The **nccfloats** package[*]

Alexander I. Rozhenko
rozhenko@oapmg.sscc.ru

2006/01/07

The standard LaTeX floating environments, namely `figure` and `table`, allow user to place floating material in a document. But they do not introduce a style in which this material must be formatted. In this package, styles are joined with floats and mini-floats are introduced. Mini-floats are prepared at a mini-page and allow captions within. Basing on mini-floats, a number of service commands for figures and tables are defined. The ability to create other types of mini-floats is introduced with the `\newminifloat` command.

Contents

1 Basic Commands

`\FloatStyle` The `\FloatStyle[`⟨*type*⟩`]{`⟨*style*⟩`}` command sets a style for the float of the given ⟨*type*⟩. If the ⟨*type*⟩ is omitted, the default style is specified. It will be applied to

---

[*] This file has version number v1.2, last revised 2006/01/07.

a float or mini-float if no specialized style was defined. This command is available in the preamble only.

To specify the style of a mini-float and of service commands based on it, add the '`mini`' prefix to the float type. The default styles are

```
\FloatStyle{}
\FloatStyle[minifigure]{\footnotesize\centering}
\FloatStyle[minitable]{\footnotesize\centering}
```

This means that the formatting of floats prepared with standard `figure` and `table` environments does not changed, but service commands based on mini-figures and mini-tables use a special formatting with `\footnotesize` font and the centered alignment.

`\normalfloatstyle` This command is applied inside floats or mini-floats to reset formatting style of subsequent floating material to the standard formatting with paragraph alignment and the normal font of normal size.

`\minifig` `\minitabl` We start with the basic commands, namely `\minifig` and `\minitabl`. They prepare a material in a mini-page and allow using the `\caption` command in the body. Their syntax is similar to the `\parbox` command:

```
\minifig [⟨pos⟩][⟨height⟩][⟨inner-pos⟩]{⟨width⟩}{⟨body⟩}
\minitabl[⟨pos⟩][⟨height⟩][⟨inner-pos⟩]{⟨width⟩}{⟨body⟩}
```

The ⟨*pos*⟩ is a vertical alignment parameter for mini-page (`t`, `b`, or `c`) with respect to surrounding text; the ⟨*height*⟩ is a mini-page height required; the ⟨*inner-pos*⟩ is a vertical alignment of text inside the mini-page (`t`, `b`, `c`, or `s`); and the ⟨*width*⟩ is the mini-page width. The ⟨*body*⟩ is prepared in the style specified by the `\FloatingStyle` command and can contain the `\caption` command inside.

All other floating extension commands are based on these two commands.

2 Side Figures and Tables

For small figures and tables, it is preferable to insert them inside a text instead of using floating mechanism. The typographic rules usually require an illustrative material to occupy an outer side of page. In two-side mode, this means figure and tables should be on the right side if a page number is odd and on the left side if page number is even. In one-side mode, figures and tables must occupy the right side of page.

`\sidefig` `\sidetabl` The following commands support such a placement:

```
\sidefig[⟨pos⟩](w_1)(w_2){⟨figure⟩}{⟨text⟩}
\sidefig*[⟨pos⟩](w_1)(w_2){⟨figure⟩}{⟨text⟩}
\sidetabl[⟨pos⟩](w_1)(w_2){⟨table⟩}{⟨text⟩}
\sidetabl*[⟨pos⟩](w_1)(w_2){⟨table⟩}{⟨text⟩}
```

We use the term *mini-float* for the small illustrating material (figure or table), however taking into account that it is not a float at all. It is inserted in the main flow next to a paragraph box specified in the last parameter of above described commands.

The no-star forms of above described commands place a mini-float next to the specified text on the outer side of page (to the right for odd page and to the left for even page). In two-column or one-side mode, mini-float is always posed to the right. The star-forms provide the reverse placement. By default, mini-float is vertically centered with respect to the text and the `\strut` command is inserted at the beginning and at the end of the ⟨*text*⟩ to provide normal baseline distances of the first and last lines of the text from surrounding text lines.

