

# The `pagesel` package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2008/08/11 v1.8

## Abstract

Single pages or page areas can be selected for output.

## Contents

<b>1</b>	<b>Usage</b>	<b>1</b>
1.1	Page selecting	2
1.1.1	Options for selecting pages	2
1.1.2	Examples	2
1.2	Auxiliary files	2
1.2.1	Options for handling auxiliary files	2
1.2.2	Package <code>hyperref</code>	3
<b>2</b>	<b>Implementation</b>	<b>3</b>
2.1	Package	3
2.2	AtBeginDvi hook support	6
<b>3</b>	<b>Installation</b>	<b>7</b>
3.1	Download	7
3.2	Bundle installation	7
3.3	Package installation	8
3.4	Refresh file name databases	8
3.5	Some details for the interested	8
<b>4</b>	<b>History</b>	<b>9</b>
	[1999/03/01 v0.9]	9
	[1999/04/05 v1.0]	9
	[1999/04/13 v1.1]	9
	[2003/06/05 v1.2]	9
	[2006/02/20 v1.3]	9
	[2006/03/02 v1.4]	9
	[2006/03/07 v1.5]	9
	[2007/04/11 v1.6]	9
	[2007/04/12 v1.7]	9
	[2008/08/11 v1.8]	9
<b>5</b>	<b>Index</b>	<b>10</b>

## 1 Usage

The package `pagesel` is a  $\text{\LaTeX}$  2 $\epsilon$  package:

```
\usepackage[<options>]{pagesel}
```

(For plain $\text{\TeX}$  and  $\text{\LaTeX}$  2.09 the similar package `selectp`<sup>1</sup> from Donald Arsenau<sup>2</sup> can be used.)

Depending on the options the package works in two modes:

1. If no page selecting option is present, so the package ignores the other options and finishes itself. So no page will be suppressed by the package and auxiliary files will be written.
2. With at least one page selecting option the specified pages are selected and the other are suppressed. The default for this mode is that auxiliary will not be overwritten. (This can be changed by an option.)

## 1.1 Page selecting

The package `pagesel` sets up a new counter that is incremented by each `\shipout`. In this way the package counts the output pages regardless the value of the page counter. So each page can individually be addressed, even if there are several pages with the same page number.

### 1.1.1 Options for selecting pages

**odd:** The output pages must have an odd number. All even output pages are suppressed. If there are no page areas specified so all odd pages are print. With page areas only the odd pages in this areas are selected.

**even:** The opposite of option `odd`.

**Page area:** A page area consists of three elements: the starting output page number, an “area” hyphen, and the output page number of the last page in this area. Each component is optional, so there are four kinds to specify a page area:

`<m>-<n>`: All pages between `<m>` and `<n>` inclusive.

`-<n>`: All pages until `<n>` inclusive.

`<m>-:` The page area starts with `<m>` and all pages to the end of document are selected.

`-:` All pages (not very useful).

`<s>`: The single page `<s>`.

### 1.1.2 Examples

Options	Output pages
<code>[1, 4, 9]</code>	1, 4, and 9
<code>[7-10, 3]</code>	3, 7, 8, 9, and 10
<code>[odd, 3-6]</code>	3, and 5
<code>[-4, 3, even, 7-8]</code>	2, 4, and 8

## 1.2 Auxiliary files

If a page is suppressed, the `\write` commands are not performed. Labels, index entries, or entries for the table of contents aren’t written. So it is likely that the table of contents, registers, and lists are incomplete.

### 1.2.1 Options for handling auxiliary files

**nofiles:** This is the default. Auxiliary files are read but not written or changed. Also the job is aborted after the last selected page for saving time.

**nonofiles/files:** Auxiliary files are written.

<sup>1</sup>Url: <ftp://ftp.ctan.org/tex-archive/macros/latex/contrib/misc/selectp.sty>

<sup>2</sup>Donald Arsenau’s email address: [asnd@triumf.ca](mailto:asnd@triumf.ca)

### 1.2.2 Package hyperref

In old versions of `hyperref` [1999/04/12 v6.55] (and below) there is a bug with `\nofiles`:

- Some “garbage” appears on terminal and in the log file. This is harmless and can be ignored.
- The outline auxiliary file `\jobname.out`, however, is opened and truncated to zero bytes. Version 1.0 of this package had loaded a patch file `hypnofil.tex`, if it detects `hyperref` to get `\nofiles` work.

