

The floatrow package^{*}

Olga Lapko
Lapko.0@g23.relcom.ru

2007/08/24

Abstract

This package was created as extension of the float package. The floatrow package borrows core code from the float¹ and rotfloat² packages, so you *must not* load these packages.

The float package has a good mechanism for the creation (and easy modification) of common layout for all floats without adding any repeated code in the document; besides, this package allows to create new float type; it deals only with alone (plain) combinations “object (float contents)—caption”. The rotfloat package changes environments of rotated floats (rotating package) to adapt them to float’s settings.

Current package extends these possibilities and, at last, it allows:

- to use mechanism, borrowed from float package, for creation of new float type;
- to change width of float box, also 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 and/or modify special layout for each type of float and for different positioning of float and its components, e.g. two-column or rotated float.

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

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

^{*}This file has version number v0.2a, last revised 2007/08/24.

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

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

Document Terminology

- float** could include *object*, *caption*, and *foot material*; *float* is created by `figure` or `table` environments (*plain float*), or by `\floatbox` command and its modifications;
- float type** means standard environments `figure` and `table`, also their subtypes, like e.g. `wrapfigure` (`wrapfig` package), `sidewaysfigure` (`rotating` and `rotfloat` packages), `longtable` (`longtable` package) etc.;
- object** tabular or graphics, as contents of `table` (`table`) or `figure` (`figure`) 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).

Frequently Appeared Design

Caption	here! (option H)	sec. 4.1.3
above float (table’s object, <code>\ttabbox</code>)	row (<code>floatrow</code> env.) .	<i>Intro</i> , sec. 1.3
.	float(box) in the row occupies the	
beside float (figure’s object,	rest space <i>see</i> float width, the rest	
<code>\fcapside</code>) . . .	space of the row	
width equals to <code>longtable</code> ’s	floats of different types side by side	
(<code>LTcapwidth=key</code>)	18, 16
width equals to object’s <i>see</i> float box	like in plain \LaTeX (<code>\RawFloats</code>)	
width equals to object’s	sec. 1.4
Creation of new float type	width	
(<code>\DeclareNewFloatType</code>)	option in <code>\floatbox</code> commands	
.	sec. 1.2
Layout of Float types (<code>\floatsetup</code>)	equals to object’s (option <code>\FBwidth</code>	
Float	(<code>\floatbox</code>)) . .	<i>Intro</i> , sec. 1.2
box (<code>\floatbox</code>)	the rest space in the row (option	
figure box (<code>\ffigbox</code>)	<code>\Xhsize</code> (<code>\floatbox</code>)) . . .	14
.	wrapped	sec. 5.4–5.6
table box (<code>\ttabbox</code>)	Footnote inside float	sec. 1.5
.	footnote mark (<code>\mpfootnotemark</code>)	18
rotated (<code>sideways.. env.</code>) . .	Legend-like macro (<code>\floatfoot</code>)	sec. 1.5.1
empty (special) page style	Subfloat	
(<code>\emptyfloatpage</code>)	subcaption above	48
placing on the facing pages . . .	subfloat label beside	49

Contents

1	Macros for Building Floats	7
1.1	Introduction	7
1.2	The <code>\floatbox</code> Macro	9
1.2.1	Creation of Personal Commands for Float Boxes	12
1.3	Building Float Row	12
1.3.1	Mixed Row	15
1.4	Running Floats In The Raw \LaTeX mode	16
1.5	Usage of Footnotes Inside Float Environment	18
1.5.1	The Legend-Like Macro	19
1.6	Fine Tuning of Vertical Spaces of Float	20
1.7	Do not write that with <code>floatrow</code> package	20
2	Float Layout Settings	22
2.1	Floatsetup Keys	24
2.1.1	Float Style	24
2.1.2	Font settings	26
2.1.3	Position of Caption and of Other Elements	27
2.1.4	Defining Float Foot Position (Legends and Footnotes)	30
2.1.5	Defining Vertical Alignment	30
2.1.6	Facing Layout	31
2.1.7	Defining Float Margins and Object Settings	31
2.1.8	Defining Width of Object	32
2.1.9	Defining Float Separators	33
2.1.10	Defining Float Rules/Skips	33
2.1.11	Defining Float Frames	34
2.1.12	Defining Float Skips	35
2.1.13	Defining Float Footnote Rule's Style	36
2.1.14	Loading Style for Fancy Boxes	36
2.1.15	Managing Floats with <code>[H]</code> placement option	36
2.2	Settings for Current Float Environment	36
2.3	Clearing of Settings for Current Float Type	36
2.4	Temporary Clearing of All Float Settings	37
2.5	The Default Float Type Settings	37
2.6	Defining new options (Do it yourself!)	37
2.6.1	Float Style option (<code>style=</code>)	37
2.6.2	Option for Float Rules/Skips (<code>precode=</code> etc.)	38
2.6.3	Object Justification Option (<code>objectset=</code>)	39
2.6.4	Option for Float Box Alignment/Settings (<code>margins=</code>)	40
2.6.5	Float Separators Options (<code>floatrowsep=</code> , <code>capbesidesep=</code>)	41
2.6.6	Float Font Option (<code>font=</code>)	42
2.6.7	Option for Footnote Rule's Style (<code>footnoterule=</code>)	42
3	Creation of New Float Type	43
3.1	How to replace <code>\newfloat</code> with <code>\DeclareNewFloatType</code>	43

4	Borrowed code	44
4.1	float Package: Compatibility	44
4.1.1	How macros from float package work in floatrow	44
4.1.2	Printing of Float List [float]	44
4.1.3	The User Interface—[H] Placement Specifier [float]	45
4.1.4	The [H] Placement Specifier—Managing of Page Breaks	46
4.2	The rotfloat Package	47
5	Style Tandems	48
5.1	The caption Package	48
5.2	The subfig Package	48
5.2.1	Additions in floatrow	48
5.3	The longtable Package	52
5.3.1	Additions in floatrow	52
5.4	The wrapfig Package	53
5.5	The floatflt package	54
5.6	The picins Package	54
5.7	The rotating Package and sideways... Environment	55
5.8	The lscape Package and landscape Environment	57
5.9	The listings Package	57
5.10	The hyperref and hycap Packages	58
5.11	The Incompatibilities	58
6	Limitations	58
7	Acknowledgements	59
8	Appendix	61
8.1	Miscellaneous	61
8.1.1	Usage of Captionsetup Inside Floatbox Stuff	61
8.1.2	Predefined Beside Caption Width	62
8.1.3	Predefined Beside Caption Width with The Rest Space for Object	62
8.1.4	Caption Beside with Centered Object	63
8.1.5	Width Definition for Beside Caption—Object Box in Float Row	64
8.1.6	Caption Above/Below and Caption Beside at The Float Row	64
8.1.7	Photo-Album-like Layouts	66
8.2	Sample files	68
8.3	Obsolete Commands	69
8.3.1	The User Interface—New Floats [float]	69
8.3.2	The \floatsetup keys, renamed or deleted after version 0.1b	71

List of Figures

1	A simple figure box (\ffigbox)	7
2	A figure box (\ffigbox) with the width equal to graphics	8
3	Beside caption ('one-column' width)	8
4	Beside caption (occupies rest space of defined width beside float object)	9
5	The simple example of \floatbox	10
6	Beside caption (example width complex preamble in \floatbox)	11
7	Left beside figure (floatrow), the float box has width of graphic	14
8	Right beside figure (floatrow), occupies the rest space of row	14
9	Figure I in the row (floatrow), "column" width	15

10	Figure II in the row (<code>floatrow</code>), graphics width	15
11	Figure III in the row, float's width box has the half of the rest space of row	15
12	Figure IV in the row, occupies the rest space of row	15
13	A figure in the mixed row	16
14	A figure in raw \LaTeX 's mode	17
15	A figure in <code>\floatbox</code> and inside <code>floatrow</code> in raw \LaTeX 's mode	17
16	A figure caption, disappeared, wrong expected row	21
17	Caption beside framed object, (caption has width 4cm), aligned by top of object contents	29
18	Caption beside framed object, (object has width 9cm), aligned by top of frame	29
19	The left beside figure uses settings for vertical centering	31
20	The beside figure at the right side uses settings for vertical centering	31
21	Graphics with <code>floatwidth=0.3\hsize</code>	32
22	Caption beside graphics with <code>floatwidth=0.3\hsize</code>	33
23	The frame around graphics climbs out to the right and left sides	35
24	The frame around graphics fits to the width of float box (here: caption)	35
25	Plain figure in <code>MyBoxed</code> style	38
26	The left beside figure inside float row with defined row rules above and below	39
27	The beside figure at the right inside float row with defined row rules above and below	39
28	Alone figure with defined row rules above and below	39
29	The left beside figure inside float row with defined rules for float box	39
30	The beside figure at the right inside float row with defined rules for float box	39
31	Alone figure with defined rules above and below for float box	39
32	Multi-colored figure and beside caption	42
33	Subfloat row	50
34	Subfloat row (labels beside)	50
35	The graphic with subfloat labels	51
36	Wrapped plain figure (<code>wrapfig</code> package)	53
37	Wrapped figure in <code>\ffigbox</code> (<code>wrapfig</code> package)	53
38	Wrapped figure inside <code>floatingfigure</code> environment (<code>floatflt</code>)	54
39	Wrapped figure (<code>\parpic</code>); <code>\floatbox</code>	54
40	Figure inside <code>sidewaysfigure</code> environment	56
41	Figure in the row I, top of object box	60
42	Figure in the row II, bottom of object box	60
43	Figure in the row III, center of object box	60
44	Figure in the row IV	60
45	Float in the row with beside caption (graphic box has width of its contents)	61
46	Float in the row with beside caption (occupies rest space)	61
47	Beside caption, width equals to caption's text	62
48	The box of beside caption has width of caption contents (here: caption label)	63
49	Beside caption with "hidden" width (<code>=6cc</code>), object is centered at the full <code>hsize</code>	63
50	Left figure with beside caption in the row	64
51	Right figure with beside caption in the row	64
52	Float box <code>\fcapside</code> in float row beside <code>\ffigbox</code>	65
53	Float box (<code>\ffigbox</code>) width of rest float row space	65
54	Float box (<code>\ffigbox</code>) in mirror float row	65
55	Float box (<code>\fcapside</code>) in mirror float row	65
56	Photo-album-like layout: left float	66
57	Photo-album-like layout: upper right float	66
58	Photo-album-like layout: lower right float	66
59	Photo-album-like layout, mirror: upper left float	67
60	Photo-album-like layout, mirror: lower left float	67
61	Photo-album-like layout, mirror: right float	67

