed float row with figure 22 and table 11:

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

Note. Both figure and table boxes have got width equal to contents of objects. The `\ffigbox` command has optional argument `[\FBwidth]`, but `\ttabbox` have not any option—it uses `[\FBwidth]` option as default (see definitions on page 28).

9.1.2 Usage of `Captionsetup` and `Thisfloatsetup` Inside `Floatbox` Stuff

Example of figures in row (figures 23 and 24). There were used predefined float commands `\fcapsideleft` and `\fcapsideright` with added `\captionsetup` and `\thisfloatsetup`:

```
\newfloatcommand{fcapsideleft}{figure}[\capbeside
\captionsetup[capbesidefigure]{format=break,labelsep=none,
```



Figure 22. Figure (Boxed style) beside table (vertical centering)

Table 11
Table beside Boxed-styled figure (vertical centering)

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

```

justification=raggedleft}%
\thisfloatsetup{capbesideposition=left}}}%
[\FBwidth]
\newfloatcommand{fcapsideright}{figure}[{\capbeside
\captionsetup{capbesidefigure}{format=break,labelsep=none,
justification=raggedright}%
\thisfloatsetup{capbesideposition=right}}}%
[\FBwidth]

```

The caption format break was defined by following way:

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

For these figures created following `\floatsetup` settings:

```

\floatsetup[figure]
{style=Boxed,caposition=beside,
objectset=centering,floatwidth=\columnwidth,
capbesidewidth=5.75cc,capbesidesep=cicero,margins=centering,
capbesideframe=yes,floatwidth=sidefill}

```

The key `floatwidth=sidefill` means that in the case of caption has predefined width, object occupies the rest space of `\hsize` (in the case of no `<width>` option used, or used `\hsize` option).

Left figure uses width of graphics, left one occupies the rest of float row—for this reason just before `\fcapsideright` command was written tricky setting:

Figure 23
Float box
(`\fcapsideleft`)
width of graphics

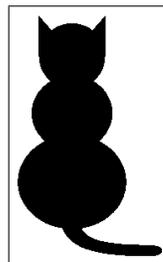


Figure 24
Float box
(`\fcapsideright`)
width of rest float
row space

```
\hsize\Xhsize
```

which, in case of caption placed beside float object defines width of full float box.

The default $\langle width \rangle$ option of `\fcapsideleft` and `\fcapsideright` commands is `\FBwidth`, so for figure 24 on the right (created by `\fcapsideright` command) in the $\langle width \rangle$ option was set `\hsize` command to fill the rest space of float row.

Since in float style the key `heightadjust=object` (included in definition of `Boxed` style), both objects have the same height.

9.1.3 Predefined Beside Caption Width

This example includes `\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 counts natural width of caption contents (see figure 25¹). The object—caption box in this case aligned using alignment settings from `margins` key (its options defined by `\setfloatmargins` or `\floatcapbesidemargins` macro). In this documentation they are centered (see page 18).

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



Figure 25. Beside caption

The settings for current float:

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

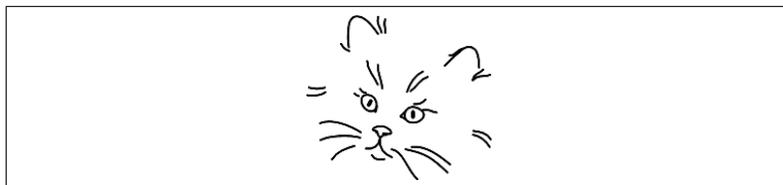
9.1.4 Predefined Beside Caption Width with The Rest Space for Object

The figure 26 uses following float style:

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

¹Inside `\floatbox` you may not set predefined width of caption, but remember that you *must* define width of caption in case of usage of plain float environment.

Figure 26



```
floatwidth=sidefil}
```

The `\Mylen` dimension was defined as width of caption label:

```
\newdimen\Mylen
\settowidth\Mylen{\captionfont\captionlabelfont
\figurename\ \thefigure}
```

9.1.5 Caption Beside with Centered Object

The figure 27 uses tricky float style:

```
\floatsetup[figure]
{style=plain,capposition=beside,objectset=centering,
floatwidth=\columnwidth,capbesidewidth=6cc,
capbesideposition=left,capbesidesep=mcapwidth,
margins=centering,capbesideframe=yes,
floatwidth=sidefil}
```

the `mcapwidth` option is a negative skip equal to caption width:

```
\DeclareFloatSeparators{mcapwidth}{\hskip-\FCwidth}
```

In this case all figures with beside captions centered accordingly to text `\hsize`. Usage of such float layout supposes that all float objects are narrower than `\textwidth` by at least 2 caption widths.

There were also temporarily, in group, changed settings for caption in figure 27:

```
\captionsetup[capbesidefigure]{format=break,labelsep=none,
justification=raggedright}
```

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

The float row with predefined width of beside object—caption boxes (figures 28 and 29): just define before `\fcapside` command something like

Figure 27
Plain figure



```
\setlength\hsize{1.098\hsize}%
```

as was defined for first figure, or

```
\setlength\hsize\Xhsize
```

before second (here is clear that you can set `\setlength\hsize{.902\hsize}`).

