The semioneside package*

Stephan Hennig
stephanhennig@arcor.de

September 20, 2005

Abstract

This package supports the creation of semi one sided documents. That is, two sided documents, where all text is output on right-hand pages—like in one sided documents—and only special contents is output on left-hand pages on user request, *e. g.*, floating elements.

1 Introduction

Some institutions require a document layout where running text appears on right-hand pages only—as in one sided documents—, but figures and tables, and only those, are displayed on left-hand pages. This package supports the preparation of such semi one sided documents. That is, all running text is output on right-hand pages and only special content, *e. g.*, floating objects, are output on left-hand pages on user request. Otherwise left-hand pages stay empty.

References to left-hand pages are possible by defining a proper label on a left-hand page. The page counter can be configured to:

- keep left-hand pages syncronized with the facing right-hand page,
- keep left-hand pages syncronized with the preceeding right-hand page, *i. e.*, page numbers refer to single sheets, or
- count every left- and right-hand page as in normal two sided documents.

When using this package keep in mind that this is more a quick hack than a robust solution! Several problems are listed in section 4.[1]

*This document corresponds to semioneside v0.41, dated 2005/09/19.

[1]The best solution to avoid all sorts of problems is not to use this package. You have been warned!

2 How It Works

Before describing the usage of the semioneside package lets have a look at how it works behind the scenes.

The functionality of the semioneside package is based on the `\afterpage` macro from the `afterpage` package. What happens after a page break is essentially this:

1. Since the current page break happend to occur at the end of a right-hand page—remember that the first page of a document is a right-hand page—, a new left-hand page begins. Now the `afterpage` package calls the working macro `\semioneside@skipleftpage` which in turn calls these macros:

   ```
   \leftpagecontrolstart
   \leftpagecontent
   \leftpagecontrolend
   \clearpage
   \rightpagecontrolstart
   ```

2. The `\clearpage` command in the macro sequence above finishes the left-hand page and so macro `\rightpagecontrolstart` is called at the beginning of a new right-hand page.

3. Now, everything is ready to continue running text from the last right-hand page. At the next page break again `afterpage` calls `\semioneside@skipleftpage` and steps 1 to 3 are repeated.

The complete procedure is illustrated in Figure 1.

3 Usage

`\leftpagecontent`

As can be seen, four macros are called during type setting left- and right-hand pages. Macro `\leftpagecontent` is the main user macro. It contains the stuff that shall be output on the next left-hand page and can be modified via `\renewcommand`. If you want to have a figure on a left-hand page you should pass the whole figure environment including caption and label to `\leftpagecontent` (see the example code in file `example.tex`).

`\leftpagecontrolstart`
`\leftpagecontrolend`
`\rightpagecontrolstart`

The remaining three macros `\leftpagecontrolstart`, `\leftpagecontrolend` and `\rightpagecontrolstart` are hooks and can be used to adjust page numbering, the page style of left-hand pages *etc.* By default `\leftpagecontrolstart` calls `\thispagestyle{empty}`, macro `\leftpagecontrolend` is empty and macro `\rightpagecontrolstart` decreases the page counter by one. That is, left-hand pages have no head and foot line and the page number equals that of the facing right-hand page. Note, the page counter is not adjusted on a left-hand page, but on the following right-hand page.