List of Tables

1	A small table with caption text above (<code>\ttabbox</code>)	7
2	Simple tabular with caption above (<code>\floatbox</code> with <code>\capttop</code> in preamble argument)	11
3	Beside table I long header	13
4	Beside table II with top aligned caption	13
5	A table in the mixed row	16
6	A table in raw \LaTeX 's mode	17
7	A table in <code>\floatbox</code> and inside <code>floatrow</code> in raw \LaTeX 's mode	17
8	Table with footnote	19
9	Table with foot material (e.g. legend)	19
10	A table caption must be placed above, wrong expect	20
11	A table caption, wrong expected row	21
12	Float layout styles	24
13	Long caption of table I with key <code>capposition=top</code>	27
14	Table II in the row with caption with key <code>capposition=top</code>	27
15	Long top-aligned caption of table I key <code>capposition=TOP</code>	28
16	Table II in the row with caption, aligned at the top line with key <code>capposition=TOP</code>	28
17	Two <code>\subtable</code> 's (created with <code>subfig</code> package)	49
18	Pascal's triangle. This is a re-styled \LaTeX table.	70

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

Caption's width equals to
object

The default layout of `\ffigbox` (figure 1) is similar to the plain `figure` environment. But if you set option `[\FBwidth]` like below:

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

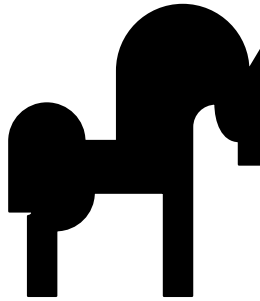


Figure 2. A figure with
the width equal to graphics
with long long long long
long multilined caption

You get caption width equal to the width of picture (figure 2).

The third macro—`\fcapside` (figure 3)—puts caption beside:

```
<preamble>
\usepackage[...capbesidesep=quad,facing=yes,
  capbesideposition=inside,...]{floatrow}[2007/06/24]
...
<preamble>
\begin{figure}
\fcapside
  {\caption{...}\label{...}}
  {...}
\end{figure}
```

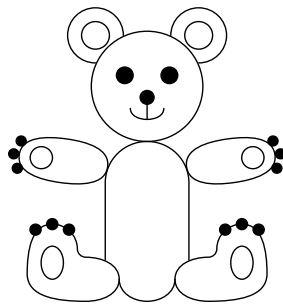


Figure 3. Beside caption (width of caption
equals to the width of object) and more text and
some more text and a bit more text and a little
more text and a little piece of text to fill space

The width of main text or predefined value of `\hsize` before `\fcapside` command, by default, divided into two columns, their width equals to the half of current `\hsize` (figure 3) excluding float margins and separation material between float and caption. One column is occupied by object, another—by caption and foot material (explications or legends and footnotes).

Caption's width equals to object

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

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

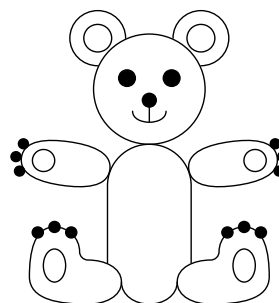


Figure 4. Beside caption (the caption text occupies the rest space of `\hsize`) and more text and some more text and a bit more text and a little more text and a little piece of text to fill space

The graphic column width will be equal to the graphics and caption occupies the rest space (see figure 4).

1.2 The \floatbox Macro

`\floatbox` 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⟩]{⟨captype⟩}[⟨width⟩][⟨height⟩][⟨vert pos⟩]
{⟨caption⟩}{⟨object⟩}1
```

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); `\captop` (to put caption above); or another systematic command (even with usage of `\captionsetup` and `\thisfloatsetup` see examples in documentation and appendix).

`⟨captype⟩` the type of float this command 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). Empty width option, `[]`, and option `[\hsize]` mean the same.

Caption's width equals to object

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

¹The 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.

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

$\langle height \rangle$ the height of object—caption box (in case of caption above or below object), or height of object box (if caption stays beside object);

$\langle vert pos \rangle$ vertical alignment of object contents in object's box in case of $\langle height \rangle$ argument has larger value than height of object contents, float placed inside `floatrow`.

$\langle caption \rangle$ text of caption and legends (you may use `\floatfoot` command (see page 19); you may also use `\footnote`/`\mpfootnotemark`/`\footnotetext` stuff for footnotes inside float, or `\floatfoot` command;

$\langle object \rangle$ Contents of float; you may use `\footnote`/`\mpfootnotemark`/`\footnotetext` stuff or `\floatfoot` command.

The commands `\ttabbox`, `\ffigbox`, and `\fcapside`, used in Introduction, abbreviate `\floatbox` command, which creates float boxes:

- the `\ttabbox` command we can replace with the code
`\floatbox[\captop]{table}[\FBwidth];`
- the `\ffigbox` one—with `\floatbox{figure}` (simplest definition); and
- the `\fcapside` one—with `\floatbox[\capbeside]{figure}`.

You see in example with figure 2 that you can change width for `\ffigbox` command to get caption (and also full float box) width equal to the width of object. Also you may create your own command based on `\floatbox` command and set float box's width equal to object's, see section 1.2.1 about this.

The three examples of float placing. First simplest example:

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

created the picture in the figure 5. The width of box equals to 4cm.



Figure 5. The simple example of `\floatbox`

The more preferable way for usage of \floatbox is to put table caption *above* table contents and set caption width equal to table width.

```
\begin{table}
\floatbox[\captop]{table}[\FBwidth]
{\caption{Simple tabular ...}\label{...}}
{\begin{tabular}{lcc}
...}
\end{table}
```

Table 2

Simple tabular with caption above (\floatbox with \captop in preamble argument)

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

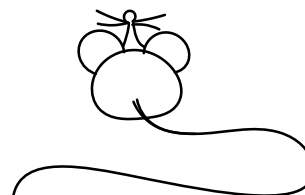
as shown in table 2. To fit size of caption (i.e full float box) to size of tabular environment there was used \FBwidth command in *<width>* argument.

The next example shows figure environment with beside caption. In this example *<preamble>* argument consists of rather complex definition, and \FBwidth in *<width>* 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
\captionsetup[capbesidefigure]{format=break,labelsep=none,
justification=raggedleft}%
\thisfloatsetup{capbesideposition=left}}]{figure}[\FBwidth]
{\caption{...}\label{...}}
{...}
\end{figure}
```

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



Please note that complex preamble, with more than one command option must be placed in group. (See section 2 about settings for floats.)

1.2.1 Creation of Personal Commands for Float Boxes

The usage of `\floatbox` command with options (which could be cumbersome) sometimes is rather complex. The Introduction demonstrates commands-abbreviations of this commands. You may define commands-abbreviations (or redefine existing) for your own purposes and include there some additional style definitions and 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 9 and other layout commands, like was shown in examples; you may try to add any other regular command (e.g. `\captionsetup` or `\thisfloatsetup` stuff);

<default width> the first reason of usage of this option is to put `\FBwidth` command: in case of caption above or below object (there could be defined `\nocapbeside` in *<preamble>* option) this command defines width of full float (object—caption) box (see figure 2); in the case of caption beside object (there must be defined `\capbeside` in *<preamble>* option) this command defines width of object (caption occupies the rest space, see figure 4).

You may put also any dimension like 6cm or `.75\textwidth` but usage of option by this way is much less reasonable.

Let's repeat three already defined commands-abbreviations:

```
\newfloatcommand{ffigbox}{figure}[\nocapbeside]
\newfloatcommand{fcapside}{figure}[\capbeside]
\newfloatcommand{ttabbox}{table}[\capttop][\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).

Your defined commands you may use in following way (example for `\ffigbox`):

```
\ffigbox[<width>][<height>][<vert pos>]
```

first option, *[<width>]*, cancels the default width settings, which were used during (re)definition of current command. The other two options are analogous to the `\floatbox`'s ones (see page 9).

1.3 Building Float Row

Floats of one type
side by side

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

This environment looks like:

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

The `floatrow` environment allows to put floats side by side. This environment creates necessary number of “columns” (usually two), where floats are placed. You may redefine the width of each float box, e.g. the table boxes 3 and 4 have the width of their contents (remember, the `[\FBwidth]` is default option of `\ttabbox`).

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

```
\preamble
\usepackage[...floatrowsep=qquad,...]{floatrow}
\floatsetup[table]{style=Plaintop}
\preamble
\begin{figure}
\begin{table}
\ttabbox
{\caption{...}\label{...}}
{...}

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

Table 3
**Beside table I with long long
long long long long and top
aligned caption**

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

Table 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

Since `\ttabbox` uses `\FBwidth` as default option, the caption width equals to tabular contents.

The next example with figures uses `[\FBwidth]` for the left float and `[\Xhsize]` for the right.

```

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

```

Float occupies the rest space
in the row

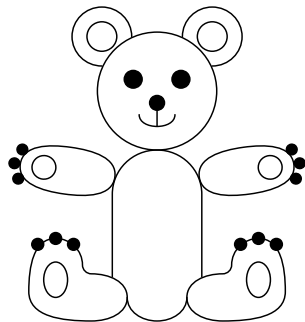


Figure 7. Left beside figure, the width of graphic



Figure 8. Beside figure at the right side of simple figure row, the box width occupies the rest space of row

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 `\width` argument like `\Xhsize/2` and then `\Xhsize` for two last `\floatbox`'es—the next example just uses it.