All parameters in square and round brackets are optional and mean the following:

⟨*pos*⟩	specifies mini-float alignment (`t`, `b`, or `c`; default is `c`) with respect to text box and can contain additional chars controlling the text body preparation: `j` means the last line of the text to be justified to the right and `n` means suppressing of struts insertion (they should be inserted manually if necessary);
w_1	is the width of mini-float; and
w_2	is the width of the text box.

You can omit units in the width parameters. In this case, the width value is considered as a multiple of `\unitlength` (similarly to the use of length dimensions in the picture environment).

If both width parameters are absent, the width of both mini-float and text body is calculated as `(\linewidth-1.5em)/2`. If w_2 is absent, the text body width is calculated as `\linewidth-`w_1`-1.5em`.

The placement of side-floats in the document consists in the following steps:

1. Decide where you want to insert a side-float;
2. Insert a `\sidefig` or `\sidetabl` command after a word that finishes the line before the supposed side-float position;
3. Specify a width of float in its parameter and set the top alignment as the ⟨*pos*⟩ parameter (e.g. `\sidefig[t](`w_1`)`);
4. Prepare the side-float in the first mandatory parameter of the command (e.g. `\sidefig[t](`w_1`){`⟨*figure*⟩`}`);
5. Enclose enough text going after the command in braces;
6. Translate the document;
7. Find what part of the text is redundant in the ⟨*text*⟩ parameter;
8. Move it after the close brace;

9. If the same paragraph continues after the close brace, add the `j` letter to the ⟨*pos*⟩ parameter. Also change the `t` alignment to `c` alignment in the ⟨*pos*⟩ parameter;
10. Translate the document once more;
11. If the side-float has a wrong placement (this can appear when paragraph with a side-float begins at the end of page), insert the star after the side-float command and translate the document once more.

`\ifleftsidefloat`

While preparing a side-float, it is sometimes necessary to provide conditional placement depending on the side a mini-float is posed. The command

```
\ifleftsidefloat{⟨left-clause⟩}{⟨right-clause⟩}
```

provides this. It is useful in parameters of `\sidefig` or `\sidetabl` and processes ⟨*left-clause*⟩ if the mini-float is posed to the left and ⟨*right-clause*⟩ otherwise.

Side-floats can be also used within floating environments to pos a caption near a figure or table.

3 Floating Figures and Tables

`\fig`
`\tabl`

The following commands envelop floating environments:

```
\fig[⟨placement⟩](w){⟨body⟩}
\fig*[⟨placement⟩](w){⟨body⟩}
\tabl[⟨placement⟩](w){⟨body⟩}
\tabl*[⟨placement⟩](w){⟨body⟩}
```

The ⟨*placement*⟩ is a float placement parameter describing places where a float can appear. The default value is `ht` (here or at the top of page). The optional w parameter defines a width of box occupied by the float (the width of nested `\minifig` or `\minitabl`). If it is omitted, the float has the maximum width equal to the `\linewidth`.

The `\fig` and `\tabl` commands envelop the `figure` and `table` environments respectively. Their star-forms envelop `figure*` or `table*` environments respectively.

4 Two Floating Figures or Tables Side by Side

`\figs`
`\tabls`

The following commands place two figures or tables side by side.

```
\figs[⟨placement⟩](w_1)(w_2){⟨body1⟩}{⟨body2⟩}
\figs*[⟨placement⟩](w_1)(w_2){⟨body1⟩}{⟨body2⟩}
\tabls[⟨placement⟩](w_1)(w_2){⟨body1⟩}{⟨body2⟩}
\tabls*[⟨placement⟩](w_1)(w_2){⟨body1⟩}{⟨body2⟩}
```

The ⟨*body1*⟩ is a body of the left figure or table and the ⟨*body2*⟩ is a body of the right figure or table. Other parameters are optional. The meaning and default value of the ⟨*placement*⟩ parameter is the same as described above. The w_1 and w_2 parameters are widths of left and right boxes. If they both are omitted, the left and right boxes will have the width equal to `(\linewidth-1em)/2`. If w_2 is omitted, the right box will occupy the rest of horizontal space minus `1em`. If both parameters are specified, the rest space is inserted between boxes. If the total width of left and right floats exceeds the `\linewidth`, the floats will overlap at the middle (a negative horizontal space is inserted between them).