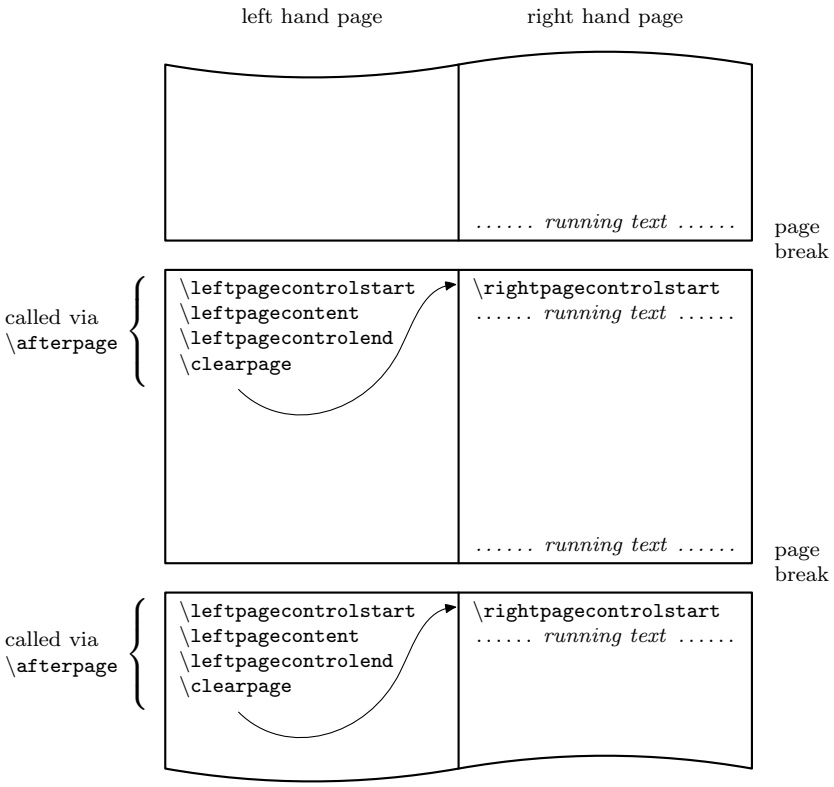

Figure 1: How the semioneside package intercepts page breaks.

Issuing the following macro sequence in the document preamble results in a layout where left-hand pages have the same page number as the preceeding right-hand page, *i. e.*, page numbers refer to sheets. Additionally page numbers are shown on left-hand pages:

```
\renewcommand*{\leftpagecontrolstart}{%
  \pagestyle{plain}%
  \addtocounter{page}{-1}%
}
\renewcommand*{\leftpagecontrolend}{}
\renewcommand*{\rightpagecontrolstart}{}
```

`\semionesideoff` `\semionesideon`

There are two more user macros, `\semionesideoff` and `\semionesideon`, which can be used to turn semi one sided layout off and on manually in the document. You might have to insert an empty page manually before `\semionesideon` to keep left- and right-hand pages syncronized. By default, semi one sided layout is activated.

4 Known Problems

Since this package is everything but a robust solution, there are some issues you have to take care of:

1. Contents that shall be output on a left-hand page has to be declared *on the preceeding* right-hand page. Since there is only a narrow window of approx. one page to declare left-hand page contents, it is recommended to finish running text first and only deal with left-hand page contents when page breaks of the document are stable.

2. Paragraphs longer than one page can cause the whole mechanism to get out-of-sync. This seems to be due to TEX's page breaking mechanism. When a paragraph breaks across several pages the `\afterpage` command is not executed at a page break (at least not reliably) and all gets mixed up. Beware of too long paragraphs!

3. When using document class `book` a blank page seems to be inserted before the first `\chapter` or `\maketitle` command. A workaround is to call macro `\semionesideoff` directly after loading the package in the document preamble and call macro `\semionesideon` manually after `\maketitle` or the first `\chapter` command. See also the next issue.

4. A similar problem was reported with the `memoir` class and `\part` command, which inserts two empty pages instead of only one. A workaround was to copy `memoir`'s definition of `\part` and remove a `\newpage` call there. In case you call this dirty hacking, you have been warned before.

5. When switching semi one sided layout on and off manually in the document pages can get out-of-sync. To avoid this, try to add a `\clearpage` or `\cleardoublepage` command before `\semionesideon`. If pages are still out-of- sync try to add the following macro sequence:

   ```
   \clear(double)page
   \leftpagecontrolstart\null\leftpagecontrolend
   \clearpage
   \rightpagecontrolstart
   ```

6. At the end of the document one or two empty pages may be appended. To get rid of those just call `\semionesideoff` before `\end{document}`.