And here is more complex 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 `\width`), the last two occupy the rest horizontal space of page which was counted with command `\Xhsize` and `calc` package.

```

\preamble
\DeclareMarginSet{hangleleft}{\setfloatmargins
{\hskip-\marginparwidth\hskip-\marginparsep}{\hfil}}
\floatsetup[widfigure]{margins=hangleleft}
\preamble
\begin{figure*}
\begin{floatrow}[4]
\ffigbox
{\caption{Beside figure~I...}}{...}
{...}

```

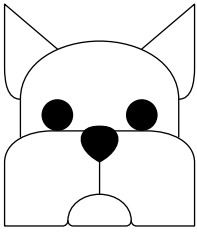


Figure 9. Figure I in the row (floatrow), “column” width



Figure 10. Figure II in the row (floatrow), graphics width

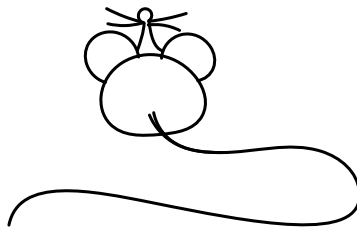


Figure 11. Figure III in the row, float’s width box has the half of the rest space of row

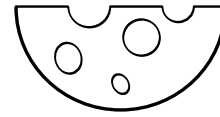


Figure 12. Figure IV in the row, occupies the rest space of row

```
\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 9–12. Please note that in all examples the vertical alignment of floats lays on the bottom of upper part (here: objects) of float and the top of lower part (captions).

1.3.1 Mixed Row

The problem in mixed row 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).

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

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

which align *full* float boxes by center, top or bottom lines. These macros were created by `\buildFBBBOX` 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 13 and table 5:

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

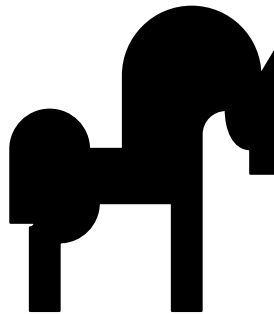


Figure 13. A figure in the mixed row

Table 5
A table in the mixed row

A	B
C	D

Note. Both figure and table boxes have got width equal to contents of objects. The `\ffigbox` command has optional argument `[\FBwidth]`, but `\ttabox` have not any option—it uses `[\FBwidth]` option as default (see definitions on page 12).

1.4 Running Floats In The Raw \LaTeX mode

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 do that by one of three ways.

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

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

And you'll get figure 14 and table 6¹.

¹That could work correct if you do not use any fancy float layout at least for one of floats.

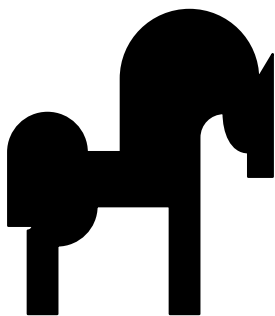
**Figure 14.** A figure in raw L^AT_EX's mode

Table 6

A table in raw L^AT_EX's mode

A	B
C	D

Compare this example with example in the section 1.3.1 and the following example.

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

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

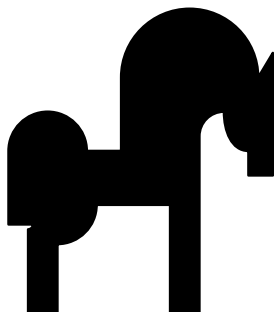
**Figure 15.** A figure in `\floatbox` and inside `floatrow` in raw L^AT_EX's mode

Table 7

A table in `\floatbox` and inside `floatrow` in raw L^AT_EX's mode

A	B
C	D

2. Canceling of `floatrow`'s behavior of *all floats of one type* is set outside any float environment, these settings are better for preamble. In this case the `\RawFloats` command needs optional argument with name(s) of float type. You may set that by two ways:

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

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

The second way, with second optional argument, cancels `floatrow`'s behavior for mentioned float “subtype(s)” of *one* float type in second optional argument you may use `float`, `widfloat`, `rotfloat`, `widerotfloat`—the meaning of this options analogous to options of `\floatsetup` macro (see section 2) but you may use here only options which include “float” word.

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

The `floatrow` environment (section 1.3) and commands of `\floatbox` stuff (section 1.2) still work after `\RawFloats`' and `rawfloats=` settings (see example width figure 15 and table 7)

1.5 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` in standard \LaTeX always creates the sign of main text footnote. For these cases current package offers `\mpfootnotemark` macro instead of `\footnotemark`¹.

Footnotemark
inside float

```

<preamble>
\floatsetup{...,footnoterule=limited,footskip=\skip\footins,...}
<preamble>
\begin{table}
\ttboxed
{\caption{...}\label{...}}%
{\begin{tabular}{...}
\hline
First row & \phantom{01} & 2\mpfootnotemark[1] \\
Second row & \phantom{06}\mpfootnotemark[1] & 4\mpfootnotemark[1] \\
...
\hline
\end{tabular}}%
\footnotetext[1]{Even numbers.}}
\end{table}

```

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

Table 8

Table with footnote

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

^aEven numbers.

The `\floatbox` macro uses special own definition of footnote rule (the `footnoterule=` key) and skip before footnotes and explications or legends (`footskip=`). (See also page 36 for variants of footnote rule.)

1.5.1 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 text justification, but star form (`\floatfoot*`) prints its contents as plain unindented paragraph (see table 9).

For defining of explication font use `footfont=` option (page 27) in `\floatsetup`. You may try to define special settings for float foot using `\captionsetup[floatfoot]`.

```
\begin{table}
\caption{\caption{...}\label{...}}
\begin{tabular}{...}
...
\end{tabular}%
\floatfoot*{'Data I' column ...}
\end{table}
```

Table 9

Table with 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—2ⁿ values

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

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

3) Foot material (footnotes and text in `floatfoot`) can be placed in several variants: at the very bottom of float box, below caption (even if caption is above float object). See page 30 and sample file `frsample01.tex`. In case of caption beside float object, footnotes and foot text always placed below caption.

1.6 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 `\FBskip` and `\FBbskip`. For example define

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

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

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

1.7 Do not write that with floatrow package

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

```
<preamble>
\usepackage{floatrow}
<preamble>
\begin{figure}
  \caption{A table caption...}
  \centering \begin{tabular}{cc} A & B \\ C & D \end{tabular}
\end{figure}
```

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

A	B
C	D

Table 10

A table caption must be placed above, wrong expect

If you want, e.g. to put table captions above its contents, you must write `\floatsetup[figure]{style=plaintop}` in the preamble (see also section 2 about `\floatsetup` mechanism). Another example. If you put beside floats by following way:

```
<preamble>
\usepackage{floatrow}
<preamble>
```

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.

```

\begin{figure}
\begin{minipage}{0.5\textwidth}
\centering ...
\caption{A figure caption, disappeared, wrong expected row}
\end{minipage}
\begin{minipage}{0.5\textwidth}
\captionof{table}{A table caption, wrong expected row}}
\centering
\begin{tabular}{cc}
A & B \\
C & D
\end{tabular}
\end{minipage}
\end{figure}

```

you also won't get expected result.



A	B
C	D

A table caption, wrong expected row

Table 11

Read back the sections 1.3.1 and 1.4 and correct the error.

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 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 modification for captions and special settings for each type of float is supported by caption package, version 3.x.

The layout settings of floatrow package are built similarly to the settings from the caption 3.x package and use `\floatsetup`¹ mechanism which is analog to `\captionsetup` one².

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

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

You may write

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

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

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

The lines above declare the `boxed` style and the `\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 the floats of one type, you can use optional argument with name of necessary float type. The following command

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

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

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, wrapped floats. 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 8.3).

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

in the lists below decreases from previous item to next one (examples based on `figure` environment)¹:

- Wide or two-column floats (e.g. `figure*`: you may load special settings for starred environments even in one-column document; that could be different margins for floats, different alignment of captions):
 - `\floatsetup[widfigure]`—the “strongest” settings; if they absent, package uses settings from next item;
 - `\floatsetup[widfloat]`—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 37);
- Wrapped floats (`wrapfigure`, used with `wrapfig` package):
 - `\floatsetup[wrapfigure]`;
 - `\floatsetup[wrapfloat]`;
 - `\floatsetup[figure]`;
- Rotated floats (in environments, like `sidewaysfigure`, used with rotating package):
 - `\floatsetup[rotfigure]`;
 - `\floatsetup[rotfloat]`;
 - `\floatsetup[figure]`;
- Wide or two-column rotated floats (`sidewaysfigure*`)²:
 - `\floatsetup[widerotfigure]`;
 - `\floatsetup[widerotfloat]`;
 - `\floatsetup[rotfigure]`;
 - `\floatsetup[rotfloat]`;
 - `\floatsetup[figure]`;
- Floats with beside captions (please note, that settings in these options are limited, see below):
 - `\floatsetup[capbesidefigure]`;
 - `\floatsetup[capbesidefloat]`;
 - setting of outer environments from previous items: `sidewaysfigure*`, `sidewaysfigure`, `figure*` and `figure`.

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.

¹In the case of the boolean key `rawfloats=` is switched to `true`, the options below for float layout are canceled.

²The settings for wide float (`widfloat`, `widfigure`) removed—use settings for `widerotfloat` and—here—`widerotfigure`

2.1 Floatsetup Keys

2.1.1 Float Style

`style` The *float style* could include alignment of float contents, margins, boxes or lines, if necessary, and other options. The *float style* is specified by following way:

`style=<float style name>` , the name of the *<float style name>* option you may take from table 12.
... You may create your own options with `\DeclareFloatStyle` command, see page 37.

As you may see in the table 12, there are included all float styles which emulate co-named ones from the float package.

Please note, that usage of `style=` key for floats with beside captions, i.e. using `\floatsetup` settings with options like, e.g., `[capbesidefigure]` or `[capbesidefloat]` cancels settings for beside position of caption. Usually the floats with beside captions can set different alignment of captions, so the radical changes for float layout with beside caption in current version could get wrong results (see section 2.4 how to solve this problem).

The caption package uses its own settings and names for caption layout styles. The caption’s ruled style is the only one from float package, which was predefined in caption package. To use it, write

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

Table 12

Float layout styles

Style	\floatsetup keys	Description
Offered by floatrow package		
<code>plain</code> ¹²³	<i><none></i>	The style <code>plain</code> is standard L ^A T _E X’s layout. Put captions always below float object’s contents.
<code>plaintop</code> ¹	<code>capposition=top</code>	The style <code>plaintop</code> is the same as <code>plain</code> style, but puts captions above float object’s contents—this style is analog of the co-named style from the float package. Capitalized form, <code>Plaintop</code> , aligns captions of the floats, which were placed in the one row (in the <code>floatrow</code> environment), by top line (see example on the page 13)
<code>Plaintop</code>	<code>capposition=TOP</code>	

Continued on next page

Table 12

Float layout styles (Continued)

Style	\floatsetup keys	Description
ruled ¹³	capposition=top, precode=thickrule, midcode=rule, postcode=lowrule, heightadjust=all	The first style, ruled, emulates co-named style from the float package. It places thick rule above float box, and thin rules between caption and object and below float. Rules are separated from contents by small 2pt skip. Capitalized form, Ruled, aligns captions of the floats, which were placed in the one row (in the floatrow environment), by top line (see example on the page 60)
Ruled ²	style=ruled, capposition=TOP	
boxed ¹²³⁴	captionskip=2pt, framestyle=fbox, heightadjust=object, framearound=object	The first style, boxed, emulates (again) co-named style from the float package. The width of object equals to the \textwidth, predefined \hsize, or the width in \floatbox's option; frame climbs out to the right and left sides (see example on the page 35). Frame separation and rule width equal to current \fboxsep and \fboxrule settings. (Default values are 3pt and .4pt consequently.) In capitalized form, Boxed, the frame around object fits the \textwidth, predefined \hsize, or the width in \floatbox's option; the width of object is reduced to fit inside frame (see example on the page 38). Uppercase form, BOXED, draws frame which fits to current \textwidth, predefined \hsize, or the width in \floatbox's option, but around all float elements: caption, object and foot material (see example on the page 53)
Boxed ²³	style=boxed, framefit=yes	
BOXED ²³	framestyle=fbox, framefit=yes, heightadjust=all, framearound=all	
Offered by fr-fancy package. They also need fancybox package.		
shadowbox ⁴	style=boxed, framestyle=shadowbox	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \shadowbox from fancybox package (see example on the page 50). Besides \fboxsep and \fboxrule, there is added parameter \shadowsize—the width of shadow, default is 4pt
Shadowbox	style=Boxed, framestyle=shadowbox	
SHADOWBOX	style=BOXED, framestyle=shadowbox	

Finished on next page

Table 12

Float layout styles (Finished)

Style	\floatsetup keys	Description
doublebox ⁴	style=boxed, framestyle=doublebox	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \doublebox from fancybox package. The frame shape control \fboxrule and \fboxsep parameters
Doublebox	style=Boxed, framestyle=doublebox	
DOUBLEBOX	style=BOXED, framestyle=doublebox	
Additional float styles. They also need fancybox package.		
wshadowbox ⁴	style=boxed, framestyle=wshadowbox	The same as boxed, Boxed and BOXED consequently. The \fbox frame changed to \wshadowbox, based on \shadowbox (but drops white shade from frame, or draws edges of “second copy”) from fancybox package (see example on the page 54), you may use the same frame parameters like shadowbox style.
Wshadowbox	style=Boxed, framestyle=wshadowbox	
WSHADOWBOX	style=BOXED, framestyle=wshadowbox	

When there is set float style with frame around object is fitted to the box 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 9), the width of all float other elements in this case enlarged to get width of framed object.

¹The styles co-named and analogous to float package style.

²This style used in the sample file frsmaple01.tex

³This style used in the sample file frsmaple02.tex

⁴During usage of these styles in floatrow environment you ought to enlarge space between floats, using key floatrowsep.

2.1.2 Font settings

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

Available font setting options:

- scriptsizeVery small size
- footnotesizeThe size usually used for footnotes
- smallSmall size
- normalsizeNormal size
- largeLarge size
- LargeEven larger size
- upUpright shape
- itItalic shape
- slSlanted shape
- scSMALL CAPS SHAPE
- mdMedium series

bf **Bold series**
rm Roman family
sf Sans Serif family
tt Typewriter family
... You may create your own options with `\DeclareFloatFont` command, see page 42.

You may set font for float object like

`font=small`

this option is used in current documentation, or

`font={small,sf}` .

footfont Defines font for legends or explications (defined by `\floatfoot` command, see section 1.5.1). 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`.

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 36):

Caption above
table object

top caption above object;
TOP caption above object and also aligned by top line in float row. For example the `Plaintop` style is the variant of `plaintop` where used `capposition=TOP` settings, see tables 15–16;
bottom caption below object;
beside caption beside object.

Compare two examples:

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

Table 13

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

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

Table 14

Table II in the row with caption

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

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

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

Table 15
**The table I in the row with long, long,
long, long, long, long caption, aligned
by the top line**

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

Table 16
**Table II in the row with caption, aligned at
the top line**

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

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

`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.

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}` .

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

`capbesidewidth` Defines width of beside caption. This option could be more preferable in settings for one environment, see about `\thisfloatsetup` on the page 36. You may set:

`capbesidewidth=4cm` .

If you'll write `capbesidewidth=none` or `capbesidewidth=sidefil` (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 4 on the page 9).

capbesideframe This boolean key declares whether the beside caption stays near the 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**). This key was switched on in the example on the page 50.