In the `\tabls` command, boxes of the left and right bodies are top-aligned, but, in the `\figs` command, the bottom alignment is used. The star-forms of this commands are based on the `figure*` or `table*` environments respectively.

5 Declare a Mini-float and Service Commands

`\newminifloat`

If a new type of float is introduced, the respective mini-float and service commands can be helpful for it. To prepare them, use the following declaration:

`\newminifloat{`⟨*gen*⟩`}{`⟨*type*⟩`}{`⟨*placement*⟩`}{`⟨*pos*⟩`}`

Here ⟨*gen*⟩ is a root for command names to be generated, ⟨*type*⟩ is a float type, ⟨*placement*⟩ is a default placement on the page, and ⟨*pos*⟩ is a vertical alignment for pair of floats.

This command declares 4 commands: `\mini`⟨*gen*⟩, `\side`⟨*gen*⟩, `\`⟨*gen*⟩, and `\`⟨*gen*⟩`s`. For example, the commands described in previous sections are declared as follows:

```
\newminifloat{fig}{figure}{ht}{b}
\newminifloat{tabl}{table}{ht}{t}
```

6 The Implementation

The package uses some commands of the `nccboxes` package. Load it here:

```
⟨*package⟩
\RequirePackage{nccboxes}[2002/03/20]
```

6.1 Float Style

`\FloatStyle`

`\FloatStyle[`⟨*type*⟩`]{`⟨*style*⟩`}` specifies a style for a given float type.

```
\newcommand*{\FloatStyle}[2][]{%
  \expandafter\def\csname NCC@fltstyle@#1\endcsname{#2}}
\@onlypreamble\FloatStyle
```

`\NCC@setfltstyle`

`\NCC@setfltstyle{`⟨*prefix*⟩`}` applies a style for a float of `\@captype` type. While selection a style to be applied it adds the given ⟨*prefix*⟩ to the float type.

```
\def\NCC@setfltstyle#1{%
```

```
  \edef\@tempa{NCC@fltstyle@#1\@captype}%
  \@ifundefined{\@tempa}{\NCC@fltstyle@}{\csname\@tempa\endcsname}%
}
```

We add this style with empty prefix to the `\@floatboxreset` hook which is applied at the end of preamble of a float.

```
\g@addto@macro\@floatboxreset{\NCC@setfltstyle{}}
```

`\normalfloatstyle` Reset a float style to par-box formatting with normal font of the normal size.

```
\newcommand\normalfloatstyle{%
  \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
  \parfillskip\@flushglue \let\\\@normalcr
  \reset@font \normalsize
}
```

6.2 The Kernel

`\NCC@minifloat` The base for mini-floats

`\NCC@minifloat[`⟨*pos*⟩`][`⟨*height*⟩`][`⟨*inner-pos*⟩`]{`⟨*width*⟩`}{`⟨*body*⟩`}`

It finishes a mini-float with extra `\endgroup` command. A `\@captype` should be specified before it.

```
\newcommand*\NCC@minifloat[1][c]{%
  \@ifnextchar[{\NCC@mflt{#1}}{\NCC@@mflt{#1}\relax[s]}}
\def\NCC@mflt#1[#2]{%
  \@ifnextchar[{\NCC@@mflt{#1}{#2}}{\NCC@mflt{#1}{#2}[#1]}}
\long\def\NCC@@mflt#1#2[#3]#4#5{%
  \@iiiminipage{#1}{#2}[#3]{#4}\normalfloatstyle
  \NCC@setfltstyle{mini}#5\endminipage\endgroup
}
```