With the new version of `hyperref` [1999/04/13 v6.56] `\nofiles` works now. Therefore the workaround code is no longer needed and removed.

## 2 Implementation

### 2.1 Package

```
1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{pagesel}
4 [2008/08/11 v1.8 Selecting shipout output pages (H0)]%
```

If the package is loaded twice, the package code does not work. So stop loading the package, if it is already loaded.

```
5 \@ifundefined{ps@makevoid}{-}{%
6   \PackageWarningNoLine{pagesel}{Package already loaded.}%
7   \endinput
8 }
```

`\ps@makevoid` Macro `\ps@makevoid` clears the output box. Because nothing is shipped out and this is intended, we reduce the counter `\deadcycles` in order to avoid problems, if more than `\maxdeadcycles` pages are omitted.

```
9 \newcommand*{\ps@makevoid}{%
10   \global\setbox\@cclv\copy\voidb0x
11   \begingroup
12     \count@=\deadcycles
13     \advance\count@ by -1\relax
14     \deadcycles=\count@
15   \endgroup
16 }
```

`\ps@oddpages`

```
17 \newcommand*{\ps@oddpages}{0}
18 \DeclareOption{odd}{\renewcommand*{\ps@oddpages}{1}}
19 \DeclareOption{even}{\renewcommand*{\ps@oddpages}{2}}

20 \DeclareOption{nofiles}{\let\ps@nofiles\nofiles}
21 \DeclareOption{nonofiles}{\let\ps@nofiles\@empty}
22 \DeclareOption{files}{\let\ps@nofiles\@empty}
23 \ExecuteOptions{nofiles}

24 \DeclareOption*{%
25   \begingroup
26     \expandafter\ps@checkoption\CurrentOption-\END
27     \edef\x{\endgroup\noexpand\ps@store{\ps@first}{\ps@last}}}%
28   \x
29 }
```

`\ps@checkoption`

```
30 \newcommand\ps@checkoption{}
31 \def\ps@checkoption#1-#2\END{%
32   \ifx\\#2\\%
```

```

33 \ifx\#1\%
34 % empty option
35 \def\ps@first{\maxdimen}%
36 \def\ps@last{\maxdimen}%
37 \else
38 \edef\ps@first{#1}%
39 \edef\ps@last{#1}%
40 \fi
41 \else
42 \ifx\#1\%
43 \def\ps@first{-\maxdimen}%
44 \else
45 \edef\ps@first{#1}%
46 \fi
47 \ps@checklast#2%
48 \fi
49 }

```

\ps@checklast

```

50 \newcommand\ps@checklast{}
51 \def\ps@checklast#1-{%
52 \ifx\#1\%
53 \def\ps@last{\maxdimen}%
54 \else
55 \edef\ps@last{#1}%
56 \fi
57 }

```

\ps@store

```

58 \newcommand*\ps@store}[2]{%
59 \expandafter\def\expandafter\ps@testlist\expandafter{%
60 \ps@testlist\ps@pagetest{#1}{#2}%
61 }%
62 }

```

\ps@testlist

```

63 \newcommand*\ps@testlist{}
64 \ProcessOptions
65 \begingroup
66 \edef\x{%
67 \ifnum\ps@oddpages>0 \relax\fi
68 \ifx\ps@testlist\@empty\else\relax\fi
69 }%
70 \ifx\x\@empty
71 \endgroup
72 \PackageInfo{pagesel}{Nothing to do}%
73 \expandafter\endinput
74 \fi
75 \endgroup
76 \RequirePackage{everyshi}
77 \ps@nofiles