I'm sure there are a lot more problems one can run into. In case you can't solve them by hand, try to set your document without the semioneside package and present this to your editor or boss for approval.[2] Since figures on left-hand pages were originally meant as a help when preparing manuscripts on type writers, there is no real benefit in it in digital document preparation.

[2] Don't tell them the rules are bad. Just asking if the document is ok, might be the better way.

5 Tips & Tricks

Symmetric and asymmetric type area This package works with the standard classes `article`, `report` and `book`. The `article` class leads to a symmetric layout. You can get a symmetric layout with the `report` or `book` classes, too, by calling them with the `oneside` document class option (see issue 3 in section 4). Omitting this option unfortunately leads to an alternating type area on right-hand pages. (I don't know why.) In case you need an asymmetric layout with the `book` class have a look at the `asymmetric` option of the `geometry` package.

6 Bugs and Development

As mentioned before, this package is no perfect solution to the problem it tries to solve, but rather a quick shot. Since I don't think I can improve on it much, I would appreciate if someone would take over development or rewrite this package from scratch to make the code more robust.

Anyway, if you find bugs (and hopefully workarounds) feel free to contact me at `stephanhennig@arcor.de`. I'll try to list them in the *Known Problems* section at least.

¡*package¿

7 Implementation

Make sure the `afterpage` package is loaded.

```
\RequirePackage{afterpage}
```

`\leftpagecontent` Define a macro that stores the contents that shall be output on a left-hand page.

```
\newcommand*{\leftpagecontent}{}
```

`\leftpagecontrolstart` Define a macro that may contain control statements that are executed at the beginning of left-hand pages before `\leftpagecontent`. Default is to make the left-hand page style `empty` here.

```
\newcommand*{\leftpagecontrolstart}{%
  \thispagestyle{empty}%
}
```

`\leftpagecontrolend` Define a macro that may contain control statements that are executed at the end of left-hand pages after `\leftpagecontent`. By default, it is empty.

```
\newcommand*{\leftpagecontrolend}{}
```

`\rightpagecontrolstart` Define a macro that may contain control statements that are executed at the beginning of right-hand pages before the running text continues. By default, the page counter is decreased by one here.

```
\newcommand*{\rightpagecontrolstart}{%
  \addtocounter{page}{-1}%
}
```

`\semioneside@longempty` Define a long *empty* macro which is used in a test (see `\ifx` below).

```
\newcommand*{\semioneside@longempty}{}
```

`\semioneside@skipleftpage` This is the working macro. It calls itself recursively via `\afterpage`. The recursion can be started by issuing the command `\semionesideon`. In fact, this is already done when the package is loaded.

```
\newcommand*{\semioneside@skipleftpage}{}
```

`emioneside@skipleftpagetemplate` Define a template for the working macro `\semioneside@skipleftpage`. This is used in `\semionesideon`.

```
\newcommand*{\semioneside@skipleftpagetemplate}{%
  \ifx\leftpagecontent\empty%
    \let\leftpagecontent\null%
  \fi%
  \ifx\leftpagecontent\semioneside@longempty%
    \let\leftpagecontent\null%
  \fi%
  \leftpagecontrolstart%
  \leftpagecontent%
  \leftpagecontrolend%
  \clearpage%
  \rightpagecontrolstart%
  \afterpage{\semioneside@skipleftpage}%
  \global\let\leftpagecontent\empty%
}
```

`\semionesideon` Define a macro to activate semi one sided layout.

```
\newcommand*{\semionesideon}{%
  \global\let\semioneside@skipleftpage\semioneside@skipleftpagetemplate%
  \afterpage{\semioneside@skipleftpage}%
}
```

`\semionesideoff` Define a macro to deactivate semi one sided layout.

```
\newcommand*{\semionesideoff}{%
  \global\let\semioneside@skipleftpage\empty%
  \global\let\leftpagecontent\empty%
}
```

Switch on semi one sided layout.

```
\semionesideon
```

¡/package¿

Change History

v0.4 revised documentation 1

General: .dtx package version with

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.