`\NCC@pair` The command

`\NCC@pair{`⟨*c1*⟩`}{`⟨*c2*⟩`}{`⟨*def-dist*⟩`}{`⟨*def-place*⟩`}*[`⟨*place*⟩`](`w_1`)(`w_2`)`

executes ⟨*c1*⟩`{`⟨*place*⟩`}{`w_1`}{`w_2`}` if star is absent or ⟨*c2*⟩`{`⟨*place*⟩`}{`w_1`}{`w_2`}` if star presents. Four first parameters are mandatory. Others a optional. The ⟨*def-dist*⟩ parameter contains a default distance value. It is saved in the `\@tempdimc` register. The ⟨*def-place*⟩ parameter contains the default value for the ⟨*place*⟩ parameter. If the last one is omitted, the ⟨*def-place*⟩ is used instead.

```
\def\NCC@pair#1#2#3#4{\setlength\@tempdimc{#3}%
  \@ifstar{\NCC@pair@{#2}{#4}}{\NCC@pair@{#1}{#4}}}
\def\NCC@pair@#1#2{\@ifnextchar[{\NCC@pair@@{#1}}{\NCC@pair@@{#1}[#2]}}
\def\NCC@pair@@#1[#2]{\def\@tempa{#1{#2}}%
  \@ifnextchar({\NCC@pair@@@}{\NCC@@pair()()}}
\def\NCC@pair@@@(#1){\@ifnextchar({\NCC@@pair(#1)}{\NCC@@pair(#1)()}}
\def\NCC@@pair(#1)(#2){\@tempa{#1}{#2}}
```

`\NCC@setwidth` The `\NCC@setwidth{`⟨*register*⟩`}{`⟨*width*⟩`}` command sets the given ⟨*width*⟩ for the ⟨*register*⟩. If units in ⟨*width*⟩ are omitted, the `\unitlength` unit is used. In other words, if ⟨*width*⟩ is a real number, it is considered as a multiple of `\unitlength`.

```
\def\NCC@setwidth#1#2{\@defaultunits#1#2\unitlength\relax\@nnil}
```

`\NCC@wcalc` The `\NCC@wcalc{`w_1`}{`w_2`}` calculates widths of left and right boxes in the `\@tempdima` and `\@tempdimb` registers. The distance between boxes must be specified in `\@tempdimc` register before the call. The algorithm:

- If w_1 is empty, `\@tempdima:=(\linewidth-\@tempdimc)/2`, otherwise, `\@tempdima:=`w_1;
- If w_2 is empty, `\@tempdimb:=\linewidth-\@tempdima-\@tempdimc`, otherwise, `\@tempdimb:=`w_2;
- If w_2 is nonempty, `\@tempdimc:=\linewidth-\@tempdima-\@tempdimb`.

```
\def\NCC@wcalc#1#2{%
  \if!#1!\@tempdima .5\linewidth \advance\@tempdima -.5\@tempdimc
  \else  \NCC@setwidth\@tempdima{#1}%
  \fi
  \if!#2!\@tempdimb \linewidth \advance\@tempdimb -\@tempdima
         \advance\@tempdimb -\@tempdimc
  \else  \NCC@setwidth\@tempdimb{#2}%
         \@tempdimc \linewidth \advance\@tempdimc -\@tempdima
         \advance\@tempdimc -\@tempdimb
  \fi
}
```

6.3 Side Floats

`\ifleftsidefloat` This command is used in parameters of side-floats.

```
\newif\ifNCC@smfltleft
\newcommand{\ifleftsidefloat}{%
  \ifNCC@smfltleft
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
```

`\NCC@sidemfloat` The command

`\NCC@sidemfloat{`⟨*command*⟩`}*[`⟨*pos*⟩`](`w_1`)(`w_2`){`⟨*mini-float*⟩`}{`⟨*text*⟩`}`

is used for preparing a side-float. The ⟨*command*⟩ parameter contains a `\mini`⟨*gen*⟩ command. The ⟨*pos*⟩ parameter specifies vertical alignment and additional flags. The w_1 and w_2 parameters (if present) specify widthes of ⟨*mini-float*⟩ and ⟨*text*⟩ boxes. Starred version reverses the position of side-float and text boxes.

