The minidocument package

Maurice Leclaire
leclaire@in.tum.de

v1.0 – 2017/04/11

Abstract

The minidocument package is used to create miniature documents inside other documents as can be seen in Figure 1.

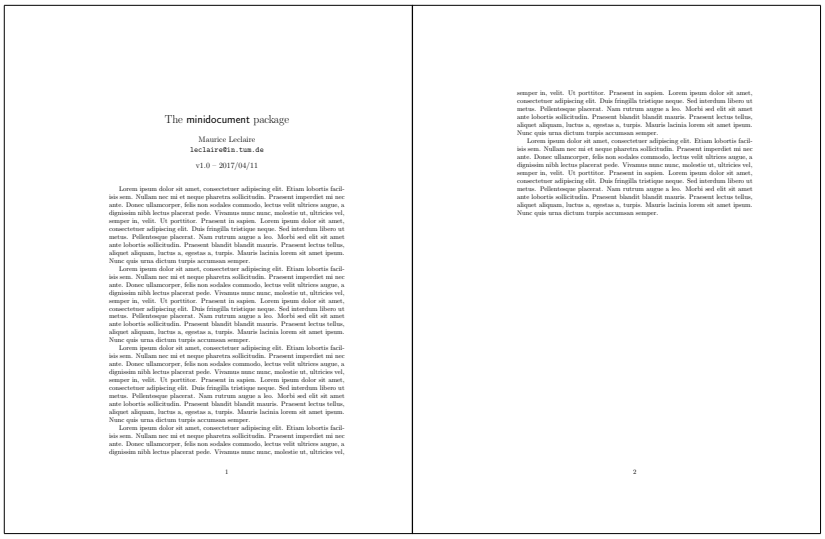

Figure 1: A minidocument example

Contents

1 Usage

`minidocument` A minidocument is created with the `minidocument` environment. The `page` counter is set to one at the beginning of a minidocument and restored afterwards. It is important, that this environment is placed in outer vertical mode. Otherwise, it will not work properly. However, it is of course possible to place the resulting minidocument everywhere. Therefore, the environment itself does not output anything. The example minidocument in Figure 1 is defined as:

```
\begin{minidocument}
    \maketitle
    \Blindtext
\end{minidocument}
```

`\lastminidocument` To output the minidocument, the command `\lastminidocument` is used, e. g.:

```
\begin{figure}
    \lastminidocument
\end{figure}
```

`\minidocumentscale` By default, the minidocument is scaled down with `\minidocumentscale`, which defaults to 0.333. This can be changed with e. g.

```
\def\minidocumentscale{0.25}
```

as it is done in this documentation.

`\minidocumentshipout` The macro `\minidocumentshipout` is called for every page in the minidocument. It takes a single argument containing the page and is responsible for postprocessing it. It can be redefined to obtain a different look. The default is defined as

```
\scalebox{\minidocumentscale}{\frame{#1}}
```

2 Implementation

The `graphics` package is required for `\scalebox`.

```
\RequirePackage{graphics}
```

`\minidocument@box` We allocate a box register for temporary storage.

```
\newbox\minidocument@box
```

`\minidocument@aggrbox` The following box will aggregate the minidocument.

```
\newbox\minidocument@aggrbox
```

`\minidocument@hold` This box is used to store the current vertical list when entering the minidocument.

```
\newbox\minidocument@hold
```

`\minidocument@orig@output` We need a token register to store the output routine.

```
\newtoks\minidocument@orig@output
```

The following method of overloading the shipout procedure is borrowed from the atbegshi package[1] where it is explained in detail. The problem we face here is that `\shipout` can be called with many different box specifications which can not be catched as macro arguments. Therefore, we use `\setbox` to get the box specification and `\afterassignment` to execute the code that handles the box. However, for direct box specifications the code specified via `\afterassignment` is not executed after the box specification is finished but directly at the beginning of the box. Therefore, we need an additional `\aftergroup` in the case of a direct box. We only have to determine if we are at the beginning of a direct box. We can do this by inspecting `\lastkern` as it is zero at the beginning of a box.

`\minidocument@shipout` This macro is the entry point of our shipout procedure. Inside a new group we assign and start a new box with a specified kerning. This box is only temporary and gives us a reliable environment. We install the `\afterassignment` hook and assign the passed box to a box register. This assignment is global as it must survive the group while the temporary box which is assigned to the same box register is lost.

```
\def\minidocument@shipout{%
    \begingroup\setbox\minidocument@box\hbox\bgroup\kern\p@%
    \afterassignment\minidocument@testshipout%
    \global\setbox\minidocument@box%
}
```

`\minidocument@testshipout` The following macro is called by `\afterassignment`. It checks via `\lastkern` if it is called at the beginning of a box. In this case an `\aftergroup` is inserted. Either way, when the box is finished the next macro is called.

```
\def\minidocument@testshipout{%
    \ifdim\lastkern=\z@%
        \expandafter\aftergroup%
    \fi%
    \minidocument@makeshipout%
}
```

`\minidocument@makeshipout` This macro ends the temporary box and the group. We now have the passed box in the box register. If the box register is not void, we do the virtual shipout. Therefore, we position the box at the right place of a page. Afterwards, we call `\minidocumentshipout` to postprocess the page. We then add the page to the aggregation box.

```
\def\minidocument@makeshipout{%
    \egroup\endgroup%
    \ifvoid\minidocument@box\else%
        \global\setbox\minidocument@box\vbox to \paperheight {
            \vskip1in%
            \hbox to \paperwidth {
                \hskip1in%
                \box\minidocument@box%
                \hfill
            }%
```

```
            \vfill
        }%
        \global\setbox\minidocument@aggrbox\hbox{\unhbox\minidocument@aggrbox%
            \minidocumentshipout{\box\minidocument@box}%
            \hskip0pt%
        }%
    \fi%
}
```

`minidocument` The `minidocument` environment first saves the current vertical list. Therefore, the output routine is changed and a page break is enforced. Then, the shipout procedure is switched and the page counter is set to one. At the end of the environment, the shipout procedure and the page number are restored. Afterwards, the saved vertical list is reinserted.

```
\newenvironment{minidocument}{%
    \minidocument@orig@output\output%
    \output{\global\setbox\minidocument@hold\box\@cclv}%
    \penalty-\@Mi%
    \output\minidocument@orig@output%
    \let\minidocument@orig@shipout\shipout%
    \let\shipout\minidocument@shipout%
    \edef\minidocument@page{\the\c@page}%
    \c@page\@ne%
}{%
    \clearpage%
    \let\shipout\minidocument@orig@shipout%
    \global\c@page\minidocument@page%
    \unvbox\minidocument@hold%
}
```

`\lastminidocument` The macro `\lastminidocument` unboxes the aggregated minidocument so that it can flow into the main document.

```
\def\lastminidocument{%
    \unhbox\minidocument@aggrbox\unskip%
}
```

`\minidocumentscale` We set the default scale factor to 1/3, so that two pages can fit next to each other on a page with appropriate margins.

```
\def\minidocumentscale{0.333}
```

`\minidocumentshipout` The default `\minidocumentshipout` draws a frame around the page and scales the page with `\minidocumentscale`.

```
\newcommand{\minidocumentshipout}[1]{%
    \scalebox{\minidocumentscale}{\frame{#1}}%
}
```

References

[1] Heiko Oberdiek. The atbegshi package. CTAN:
http://ctan.org/pkg/atbegshi

Change History

v1.0

General: Initial version 1

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

R

S

U