The ifoddpage Package

Martin Scharrer
martin.scharrer@web.de

CTAN: http://www.ctan.org/pkg/ifoddpage

VC: https://github.com/MartinScharrer/ifoddpage

Version v1.2 – 2022/10/18

1 Introduction

Sometimes it is required to know if a macro is placed on an odd or even page. Simply testing the value of the page counter is not reliable, because TEX reads and processes often more material than fits on the currently processed page in order to decide where to place the page break. Also moving material like floats can be placed on completely different pages than where the surrounding material is placed. The only reliable way to determine the correct page number is to write it to the auxiliary file using a label and read it back in the next compiler run. Labels use a so called *whatsit*s (see *The TeXBook*) to write the given code to the auxiliary file *just when the material is shipped out*, i.e. a full page has been filled and is written to the output file. This way ensures that the determined page number is always correct, but adds an overhead.

This package implements the described technique. A macro must be used to generate and read-back the label before the conditional can be used. A counter is used internally to provide unique names to the labels.

2 Usage

The following macros are provided for LATEX document authors:

`\checkoddpage`

This macro checks if the current page is odd or even by placing a label to the auxiliary file and reading it back in. This requires a second compiler run to work perfectly, but the code falls back to use the page counter value if the label is not yet defined, i.e. the label was just added or no auxiliary file existed. This fallback code might return wrong values for material just around page breaks, but is better than using a constant.

Note that this macro inserts an invisible element (a so called *whatsit*) which is placed on the page. It is possible that this happens to be at the very end of the last line of the page and any other code following it would then be placed on the next page. Some code like `\checkoddpage \ifoddpage odd\else even\fi` might then place the first macro on the current, say even, page but print '`even`' at the top of the next page which is odd. In order to avoid this both the check macro and

the output text can be placed in the same box (e.g. \mbox). If the full content can't be placed in a box, e.g. because line breaking is required a \mbox can also be substituded by \leavevmode (to start a paragraph if required) and \hbox where the {} are replaced by \bgroup and \egroup so two alternative box ends can be specified:

```
\leavevmode\hbox\bgroup
\checkoddpage
\ifoddpage
  odd\egroup\ more text
\else
  even\egroup\ other text
\fi
```

This boxes the \checkoddpage with the first word of both clauses. Note that in this case it the settings are local to the box scope and \ifoddpage should be used outside of it again (without another \checkoddpage).

\ifoddpage

This TEX conditional is set (locally) by \checkoddpage. It is true if the current page is odd, or false if it is even. This is independent from whether the oneside or twoside mode is used.

\ifoddpageoroneside

This TEX conditional is set (locally) by \checkoddpage. Like the previous conditional it is true if the current page is odd, or false if it is even. However, if the oneside mode is active it is always true. This is useful for code which needs to check if the odd or even page layout is used, because in oneside mode the odd page layout is used for all pages.

The following macros are intended for package authors:

\@ifoddpage{⟨*true*⟩}{⟨*false*⟩}

This LATEX macro uses \ifoddpage and executes its first argument if that conditional is true but the second argument if it is false. The \checkoddpage must be used closely beforehand to get correct results.

\@ifoddpageoroneside{⟨*true*⟩}{⟨*false*⟩}

This LATEX macro uses \ifoddpageoroneside and executes its first argument if that conditional is true but the second argument if it is false. The \checkoddpage must be used closely beforehand to get correct results.

\oddpage@page

This macro expands (using multiple steps) to a text representation of the page number for the last \checkoddpage. It is used in this macro together with \ifodd to set the provided conditionals.