The implementation of these commands is based on the `\NCC@pair` command that parses all optional parameters. Finally the `\NCC@smflt` command is executed.

```
\def\NCC@sidemfloat#1{%
  \NCC@smfltleftfalse
  \if@twocolumn \else
    \if@twoside
      \ifodd\c@page \else \NCC@smfltlefttrue \fi
    \fi
  \fi
  \NCC@pair{\NCC@smflt{#1}}%
    {\ifNCC@smfltleft \NCC@smfltleftfalse \else \NCC@smfltlefttrue\fi
     \NCC@smflt{#1}}%
    {1.5em}{}%
}
```

`\NCC@smflt` The command

`\NCC@smflt{`⟨*command*⟩`}{`⟨*pos*⟩`}{`w_1`}{`w_2`}{`⟨*mini-float*⟩`}{`⟨*text*⟩`}`

prepares a side-float. The `\@tempdimc` register contains the default distance between the mini-float and text.

```
\long\def\NCC@smflt#1#2#3#4#5#6{%
```

Parse the ⟨*pos*⟩ parameter. Create a `\NCC@`⟨*letter*⟩ command with empty content for every ⟨*letter*⟩ from the ⟨*pos*⟩.

```
  \let\NCC@t\relax \let\NCC@b\relax \let\NCC@j\relax \let\NCC@n\strut
  \@tfor\@tempa :=#2\do {%
    \expandafter\let\csname NCC@\@tempa\endcsname\@empty}%
```

Define the vertical alignment letter in the `\NCC@c` command.

```
  \ifx\NCC@t\@empty \def\NCC@c{t}\else
   \ifx\NCC@b\@empty \def\NCC@c{b}\else
    \def\NCC@c{c}%
   \fi
  \fi
```

Define a justification hook in the `\NCC@j` command.

```
  \ifx\NCC@j\@empty \def\NCC@j{\parfillskip\z@skip}\fi
```

Define the text starting hook in the `\NCC@t` command. It will contain the `\parindent` setting command and the optional `\noindent` command.

```
  \edef\NCC@t{\parindent\the\parindent\ifvmode\else\noindent\fi}%
```

Complete the current paragraph and leave the horizontal mode.

```
  \ifvmode\else
    \unskip{\parfillskip\rightskip\par}\vskip -\parskip
  \fi
```

Prepare the side-float in `\@tempboxa`:

```
  \setbox\@tempboxa\vbox{\hsize\linewidth\noindent
```

Calculate widthes of left and right boxes and distance between them in `\@tempdima`, `\@tempdimb`, and `\@tempdimc`.

```
    \NCC@wcalc{#3}{#4}%
```

Conditionally put a side-float to the left:

```
    \ifNCC@smfltleft
      \jparbox{\Strut}[\NCC@c]\@tempdima{#1\@tempdima{#5}}%
      \nobreak\hskip\@tempdimc
    \fi
```

Put a text box:

```
    \jparbox{\NCC@n\Strut}[\NCC@c]\@tempdimb{%
      \everypar{\NCC@n\everypar{}}\NCC@t#6%
      \ifvmode \else \unskip\NCC@n\NCC@j\fi}%
```

Conditionally put a side-float to the right:

```
    \ifNCC@smfltleft \else
      \nobreak\hskip\@tempdimc
      \jparbox{\Strut}[\NCC@c]\@tempdima{#1\@tempdima{#5}}%
    \fi
  }%
```

Games with height and depth the `\@temboxa` allow us produce right line spacing with surrounding text.

```
  \@tempdima\dp\@tempboxa \advance\@tempdima\lineskip
  \dp\@tempboxa\@tempdima
  \@tempdima\ht\@tempboxa \advance\@tempdima -\ht\strutbox
  \noindent \raise-\@tempdima\box\@tempboxa
}
```