Since there were used `\fcapside` commands at the beginning of `floatrow` environment the `\FCwidth` command was defined as `\relax`—in this case the width of caption equals to the width of their contents.



Figure 28

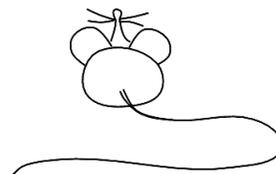


Figure 29

There was used following setup for figures:

```
\floatsetup[figure]  
{style=plain,capposition=beside,objectset=centering,  
 capbesideposition=left,capbesidesep=enskip,  
 margins=centering,capbesideframe=yes,floatwidth=sidefil}
```

9.1.7 Caption Above/Below Float and Caption Beside Float at The One Float Row

The float row with object and beside caption combined with object and caption below (figures 30 and 31). There we ought to use `\TopFloatBoxes`, `\CenterFloatBoxes`, or `\BottomFloatBoxes` commands to get correct layout—since the *height* argument in both float boxes defined as `4cm` you may use each of these three commands. Unfortunately you must set height of such beside floats by hand. The lines which create described float row:

```
\begin{figure}\CenterFloatBoxes  
\begin{floatrow}  
\hsize1.098\hsize  
  \fcapside[\FBwidth] [4cm]  
  ...  
  
  \ffigbox[\Xhsize] [4cm]  
  ...  
\end{floatrow}%  
\end{figure}
```

Figure 30. Float box (`\fcapside`) with beside caption in float row width float with caption below

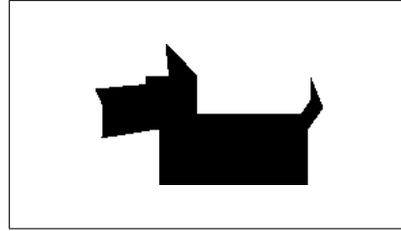


Figure 31. Float box (`\ffigbox`) width of rest float row space

The “mirror layout” (but not identical) looks like:

```
\begin{figure}\CenterFloatBoxes
\begin{floatrow}
  \ffigbox[1.2\FBwidth][4cm]
  ...

  \hsize\Xhsize
  \fcapside[\FBwidth][4cm]
  ...
\end{floatrow}%
\end{figure}
```

The figures 32–33 use following float style settings:

```
\floatsetup[figure]
{style=Boxed,frameset={\fboxsep4pt},captionskip=5pt,
 capposition=bottom,objectset=centering,capbesidewidth=none,
 capbesideposition=inside,capbesidesep=enskip,margins=centering,
 capbesideframe=yes}
```

9.1.8 Photo-Album-like Layouts

Another example of miscellaneous float row (figures 34–36, and, “mirror layout” — 37–39) were created by following lines:

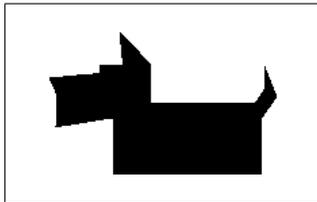


Figure 32. Float box (`\ffigbox`) width of rest float row space



Figure 33. Float box with beside caption (`\fcapside`) in mirror float row width float with caption below

```

\begin{figure}[t]\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}

```

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]
...%
\vskip0pt}\floatrowsep

\ffigbox[\Xhsize] [7cm-11pt]
...

```

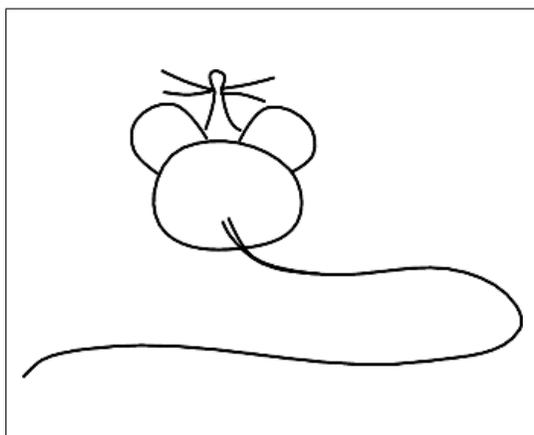


Figure 34. Float box in photo-album-like layout: alone in left column



Figure 35. Float box in photo-album-like layout: upper float in right column

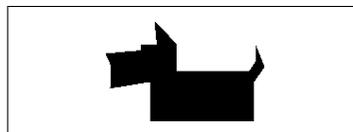


Figure 36. Lower float in right column



Figure 37. Float box in photo-album-like layout: upper float in left column



Figure 38. Float box in photo-album-like layout: lower float in left column

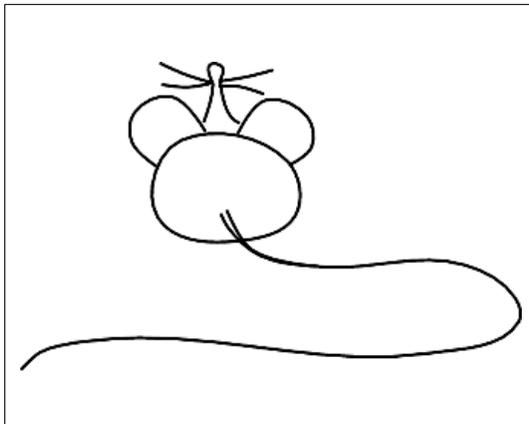


Figure 39. Float box in photo-album-like layout: alone in right column