3 Similar packages

The `changepage` package and the `memoir` class also provide the core functionality of this package, i.e. they define `\checkoddpage` and `\ifoddpage` but no other of the mentioned macros. That package uses this conditional internally in order to allow the user to change the page layout for odd or even pages, which is its main function. The `ifoddpage` should work together with that package and class, but should be loaded last to ensure the correct functionality of `\ifoddpageoroneside`. It also uses labels if it is loaded using the `strict` option. The `ifoddpage` has the following benefits over `changepage` (if the page layout doesn't have to be changed):

- Smaller code base, because no additional functionality is provided.
- The `\ifoddpageoroneside` conditional is also provided which simplifies code which has to handle odd/even margins.
- The label code uses the same internal LaTeX-core code as the normal `\label` and generates identical error, warning and info messages.
- The fallback page number is not constant (0) but the current `page` counter value is used, which is a good approximation.

4 Implementation

```
%<!COPYRIGHT>
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\ProvidesPackage{ifoddpage}[%
%<!DATE>
%<!VERSION>
%<*DRIVER>
    2099/01/01 develop
%</DRIVER>
    Conditionals for odd/even page detection]
```

\c@checkoddpage

Counter checkoddpage is needed to give each use of \checkoddpage an unique ID for

```
\newcount\c@checkoddpage
```

\thecheckoddpage

Expands to the value of checkoddpage, i.e. ID of last used \checkoddpage.

```
\def\thecheckoddpage{\number\c@checkoddpage}
```

\ifoddpage

Conditional: true if used on odd page after \checkoddpage.

```
\newif\ifoddpage
```

\ifoddpageoroneside

Conditional: true if used on odd page after \checkoddpage. Always true if used in a oneside document.

```
\newif\ifoddpageoroneside

\let\oddpage@checkpage\relax
```

\oddpage@page

```
\def\oddpage@page{1}
```

\oddpage@label

Produces a LaTeX label using the checkoddpage counter. An internal LaTeX macro is used for this.

```
\def\oddpage@label{%
    \@newl@bel{checkoddpage}%
}%
\write\@auxout{\noexpand\providecommand\noexpand\
   oddpage@label[2]{}}%
```

\oddpage@checkpage

Writes the oddpage label with the current page number to the AUX file, so it can be read back during the following LaTeX runs.

```
\def\oddpage@checkpage#1{%
    \@bsphack
    \write\@auxout{\string\oddpage@label{#1}{\the\
       c@page}}%
    \@esphack
}%
```

\oddpage@page

Returns the page number of the last \checkoddpage macro. If there is no label for it defined in the AUX file, e.g. first compile run, then the current page number is used instead as a fallback.

```
\def\oddpage@page{%
    \expandafter\ifx\csname checkoddpage@\
       thecheckoddpage\endcsname\relax
        \the\c@page
    \else
        \csname checkoddpage@\thecheckoddpage\
           endcsname
    \fi
}%
```

\checkoddpage

User macro to check if the current page has an odd page number. Increases checkoddpage counter to produce an unique ID for this macro usage. Calls \oddpage@checkpage to produce a reference entry in the AUX file. Then checks if the \oddpage@page is odd and sets \ifoddpage and \ifoddpageoroneside accordantly. Finally checks if the twoside setting is active and set \ifoddpageoroneside to true if so.

```
\DeclareRobustCommand\checkoddpage{%
    \stepcounter{checkoddpage}%
    \expandafter\oddpage@checkpage\expandafter{\
       number\c@checkoddpage}%
    \ifodd\oddpage@page\relax
        \oddpagetrue
        \oddpageoronesidetrue
    \else
        \oddpagefalse
        \oddpageoronesidefalse
    \fi
    \if@twoside\else
        \oddpageoronesidetrue
    \fi
}%
```

\@ifoddpage

Wrapper around \ifoddpage. Will expand to the next token if odd, to the second token if not.

```
\def\@ifoddpage{%
    \ifoddpage
        \expandafter\@firstoftwo
    \else
        \expandafter\@secondoftwo
    \fi
}
```

\@ifoddpageoroneside

Wrapper around \ifoddpageoroneside. Will expand to the next token if odd, to the second token if not.

```
\def\@ifoddpageoroneside{%
    \iftwosideoddside
        \expandafter\@firstoftwo
    \else
        \expandafter\@secondoftwo
    \fi
}
```