```

\preamble
\floatsetup[figure]{style=Boxed,frameset={\fboxsep8pt},
  objectset=justified,capbesideframe=no,capbesideposition={right,top}}
\captionsetup[figure]{...,strut=no}
\preamble
\thisfloatsetup{capposition=beside,
  floatwidth=sidefil,capbesidewidth=4cm}
\begin{figure}
...
  \caption{...}\label{...}
\end{figure}

```

Here goes first line of text And more text and some more text and a bit more text and a little more text and a little piece of text to fill space
 There goes second line of text
 Thence goes third line of text And more text and some more text and a bit more text and a little more text and a little piece of text to fill space
 Hence goes fourth line of text

Figure 17. Caption beside framed object, (caption has width 4 cm), aligned by top of object contents

```

\preamble
\floatsetup[figure]{...,capbesideframe=yes,...}
...
\preamble
\thisfloatsetup{capposition=beside,
  floatwidth=9cm,capbesidewidth=sidefil}

```

Here goes first line of text And more text and some more text and a bit more text and a little more text and a little piece of text to fill space
 There goes second line of text
 Thence goes third line of text And more text and some more text and a bit more text and a little more text and a little piece of text to fill space
 Hence goes fourth line of text

Figure 18. Caption beside framed object, (object has width 9 cm), aligned by top of frame

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

2.1.4 Defining Float Foot Position (Legends and Footnotes)

<code>footposition</code>	Defines position of <code>\footnote</code> 's and <code>\floatfoot</code> 's in float box with above/below captions.
<code>default</code>	if caption above float object foot material is placed below float object otherwise—below caption;
<code>caption</code>	always placed below caption;
<code>bottom</code>	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

<code>heightadjust</code>	Defines whether the common maximum height of objects or/and captions in the <code>floatrow</code> environment will be used. It has following options
<code>all</code>	adjust both caption and object heights (e.g. for styles <code>Ruled</code> and <code>BOXED</code>);
<code>caption</code>	adjust caption heights (e.g. for <code>Plaintop</code> style);
<code>object</code>	adjust object heights (e.g. for <code>Boxed</code> style);
<code>none</code>	nothing to be adjusted;
<code>nocaption</code>	no adjusting for captions;
<code>noobject</code>	no adjusting for objects;

You may define height adjustment even as followed:

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

<code>valign</code>	Defines vertical alignment of float objects in <code>floatrow</code> if <code>heightadjust=all</code> or <code>heightadjust=object</code> , or <code>\floatbox</code> stuff uses <i>height</i> argument with value, which differs from the height of object. The options of this key analogous to vertical alignment option in <code>minipage</code> environment and <code>\parbox</code> command. Default option is <code>c</code> (centered vertical alignment).
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

The settings `heightadjust=object`, `valign=c` were used in the example with figures 19 and 20. (Please look at the width options of `\ffigbox` commands: you may set the widths in this way with `calc` package.)

```
<preamble>
\floatsetup{...,heightadjust=object,valign=c,...}
<preamble>
\begin{figure}
\begin{floatrow}
\ffigbox[\FBwidth+2cm]
```

```

{...}
{\caption{The left ...}\label{...}}%

\ffigbox[\FBwidth+2cm]
{\caption{The beside ...}\label{...}}
{...}
\end{floatrow}
\end{figure}

```

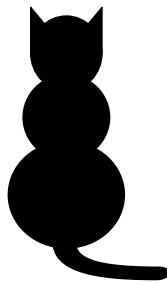


Figure 19. The left beside figure uses settings for vertical centering

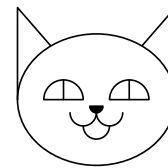


Figure 20. The beside figure at the right side uses settings for vertical centering

2.1.6 Facing Layout

facing This key defines whether facing layout is used for floats. This key works if `twoside` option switched on in document class. The combination of settings `facing=yes`, `capbesideposition=inside` (see above) is used in current documentation.

The most popular usage of `facing` key is in the case, when beside captions are placed to the inner sides of pages. The figures 3 and 4 at beginning of the document illustrate these options—`facing=yes`, `capbesideposition=inside`.

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 box flushed to the left;
raggedleft   float box flushed to the right.
...           You may create your own alignment settings with
              \DeclareMarginSet command, see page 40.

```

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

justified Blocks (in the case of a picture or text box) moved to the left, the text aligned as a normal paragraph (without indentation). (This is the default.)

centering Blocks centered, each line of the object text will be centered.
raggedright Blocks moved to the left, each line of the text shoved to the left margin.
RaggedRight As in previous item, each line of the text shoved to the left margin, too. But this time the command `\RaggedRight` of the `ragged2e` package will be used to achieve this. This difference is that this time the word breaking algorithm of \TeX will work inside the text.
raggedleft Blocks moved to the right, each line of the text shoved to the right margin.
RaggedLeft Analogous to `RaggedRight`.
... You may also create your own settings with `\DeclareObjectSet` command (see page 39)

2.1.8 Defining Width of Object

floatwidth It is used for redefinition of width of objects. This key, similar to `captionwidth=`, page 28, could be used at first for settings of one float environment:

```
floatwidth=.3\hsize
```

or

```
floatwidth=7cm
```

(see page 36 about settings for current float environment and `\thisfloatsetup`).

```
<preamble>
\floatsetup[figure]{capposition=bottom}
<preamble>
\thisfloatsetup[figure]{floatwidth=.3\hsize,capposition=bottom}
\begin{figure}
  \includegraphics[width=\hsize]{Bear}
  \caption{...}\label{...}
\end{figure}
```

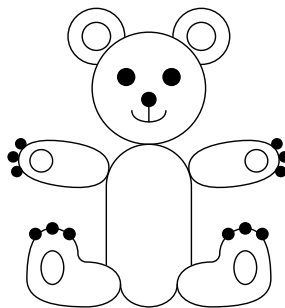
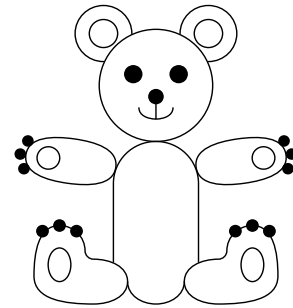


Figure 21. Graphics with
`floatwidth=0.3\hsize`

```
\thisfloatsetup{floatwidth=.3\hsize,capbesidewidth=sidefil,
  capposition=beside}
```


Figure 22. Caption beside graphics with `floatwidth=0.3\hsize`



If you use option `floatwidth=sidefil` for objects with beside captions (in the case of key `capbesidewidth=`, uses absolute value, like `capbesidewidth=4cm`) the object contents (instead of caption) occupy the rest space of float box (see figure 17 on the page 29 and appendix, figure 48 on the page 63)

2.1.9 Defining Float Separators

`floatrowsep` Sets separation material between beside float boxes in one row inside `floatrow` environment (see page 12).