```
\end{floatrow}%
\end{figure}
```

Note that in second example with “mirror” layout there was used trick with $\langle height \rangle$ definition—caption of float in the left column is one line longer, so for the right column height of float was reduced by 11pt— $\backslash baselineskip$ for $\backslash small$ size¹ (here were used `calc` package possibilities). The $\backslash vtop$ of left column ends with $\backslash vskip 0pt$, otherwise you get fanny unwanted layout.

In both examples for two floats one above another was cancelled $\backslash floatrowsep$ code inside $\backslash vbox/\backslash vtop$.

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 $\backslash Xhsize$ and/or $\backslash floatrowsep$.

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?).

9.2 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 and unnatural). The samples have not 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.

¹You get correct height with such calculations for beside floats with `caption`’s version not older than 3.0d.

The list of samples:

`frsample01.tex` all possible combinations of predefined `floatrow` styles for captions above/below floats with foot material; there were created plain float environments and `floatrows`, also there were created 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` different tests with tables;
`frsample04.tex` sample with fancy layout with usage of beside captions;
`frsample05.tex` one-column facing layout; miscellaneous caption settings.

The next bundle of samples is a few file-headers with different preamble which run the same file with various float layouts. For these examples there was created a new float type of float `textbox`—which 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.

There is also added [temporary] sample file `sample-longtable.tex` with `longtable` test of defined command `\endlasthead` which defines captions for continued and last pages of long table in three possible ways.

10 Obsolete Commands

`\floatsetup`, renamed or deleted after version 0.1b.

Removed or changed commands

Command	Changed to
<code>\renewfloatstyle</code> , <code>\newfloatstyle</code> , <code>\definefloatstyle</code>	<code>\DeclareFloatStyle</code> —this command uses <code>\floatsetup</code> mechanism
<code>\restorerestylefloat</code>	removed
<code>\captionskip</code>	command, not a skip
<code>\floatfootskip</code>	command, not a skip

Commands, replaced by keys

Deleted Command	Key Analog
<code>\floatobjectset</code>	in current version <i>do not used for definition of object settings</i> , use key <code>objectset=</code>
<code>\alignsidecaption</code>	<code>capbesideframe=yes</code>
<code>\capbesidecenter</code> , <code>\capbesidetop</code> , <code>\capbesidebottom</code> , <code>\capbesideinside</code> , <code>\capbesideoutside</code> , <code>\capbesideleft</code> , <code>\capbesideright</code>	<code>capbesideposition=center</code> <code>capbesideposition=top</code> <code>capbesideposition=bottom</code> <code>capbesideposition=inside</code> <code>capbesideposition=outside</code> <code>capbesideposition=left</code> <code>capbesideposition=right</code>
<code>\floatrowsep</code> , <code>\floatcapbesidessep</code>	in current version <i>do not used for definition of separation material</i> , use keys <code>floatrowsep=</code> <code>capbesidessep=</code>
<code>\FBcenter</code> , <code>\FBleft</code> , <code>\FBright</code> , <code>\FBnormal</code>	<code>margins=center</code> , <code>margins=raggedright</code> , <code>margins=raggedleft</code> , <code>margins=center</code> ,
<code>\setfloatstyle</code>	<code>style=</code>
<code>\Setframe</code> <code>\setframe</code>	use <code>framestyle=</code> and <code>frameset=</code> keys
<code>\setrules</code>	use <code>precode=</code> , <code>postcode=</code> , <code>midcode=</code> (also <code>rowprecode</code> and <code>rowpostcode</code>) keys

Renamed keys

Key	Changed to
<code>attachedcapstyle=</code>	<code>relatedcapstyle=</code>
<code>floatstyle=</code>	<code>style=</code>
<code>floatfont=</code>	<code>font=</code>
<code>putcaptionbeside=</code>	<code>capposition=beside</code>
<code>besidecapposition=</code>	<code>capbesideposition=</code>
<code>besidecapwidth=</code>	<code>capbesidewidth=</code>
<code>besidecapframe=</code>	<code>capbesideframe=</code>
<code>floatmarginssset=</code>	<code>margins=</code>
<code>besidecapsep=</code>	<code>capbesidessep=</code>
<code>Precode=</code>	<code>rowprecode=</code>
<code>Postcode=</code>	<code>rowpostcode=</code>
<code>framereduce=</code>	<code>framefit=</code>
options of <code>objectset=</code> and <code>margins=</code> <code>flushleft</code> , <code>flushright</code> , <code>center</code>	options of <code>objectset=</code> and <code>margins=</code> (for unification with analogous key options in <code>caption</code> package) <code>raggedright</code> , <code>raggedleft</code> , <code>centering</code>