6.4 Service Commands

`\NCC@float` The command

`\NCC@float{`⟨*type*⟩`}{`⟨*def-place*⟩`}*[`⟨*placement*⟩`](`w`){`⟨*body*⟩`}`

is the envelope for a mini-float inside a float. The ⟨*def-place*⟩ is the default placement specifier.

```
\def\NCC@float#1#2{\@ifstar{\NCC@flt{#1*}{#2}}{\NCC@flt{#1}{#2}}}
\def\NCC@flt#1#2{\@ifnextchar[{\NCC@flt@{#1}}{\NCC@flt@{#1}[#2]}}
\def\NCC@flt@#1[#2]{\begin{#1}[#2]\normalfloatstyle\centering
 \@ifnextchar({\NCC@@flt{#1}}{\NCC@@flt{#1}()}}
\long\def\NCC@@flt#1(#2)#3{%
  \if!#2!\@tempdima\linewidth \else \NCC@setwidth\@tempdima{#2}\fi
  \begingroup\NCC@minifloat[c]\@tempdima{#3}%
  \end{#1}%
}
```

`\NCC@floats` The command

`\NCC@floats{`⟨*type*⟩`}{`⟨*pos*⟩`}{`⟨*def-place*⟩`}*[`⟨*placement*⟩`](`w_1`)(`w_2`)`
`{`⟨*body1*⟩`}{`⟨*body2*⟩`}`

is the envelope for a pair of mini-floats inside a float. The implementation of these commands is based on the `\NCC@pair` command that parses all optional parameters. Finally the `\NCC@flts` command is executed.

```
\def\NCC@floats#1#2#3{%
  \NCC@pair{\NCC@@flts{#1}{#2}}{\NCC@@flts{#1*}{#2}}{1em}{#3}}
```

`\NCC@flts` The command

`\NCC@flts{`⟨*type*⟩`}{`⟨*pos*⟩`}{`⟨*placement*⟩`}{`w_1`}{`w_2`}{`⟨*body1*⟩`}{`⟨*body2*⟩`}`

prepares a pair of floats within ⟨*type*⟩ environment. The ⟨*pos*⟩ contains relative vertical alignment of floats. The w_1 and w_2 parameters (if present) specify widthes of floats. The `\@tempdimc` register contains the default distance between floats.

```
\long\def\NCC@@flts#1#2#3#4#5#6#7{%
  \begin{#1}[#3]\normalfloatstyle\NCC@wcalc{#4}{#5}%
    \begingroup\NCC@minifloat[#2]\@tempdima{#6}%
    \nobreak\hskip\@tempdimc
    \begingroup\NCC@minifloat[#2]\@tempdimb{#7}%
  \end{#1}%
}
```

6.5 Declare a New Mini-float and Service Commands

`\newminifloat` The command

`\newminifloat{`⟨*gen*⟩`}{`⟨*type*⟩`}{`⟨*def-place*⟩`}{`⟨*pos*⟩`}`

declares a new mini-float and 3 service commands.

```
\newcommand*\newminifloat[4]{%
  \edef\@tempa{%
```

Prepare `\mini`⟨*gen*⟩ definition:

```
    \noexpand\newcommand\expandafter\noexpand\csname mini#1\endcsname{%
      \noexpand\begingroup\noexpand\def\noexpand\@captype{#2}%
      \noexpand\NCC@minifloat}%
```

Prepare `\side`⟨*gen*⟩ definition:

```
    \noexpand\newcommand\expandafter\noexpand\csname side#1\endcsname{%
      \noexpand\NCC@sidemfloat{%
        \expandafter\noexpand\csname mini#1\endcsname}}%
```

Prepare \⟨*gen*⟩ definition:

```
    \noexpand\newcommand\expandafter\noexpand\csname #1\endcsname{%
      \noexpand\NCC@float{#2}{#3}}%
```

Prepare \⟨*gen*⟩`s` definition:

```
    \noexpand\newcommand\expandafter\noexpand\csname #1s\endcsname{%
      \noexpand\NCC@floats{#2}{#4}{#3}}%
  }%
```

Define all commands:

```
  \@tempa
}
\@onlypreamble\newminifloat
```

6.6 Base Mini-floats and Defaults

```
\newminifloat{fig}{figure}{ht}{b}
\newminifloat{tabl}{table}{ht}{t}
\FloatStyle{}
\FloatStyle[minifigure]{\footnotesize\centering}
\FloatStyle[minitable]{\footnotesize\centering}
⟨/package⟩
```