`capbesidessep` 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> (default);
<code>quad</code>	horizontal skip = 1 em;
<code>qquad</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.
<code>...</code>	You may also create your own settings with <code>\DeclareFloatSeparators</code> command (see page 41)

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

2.1.10 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 60).

`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 60).

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 ruled style;
<code>rule</code>	rule of default thickness (.4pt), with 2pt vertical skips above and below—middle rule in ruled style;
<code>lowrule</code>	rule of default thickness (.4pt), with 2pt vertical skip above—rule below float box in ruled style;
<code>captionskip</code>	vertical skip equal to value, defined in <code>captionskip=</code> key (default).
<code>...</code>	You may create your own options with <code>\DeclareFloatVCode</code> command, see page 38.

2.1.11 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:

<code>colorbox</code>	colored frame, needs also color package; if not defined, there is used <code>\fbox</code> ;
<code>doublebox</code>	double frame, needs also fancybox package;
<code>shadowbox</code>	frame with shadow, needs also fancybox package;
<code>wshadowbox</code>	modified shadowbox frame (frame with “white shadow”), needs also fancybox package;

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

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

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

<code>object</code>	float object contents;
<code>all</code>	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 (see page 38 and all boxed figures in current documentation); 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`).

```
\floatsetup[figure]{framestyle=fbox,heightadjust=object,
    framearound=object,frameset={\fboxrule1pt\fboxsep10pt},
    framefit=no}%≈ style=boxed
\begin{figure}
\ffigbox[4cm]
{...}{\caption{...}}
```

`\end{figure}`

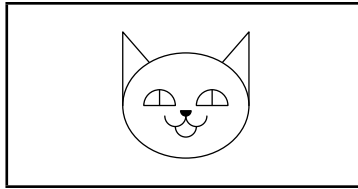


Figure 23. The frame around graphics climbs out to the right and left sides

```
\floatsetup[figure]{...,framefit=yes}%~ style=Boxed
\begin{figure}
\ffigbox[4cm]
{...}{\caption{...}}
\end{figure}
```

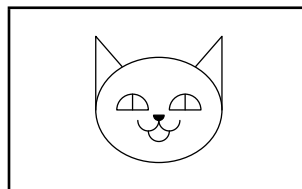


Figure 24. The frame around graphics fits to the width of float box (here: caption)

colorframeset Offered by `fr-fancy` package¹ and also needs `color` package: defines concrete color box (default is standard `\fbox`). You may use `\fcolorbox` as color box. You may define your color box like

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

2.1.12 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
```

The settings above are default. The settings of current documentation: `captionskip=5pt`.

footskip Defines vertical space before foot material and footnotes.

```
footskip=4pt
```

¹Use `fancyboxes=yes` option in `\usepackage` line, also you ought to load `color` package.

or

`footskip=\skip\footins` .

2.1.13 Defining Float Footnote Rule's Style

<code>footnoterule</code>	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: <code>\newcommand\frulemax{1in}</code>
<code>fullsize</code>	rule to full current text width.
<code>none</code>	Absent rule.
<code>...</code>	You may create your own options with <code>\DeclareFloatFootnoterule</code> command, see page 42.

2.1.14 Loading Style for Fancy Boxes

<code>fancyboxes</code>	This boolean key loads <code>fr-fancy</code> package. This key you may use only in optional argument in <code>\usepackage</code> line.
-------------------------	----------------------------------------------------------------------------------------------------------------------------------------

2.1.15 Managing Floats with [H] placement option

<code>doublefloataswide</code>	This boolean key redefines starred float in onecolumn layout like non-starred ones. (In this case the <code>[H]</code> option works in starred environments also; layout settings from <code>\floatsetup</code> as <code>widefloat</code> still work.)
<code>floatHaslist</code>	This boolean key adds values of penalties before and after this “anchored” float like in the list environment and cancels paragraph indentation, if no blank line appears after environment (see also section 46).

2.2 Settings for Current Float Environment

<code>\thisfloatsetup</code>	You may define some settings only for one float just before necessary environment. Command <code>\thisfloatsetup</code> could contain the same keys and options as in <code>\floatsetup</code> . It has only mandatory argument ¹ .
------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.3 Clearing of Settings for Current Float Type

<code>\clearfloatsetup</code>	If you want to get rid of parameters marked for an automatic use within a particular environment you can use the command ² :
-------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------

`\clearfloatsetup{<float type>}` .

where `{<float type>}`—types as `figure`, `widefloat` etc.

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

²Created as additional macro for `\clearcaptionsetup` macro, see also documentation of `caption` package about `\clearcaptionsetup` command

2.4 Temporary Clearing of All Float Settings

`\killfloatstyle` In some cases like mixed rows of floats where figure stays beside table you need to cancel layout of “foreign” float (see page 16). The `\killfloatstyle` command is used before any command of `\floatbox` stuff (see section 1.2). Other situation—layout with beside captions is quite different from other subtypes: `[figure]` option of `floatsetup` defined with `style=plain` and `[capesidefigure]` defined with `style=boxed`. In this case you may define your command, based on predefined `\fcapside`:

```
\newcommand\myfcapside{\killfloatstyle
\floatsetup[figure]{style=Boxed}\fcapside} .
```

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

2.5 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
capesideposition=inside
capesideframe=no
footposition=default
heightadjust=none
facing=no
margins=centering
objectset=centering
floatrowsep=columnsep
capesidessep=columnsep
precode=none
rawprecode=none
postcode=none
rawpostcode=none
midcode=captionskip
captionskip=10pt
```

2.6 Defining new options (Do it yourself!)

In the next few sections goes a list of commands, which define key options for `\floatsetup`.

2.6.1 Float Style option (`style=`)

`\DeclareFloatStyle` Defines new float style. Example shows definition of new float style `MyBoxed`. The figures 25, and some others in current documentation show result.

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

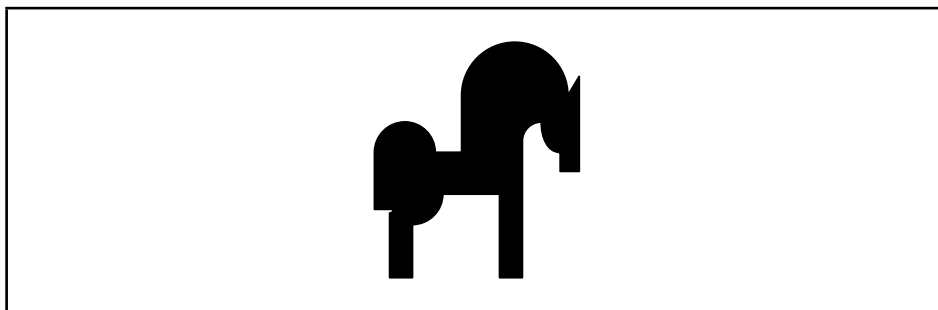


Figure 25. Plain figure in MyBoxed style

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

The same result you get with:

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

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

`\DeclareFloatVCode` This command defines the skip, rule or other analogous code above and below full float box and between caption above/below and object. The defined option you may use in `rowprecode`, `precode`, `midcode`, `postcode`, and `rowpostcode` keys.

```
<preamble>
\DeclareFloatVCode{grayruleabove}%
  {{\color{gray}\hrule height2.8ptdepth0pt\vskip4pt}}
\DeclareFloatVCode{grayrulebelow}%
  {{\color{gray}\vskip4pt\hrule height2.8ptdepth0pt}}
\floatsetup{...,
  rowprecode=grayruleabove,rowpostcode=grayrulebelow}
<preamble>
\begin{figure}
\begin{floatrow}
  \ffigbox
    {...}{\caption{The left ...}\label{...}}%

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

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



Figure 26. The left beside figure inside float row with defined row rules above and below



Figure 27. The beside figure at the right inside float row with defined row rules above and below



Figure 28. Alone figure with defined row rules above and below

```

<preamble>
...
\floatsetup{...,precode=grayruleabove,postcode=grayrulebelow}
<preamble>
...

```



Figure 29. The left beside figure inside float row with defined rules for float box



Figure 30. The beside figure at the right inside float row with defined rules for float box



Figure 31. Alone figure with defined rules above and below for float box

2.6.3 Object Justification Option (`objectset=`)

`\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.

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

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

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

or like (see also sample files)

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

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:

```
\DeclareMarginSet{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.

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 `\DeclareMarginSet` settings. For example, centering option was defined like:

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

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

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 ought to restore correct font size, family, shape (here) and series for caption contents and write¹:

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

or

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

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

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

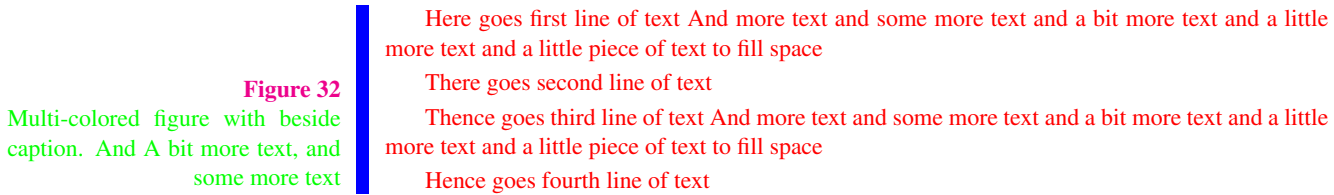
```
\DeclareFloatSeparators{columnsep}{\hskip\columnsep}
```

or more complex, using `color` package, and `\floatfacing` macro (do not forget the `facing=yes` key). You may use defined option by both `floatrowsep=` and `capbesidesep=` keys².

```
<preamble>
\DeclareObjectSet{colorred}{\parskip2pt\parindent15pt\color{red}}
\DeclareMarginSet{hangleleft}%
{\setfloatmargins
 {\hskip-\marginparwidth\hskip-\marginparsep}{\hfil}}
\DeclareFloatSeparators{colorsep}%
{\begingroup\color{blue}%
 \floatfacing{\hskip6pt\vrule width4.8pt\hskip6pt}%
 {\hskip6pt\vrule width4.8pt\hskip6pt}%
 \endgroup}
\floatsetup[widfigure]{margins=hangleleft, capbesidesep=colorsep,
 objectset=colorred,
 floatwidth=\textwidth, facing=yes}
\captionsetup[figure]{justification=justified,
 labelfont={magenta,bf}, textfont=green, labelsep=newline}
<preamble>
\begin{figure*}
...
\end{figure*}
```

¹The version 3.1 of caption package will correct that.

²You may use key options declared by `\DeclareCaptionLabelSeparator` command.



2.6.6 Float Font Option (font=)

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

To get red color for text in the example above, you may define the red color by following way:

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

and then

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

2.6.7 Option for Footnote Rule's Style (footnoterule=)

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

```
\DeclareFloatFootnoterule{limited}{\kern-3pt
\@tempdima.4\columnwidth
\ifdim\@tempdima>\frulemax\@tempdima=\frulemax\fi
\hrule width\@tempdima\kern2.6pt}
```

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, new captions are gathered in `lox`-file). This option is required in the case of creation of a separate float lists.

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 69). It consists of following code:

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

3.1 How to replace `\newfloat` with `\DeclareNewFloatType`

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

```
\newfloat{<type>}{<placement>}{<ext>}[<within>]}
```

which could be replaced with

```
\DeclareNewFloatType{<type>}%
{placement=<placement>,fileext={<ext>},within=[<within>]}
```

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.

Sections, signed with “[float]” were borrowed from float's documentation. The section about commands which describes setting commands of float package was moved in the section 8.3 (subsection 8.3.1, “The User Interface—New Floats [float]”).

4.1.1 How macros from float package work in floatrow

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

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

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

The `\floatname` command replaces the `name=` key of `\DeclareNewFlotaType` command. The `\floatplacement`—`placement=` key.

4.1.2 Printing of Float List [float]

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

```
\listof{<type>}{<title>}
```

`<type>` is the float type given in the `\newfloat` command. `<title>` is used for the title of the list as well as the headings if the current page style includes them. Otherwise, the `\listof` command is analogous to the built-in L^AT_EX commands `\listoffigures` and `\listoftables`.

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. See also section 4.1.1 about translation of these commands into `\floatsetup` command.

Floatrow note. Please remember that for each float type list you must set file extension where the entries of each float type gathered. Use key `fileext=...` in `\DeclareNewFloatType`. If two or more float types will have the same file for list of floats, you will get all floats in the first appeared `\listof`, and all next will be empty.

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

Anchored float Many people find L^AT_EX'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 L^AT_EX something along the lines of “put the float here if it's OK with you”. As it turns out, L^AT_EX 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 L^AT_EX to “put it HERE, period”. If there isn't enough space left on the page, the float is carried over to the next page together with whatever follows, even though there might still be room left for some of that. This style option provides the [H] specifier for newly defined classes of floats as well as the predefined `figures` and `tables`, thereby superseding `here`. David suggests that the `here` option be withdrawn from the archives in due course.

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

```
\begin{table}
\begin{tabular}{c}
\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 L^AT_EXed 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

Floatrow note. Please don't mix meaning of [H] and [h] options. Float with [h] and [!h] option, if succeed, appears *after completing line* of text, where it was appeared

¹For float package.

in the source file. That could be visible if you put float environment at the middle of paragraph.

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

4.1.4 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-temp*¹ 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 (surely, this command not for floats!).

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

`floatHaslist` This is the late attempts of backwards compatibility. The key, if true, uses list penalties, otherwise anchored float works without any penalty, i.e. like defined in `float` package and in `floatrow`’s version 0.0g and earlier. (I hope there could not much difference, unless you redefined penalties of lists.)

Also (added in version 0.1k with current key): since list environments do *not make indentation* in following paragraph, in the case of *no blank* line after environment, the “anchored” float environment does the same, if this option is true. Default value of `floatHaslist` is false (for backward compatibility with previous version 0.1j).

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

```
\renewcommand\floatHpenalties{\@beginparpenalty\@M}
```

or, with `listpen` package

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

you’ll never get page breaks before anchored float.

`\RestoreSpaces` There was added one simple command, the external alias of internal command `\@nobreakfalse`. It switches off `\if@nobreak` flag. The main (and most visible) usage

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

of this flag—managing of vertical spaces between two sectioning commands (it cancels the space before for next section command of the pair); spaces before list environments (cancels the space before list just after sectioning command) and other similar cases.

Usually this flag turned off at the next paragraph (or `\par` command), but in some cases this “turn off” cannot be happen in necessary point. The `\RestoreSpaces` command would help.

4.2 The 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 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.

5.1 The caption Package

Tested (and compatible) with version v3.0q.

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).

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]{\options}
```

or for wide or two-column figures:

```
\captionsetup[widelfigure]{\options}
```

The priority of `\captionsetup` optional arguments is similar to `\floatsetup` ones. In current examples `\captionsetup[widelfigure]` will be stronger than `\captionsetup[widelfloat]`. The priority and usage of “*float subtypes*” in optional arguments see on the page 23).

5.2 The subfig Package

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

5.2.1 Additions in floatrow

The example with `\subfloat`’s (table 17). 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 table]
{\begin{tabular}{|l|c|c|}
...\end{tabular}}}
```

Subcaption above subtable


```
\subfloat[Second table...]{\begin{tabular}{|l|c|c|}\n...\\end{tabular}%\\end{subfloatrow}}\\end{table}
```

Table 17

Two \subtable's (created with subfig package)

(a) First subtable

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

(b) Second subtable with long
long long subcaption

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

The subfloatrow 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 capbesidessep= keys (options of these keys defined by \DeclareFloatSeparators command, page 41).

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.)

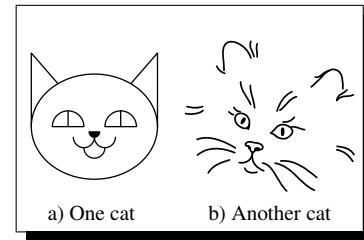
The next goes example with beside caption (see figure 33).

```
<preamble>\floatsetup[figure]{style=Shadowbox,capbesidessep=columnsep,capbesideframe=yes,capbesideposition={left,top}}\floatsetup[subfigure]{style=plain}\DeclareCaptionLabelFormat{list}{\bothIfFirst{#1}{ }#2}}\captionsetup[subfigure]{labelformat=list,justification=centerlast, strut=no}<preamble>\fcapside[\FBwidth]{\begin{subfloatrow}\subfloat[...\\label{...}]{...}\subfloat[...\\label{...}]{...}%\\end{subfloatrow}}{\caption{...}}
```

\sidesubfloat
Subcaption beside subfloat

Another addition in floatrow for subfloats is command, which puts subcaption label beside subfloat. The subcaption label always appears on the left side. The key

Figure 33. Beside caption vertically top aligned; fancy Shadowbox layout. There are two subfigures: 33a and 33b



`subcapbesideposition` `subcapbesideposition=` sets vertical alignment of beside subcaption and subfloat. The options analogous to options for `capbesideposition=` key:

<code>top</code>	subcaption label aligned to the top of object;
<code>bottom</code>	subcaption label aligned to the bottom of object;
<code>center</code>	subcaption label aligned to the center of float contents.

The figure 34 shows layout with subfloat labels beside.

```

<preamble>
...
\floatsetup[figure]{style=plain,subcapbesideposition=top}
<preamble>
\begin{figure}[H]
  \ffigbox[\FBwidth]
  {\begin{subfloatrow}
    \sidesubfloat[]{\dots\label{...}}%

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

```

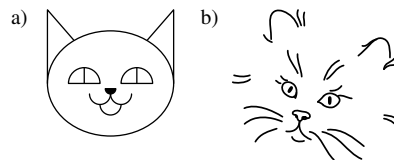


Figure 34. Beside caption vertically centered. There are two subfigures: 34a and 34b

`\subfloatlabel` There are cases when usage of something like `\subfloat[]{\label{...}}` is needed. The first case shows the figure 35—the funny picture environment where subfloat labels were `\put` as a part of subfigures. Other—when you use mechanism of `psfrag` package and replace text entries from PostScript file with \LaTeX ones. Unfortunately, the `subfig` package creates unnecessary spaces around alone subfloat label in the `\subfloat[]{\label{...}}` combination. The `fr-subfig` tries to fix this problem.

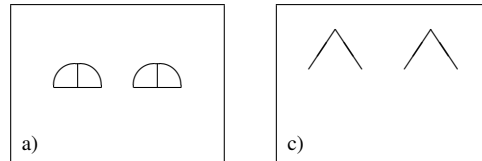
This command based on `\subfloat[]{\label{. .}}` sentence and puts alone subcaption label with necessary number. The full variant of `\subfloatlabel`

```
\subfloatlabel[(subfloat number)][(label entry)]
```

is the abbreviation of the following:

```
\setcounter{sub\@capttype}{(subfloat number-1)}
\subfloat[]{\label{(label entry)}}
(preamble)
...
\floatsetup[figure]{style=plain}
\DeclareCaptionListFormat{comma-separated}{#1,\,#2}
\captionsetup[subfigure]{listofformat=comma-separated}
(preamble)
\begin{figure}[h]
\fcapside[\FBwidth]
  {\unitlength2\unitlength\fbboxsep-.4pt
   \begin{picture}(90,30)(0,0)
     \put(0,0){\framebox(40,30)[bl]{} }
   ...
```

Figure 35. Here are two simple subfigures. First shows cat's eyes (35a), labeled with `\subfloat[]{} macro`; with `\subfloatlabel[3][subfig:bII]` sentence were labeled the cat's ears (35, c)



Please look at the definition of reference to the subfigure, key `listofformat=`. This format is used, in particular, by `\subref` command: the `\subref{subfig:bII}` gives reference to the subfigure 35, c.

5.3 The longtable Package

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

5.3.1 Additions in floatrow

There was added a patch to the `\LT@array` macro (longtable package)¹: 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`
Caption width equals to
longtable's

This key could have any value, like `5cm` or `\hsize`. In this cases this key does work of `\LTcapwidth` command. 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.

The addition with version 0.1k. There is added a beta-temp² package with simple additions, which allow creation of special head for last page of `longtable` environment and special foot for pages before last.

`\endlasthead`
`\endprelastfoot`

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

The footnote stuff inside `longtable` works like in main text and put text of footnotes at the bottom of page³. The command for legends or explications, `\floatfoot`, in current version inside `longtable` has emulation mode, and needs stuff, similar to `\noalign{\floatfoot{...}}`. Since default font definition for explications (`\floatfoot`) also set to `\footnotesize`, like for footnotes, you may put footnotes-emulations at this explication block, using `\mpfootnotemark` commands.

¹Thanks to A. Sommerfeldt for help to make this code compact.

²Again, like with `listpen` package, I hope that such support sooner or later could appear in `longtable` and think it is better to follow grammar of master-package for similar situations.

³See also `longtable` documentation.

5.4 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 unfloatable positioning, uppercase—floatable 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 `\FBskip` (`\FBbskip`) commands (see section 1.6).

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 inside `\usepackage` command).

Founded limitations.

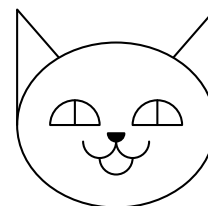
The usage of plain float environment in version 3.3 will not succeed with current version of floatrow—use `\floatbox` stuff. The version 3.6 allows usage of plain `wrap...`

Figure 36 Wrapped plain figure (wrapfig package)

Plain figure fails with package version 3.3

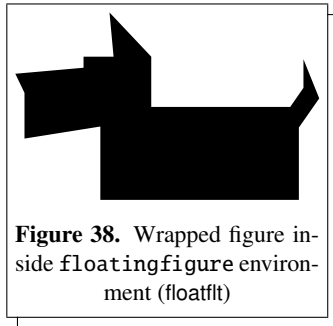


Figure 37. Wrapped figure in `\ffigbox` (wrapfig package)



environment with `plain` (or `ruled`) styles, but the framed styles, like `Boxed` (which use key `framefit=yes`, where text inside frames changes its `\hsize` to fit frames, fitted to defined `\hsize`) could work only with `\floatbox` macro, otherwise you'll get incorrect widths and layout.

5.5 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, like `\ffigbox`).

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

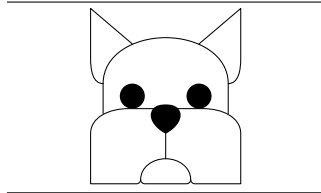
2) If you put `floatingfigure` environment just after `\...section` command you need (if you do not indentation after heads) to put `\noindent` for the first paragraph.

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).

5.6 The picins Package

Figure 39 Wrapped figure (`\parpic`); `\floatbox`



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.

5.7 The rotating Package and sideways... Environment

Tested with version v2.13 dated Sep. 1992.

There is example (figure 21) with rotated float, using `sidewaysfigure`.

```

\preamble
\floatsetup[rotfigure]{style=WSHADOWBOX}
\preamble
\begin{sidewaysfigure}\emptyfloatpage
\ffigbox[\FBwidth]
{...}
{\caption{Figure ...}}%
\label{...}}
\end{sidewaysfigure}%

```

Special settings.

You may create special settings for all rotated floats, which use `sideways...` environment (see page 23).

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 (command `\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).

Notes.

Continued rotated floats

- 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

```
\buildFBBOX{\vbox to\rottewidth\bgrou\vs\}{\egroup}
```

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

PostScript Graphic

Figure 40. Figure inside sideways figure environment

and before second one

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

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

Empty page style for rotated floats

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 \LaTeX macro `@outputpage`¹ and I hope that it could work.² Since this package uses `\label`—`\ref` mechanism, the `\floatpagestyle` command works after *second* \LaTeX run.

5.8 The lscape Package and landscape Environment

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

The example with usage of `landscape` environment from `lscape` package on the page 60, figures 41–44):

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

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

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

5.9 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 result³ for algorithm type of float.

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

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.

³Please note and read `caption` documentation: the co-operation of `caption3.x` and `listings` succeeds with version of last one not older than 1.2.

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.

5.10 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.

5.11 The Incompatibilities

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

```
ccaption, hvfloat, nonfloat, captcont, ccaption,
ftcapt, (hvfloat), multicap, topcapt, nonfloat
```

The known incompatibilities of floatrow package itself: 1) sidecap package¹: the floatrow package doesn't expands its layouts to `SCfigure` and `SCtable` environments. 2) ctable package; if you used to use ctable's tools, e.g. for tables, please set `\RawFloats[table]` in the preamble, and remember that commands like `\ttabbox` won't lose its strength (see also section 1.4).

6 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: some tools of the package use `\label—\ref` mechanism, thus, if you use float layout which demands common height of objects and/or captions in float row, you'll get correct result after second or more runs. If you change contents of float which change its height you must run \LaTeX twice or more times too.
Beside captions and other facing layout will appear correctly only after second \LaTeX 's run (sometimes you need to run more times).
- The caption 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 are allowed to use `\vspace` macro at the very end/very beginning of object contents for fine vertical tuning.)

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

- 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`.


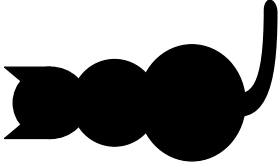
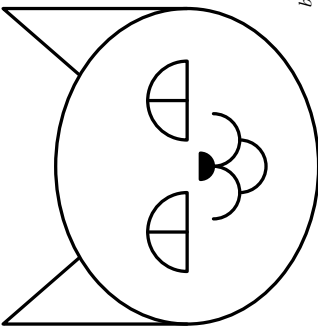
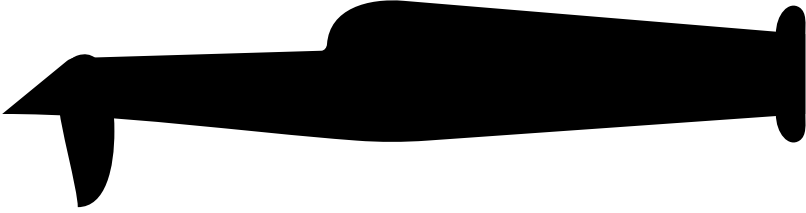
7 Acknowledgements

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)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.

<p>Figure 41 Figure in the row I, top of object box^a</p> <hr/> <p>^aThe picture was created with \qbezier macro</p>	<p>Figure 42 Figure in the row II, bottom of object box</p> <hr/> <p>There are all \qbezier macros and only two vertical lines</p>	<p>Figure 43 Figure in the row III, center of object box</p> <hr/> <p>The Mouse-animal image</p> <hr/> <p>^bLook at funny footnote mark!</p>	<p>Figure 44 Figure in the row IV</p>
			

8 Appendix

8.1 Miscellaneous

8.1.1 Usage of Captionsetup Inside Floatbox Stuff

Example of figures in row (figures 45 and 46). There were used predefined float commands `\fcapsideleft` and `\fcapsideright` with added `\captionsetup` and `\thisfloatsetup`:

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

The key `floatwidth=sidefil` 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).

Figure 45
Float box
(`\fcapsideleft`)
width of graphics

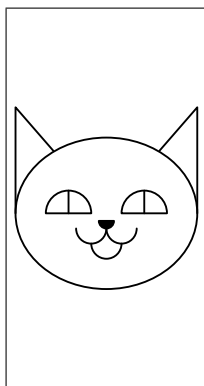
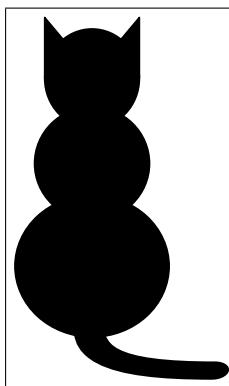


Figure 46
Float box
(`\fcapsideright`)
width of rest float
row space



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:

```
\hsize\Xhsize
```

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

The default `<width>` option of `\fcapsideleft` and `\fcapsideright` commands is `\FBwidth`, so for figure 46 on the right (created by `\fcapsideright` command) in the `<width>` 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.

8.1.2 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 47¹). 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 40).

```
<preamble>
\floatsetup[figure]{style=plain}
<preamble>
\begin{figure}
\floatbox[\capbeside\useFCwidth]{figure}[\FBwidth]
...
\end{figure}
```

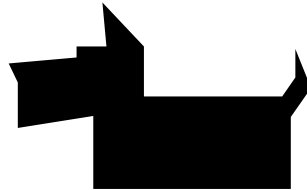


Figure 47. Beside caption

8.1.3 Predefined Beside Caption Width with The Rest Space for Object

The figure 48 uses following float style:

```
<preamble>
\renewlength\command\settowidth\Mylen{\captionfont\captionlabelfont
\figurename\ \thefigure}
<preamble>
\floatsetup[figure]
{style=Boxed,capposition=beside,objectset=centering,
```

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.

```
floatwidth=\columnwidth,capbesidewidth=\Mylen,
capbesideposition=left,capbesidessep=cicero,
margins=centering,capbesideframe=yes,
floatwidth=sidefil}
```

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

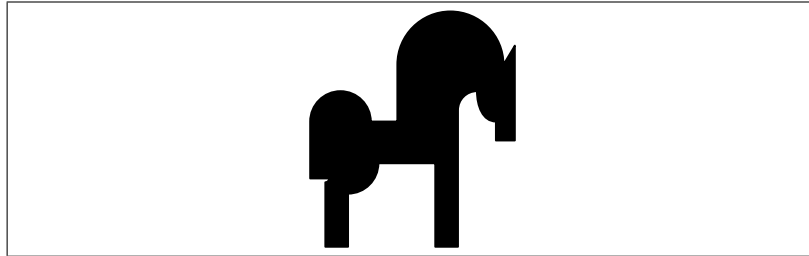


Figure 48

8.1.4 Caption Beside with Centered Object

The figure 49 uses tricky float style:

```
<preamble>
\DeclareFloatSeparators{mcapwidth}{\hskip-\FCwidth}
<preamble>
\floatsetup[figure]
{style=plain,capposition=beside,objectset=centering,
floatwidth=\columnwidth,capbesidewidth=6cc,
capbesideposition=left,capbesidessep=mcapwidth,
margins=centering,capbesideframe=yes,
floatwidth=sidefil}
```

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.



Figure 49
Plain figure

There were also temporarily, in group, changed settings for caption in figure 49:

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

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

The float row with predefined width of beside object—caption boxes (figures 50 and 51): just define before `\fcapside` command something like

```
\setlength\hsize{1.2\hsize}%
```

as was defined for first figure, or

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

before second (here is clear that you can set `\setlength\hsize{.8\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.

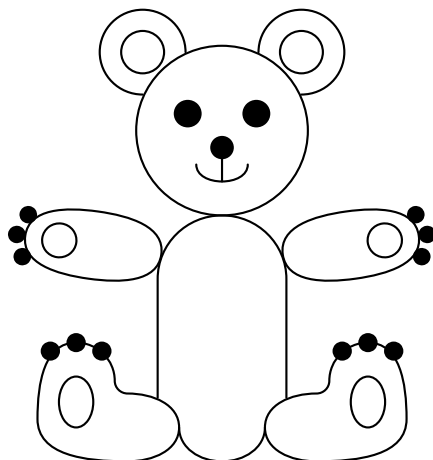


Figure 50

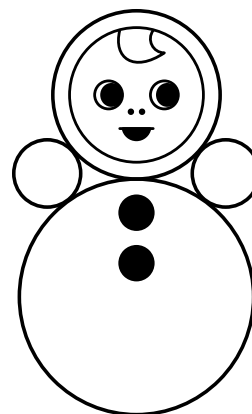


Figure 51

There was used following setup for figures:

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

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

The float row with object and beside caption combined with object and caption below (figures 52 and 53). 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][3.6cm]
...

\ffigbox[\Xhsize][3.6cm]
...
\end{floatrow}%
\end{figure}

```

Figure 52. Float box (`\fcapside`) with beside caption in float row width float with caption below

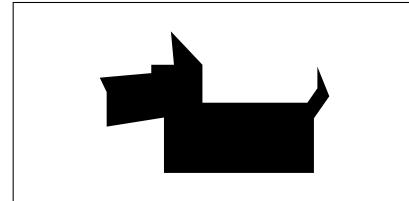
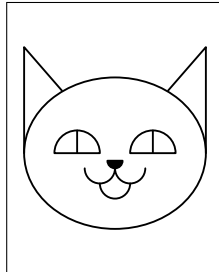


Figure 53. Float box (`\ffigbox`) width of rest float row space

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

```

\begin{figure}\CenterFloatBoxes
\begin{floatrow}
\ffigbox[1.2\FBwidth][3.6cm]
...

\hsize\Xhsize
\fcapside[\FBwidth][3.6cm]
...
\end{floatrow}%
\end{figure}

```

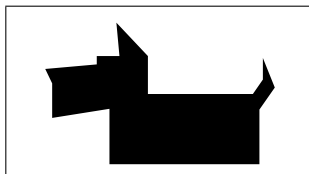


Figure 54. Float box (`\ffigbox`) in mirror float row

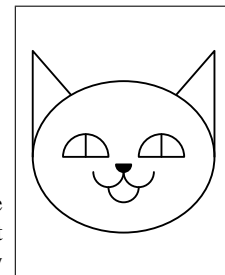


Figure 55. Float box with beside caption (`\fcapside`) in mirror float row width float with caption below

The figures 54–55 use following float style settings:

```

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

```

8.1.7 Photo-Album-like Layouts

Another example of miscellaneous float row (figures 56–58, and, “mirror layout”—59–61) were created by following lines:

```
\begin{figure}\BottomFloatBoxes
\begin{floatrow}
\hsize1.2\hsize \ffigbox[][6.7cm]
...

\vbox to6.7cm
{\floatsetup[figure]{floatrowsep=none}\killfloatstyle
\ffigbox[.8\hsize]
...
\vss
\ffigbox[.8\hsize]
...%
}%
\end{floatrow}%
\end{figure}
```

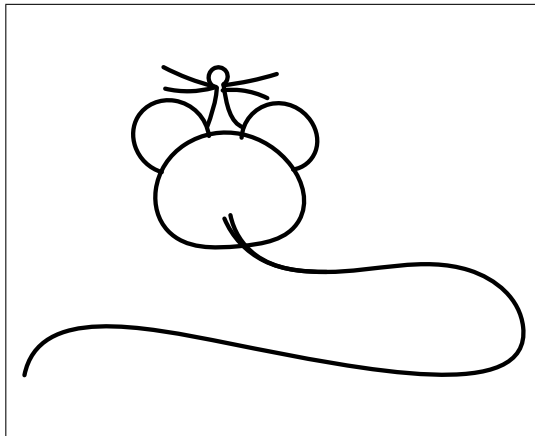


Figure 56. Float box in photo-album-like layout: alone in left column



Figure 57. Float box in photo-album-like layout: upper float in right column

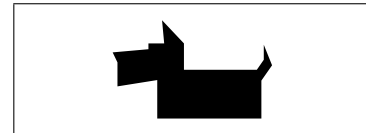


Figure 58. Lower float in right column

The “mirror” layout created by following commands:

```
\begin{figure}[t]\TopFloatBoxes
\begin{floatrow}
\vtop to7cm
{\floatsetup[figure]{floatrowsep=none}\killfloatstyle
\ffigbox[.8\hsize]
...
\vss
\ffigbox[.8\hsize]
```



Figure 59. Float box in photo-album-like layout: upper float in left column

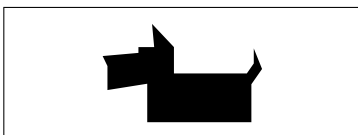


Figure 60. Float box in photo-album-like layout: lower float in the left column

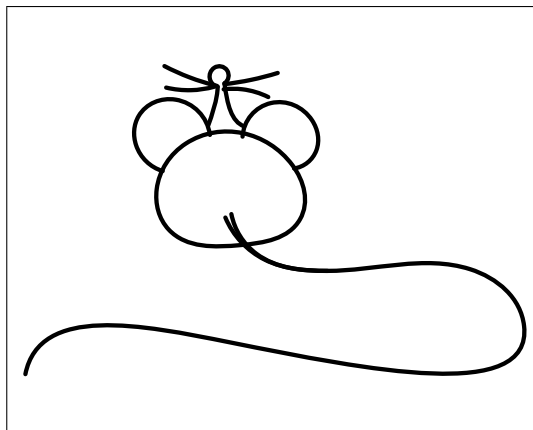


Figure 61. Float box in photo-album-like layout: alone in right column

```
...%
\vskip0pt}\floatrowsep

\ffigbox[\Xhsize][7cm-11pt]
...
\end{floatrow}%
\end{figure}
```

Note that in second example with “mirror” layout there was used trick with *height* definition—caption of float in the left column is one line longer, so for the right column height of float was reduced by 11pt—`\baselineskip` for `\small` size¹ (here were used `calc` package possibilities). The `\vtop` of left column ends with `\vskip0pt`, otherwise you get fanny unwanted layout.

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

Note that these examples are rather specific—you may try with other combinations (e.g. more-“columned”), but maybe these layouts need more care with usage of `\Xhsize` and/or `\floatrowsep`.

I suppose that last two examples could conflict with “motto” of this package—to reduce and remove layout code from document; but photo-album-like layout is rather rare in technical literature (It isn’t?).

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

8.2 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.

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 sample file `sample-longtable.tex` which uses *beta-temp* package-patch `fr-longtable` with defined commands `\endlasthead` and `\endprelastfoot` which defines captions for continued and last pages of long table in three possible ways.

8.3 Obsolete Commands

8.3.1 The User Interface—New Floats [float]

`\newfloat` The most important command in float is the `\newfloat` command¹. It is patterned on `\newtheorem`. The `\newfloat` command takes three required and one optional argument; it is of the form

```
\newfloat{<type>}{<placement>}{<ext>}[<within>]
```

- `{<type>}` is the ‘type’ of the new class of floats, like `program` or `algorithm`. After the appropriate `\newfloat`, commands like `\begin{program}` or `\end{algorithm*}` will be available.
- `{<placement>}` gives the default placement parameters for this class of floats. The placement parameters are the same as in standard L^AT_EX, i.e., t, b, p and h for ‘top’, ‘bottom’, ‘page’ and ‘here’, respectively.
- `{<ext>}` 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 `{<ext>}` as a file name.
- `[<within>]` Finally, the optional argument `<within>` determines whether floats of this class will be numbered within some sectional unit of the document. For example, if `[<within>] = chapter`, the floats will be numbered within chapters. (In standard L^AT_EX, this happens with figures and tables in the report and book document styles.) As an example, Program 8.1 was created by a command sequence similar to that shown in the following Example².

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 8.1. This is another silly floating Example. Except that this one doesn’t actually float because it uses the [H] optional parameter to appear **Here**. (Gotcha.)

`\floatstyle` The `\floatstyle` command sets a default float style. This float style will be used for all the floats that are subsequently defined using `\newfloat`, until another `\floatstyle` command appears. The `\floatstyle` command takes one argument, the name of a float style. For instance, `\floatstyle{ruled}`. Specifying a string that does not name a valid float style is an error.

`\floatname` The `\floatname` command lets you define the *float name* that L^AT_EX uses in the caption of a float, i.e., ‘Figure’ for a figure and so on. For example, `\floatname{program}{Program}`. The `\newfloat` command sets the float name to its argument `<type>` if no other name has been specified before.

Program 8.1 The first program. This hasn't got anything to do with the package but is included as an example. Note the ruled float style.

```
#include <stdio.h>

int main(int argc, char **argv) {
    int i;
    for (i = 0; i < argc; ++i)
        printf("argv[%d] = %s\n", i, argv[i]);
    return 0;
}
```

`\floatplacement` The `\floatplacement` command resets the default placement specifier of a class of floats. E.g., `\floatplacement{figure}{tp}`.

`\restylefloat` The `\restylefloat` command is necessary to change styles for the standard float types `figure` and `table`. Since these aren't usually defined via `\newfloat`, they don't have a style associated with them. Thus you have to say, for example,

```
\floatstyle{ruled}
\restylefloat{table}
```

to have tables come out ruled. The command also lets you change style for floats that you define via `\newfloat`, although this is, typographically speaking, not a good idea. See table 18 for an 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.

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$
0	1							
1	1	1						
2	1	2	1					
3	1	3	3	1				
4	1	4	6	4	1			
5	1	5	10	10	5	1		
6	1	6	15	20	15	6	1	
7	1	7	21	35	35	21	7	1

Table 18: Pascal's triangle. This is a re-styled L^AT_EX table.

¹It doubles the `\DeclareNewFloatType` command.

²Settings for Example float environment were created by `\DeclareNewFloatType` macro stuff.

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.

8.3.2 The `\floatsetup` keys, 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>rowpercode</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>floatmarginsset=</code>	<code>margins=</code>

Renamed keys	
Key	Changed to
<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>