```

\c@ps@count

```

78 \newcounter{ps@count}
79 \setcounter{ps@count}{0}

```

\ps@ReturnAfterElseFi

\ps@ReturnAfterFi

```

80 \long\def\ps@ReturnAfterElseFi#1\else#2\fi{\fi#1}
81 \long\def\ps@ReturnAfterFi#1\fi{\fi#1}

```

```

82 \newcommand{\ps@lastpage}{\maxdimen}
83 \ifx\ps@nofiles\nofiles
84   \ifx\ps@testlist\@empty
85   \else
86     \def\ps@lastpage{0}%
87     \newcommand*{\ps@pagetest}[2]{%
88       \ifnum#2>\ps@lastpage\relax
89       \def\ps@lastpage{#2}%
90       \fi
91     }%
92     \ps@testlist
93     \let\ps@pagetest\relax
94   \fi
95 \fi

\ps@ifinset

96 \newcommand*{\ps@ifinset}[4]{%
97   \ifnum#1>\value{ps@count}%
98     \ps@ReturnAfterElseFi{#4}%
99   \else
100     \ps@ReturnAfterFi{%
101       \ifnum#2<\value{ps@count}%
102         \ps@ReturnAfterElseFi{#4}%
103       \else
104         \ps@ReturnAfterFi{#3}%
105       \fi
106     }%
107   \fi
108 }

\ps@pagetest

109 \newcommand*{\ps@pagetest}[2]{%
110   \ps@ifinset{#1}{#2}{\let\ps@next\@empty}{}%
111 }

112 \EveryShipout{%
113   \stepcounter{ps@count}%
114   \ifnum\value{ps@count}>\ps@lastpage\relax
115     \global\output{%
116       \ps@cleanup@if
117       \ps@group@message
118       \typeout{%
119         Package pagesel Notice: Aborting LaTeX job %
120         after last selected page (\ps@lastpage).%
121       }%
122       \ps@message@ignore
123       \global\setbox\@ccclv\box\voidb@x
124       \deadcycles0\relax
125     }
126   }%
127 \fi
128 \let\ps@next\@empty
129 \ifx\ps@testlist\@empty
130 \else
131   \let\ps@next\ps@makevoid
132   \ps@testlist
133 \fi
134 \ifnum\ps@oddpages=1 %
135   \ifodd\value{ps@count}%
136   \else
137     \let\ps@next\ps@makevoid

```

First leave the output group before ending the job.

```

138   \fi
139 \fi
140 \ifnum\ps@oddpages=2 %
141   \ifodd\value{ps@count}%
142     \let\ps@next\ps@makevoid
143   \else
144     \fi
145   \fi
146 \ps@begindvi
147 \ps@next
148 }

149 \begingroup\expandafter\expandafter\expandafter\endgroup
150 \expandafter\ifx\csname currentiflevel\endcsname\relax
151   \let\ps@cleanup@if\@empty
152 \else
153   \def\ps@cleanup@if{%
154     \ifnum\currentiflevel>\@ne
155       \csname fi\endcsname
156       \expandafter\ps@cleanup@if
157     \fi
158   }%
159 \fi

```

Because of `\aftergroup` it is too dangerous to perform a similar cleanup for groups.

```

160 \begingroup\expandafter\expandafter\expandafter\endgroup
161 \expandafter\ifx\csname currentgrouplevel\endcsname\relax
162   \let\ps@group@message\@empty
163   \def\ps@message@ignore{%
164     \typeout{%
165       (pagesel) \space\space\@spaces\@spaces\@spaces
166       Messages (\string\end\space occurred ...) can be ignored.%
167     }%
168   }%
169 \else
170   \def\ps@group@message{%
171     \ifnum\currentgrouplevel>\@ne
172       \def\ps@message@ignore{%
173         \typeout{%
174           (pagesel) \space\space\@spaces\@spaces\@spaces
175           Message (\string\end\space occurred ...) %
176           can be ignored.%
177         }%
178       }%
179     \else
180       \let\ps@message@ignore\@empty
181     \fi
182   }%
183 \fi

```

## 2.2 AtBeginDvi hook support

The material of box `\@begindvibox` is recorded in parallel in box `\ps@begindvibox`.

```

184 \newbox\ps@begindvibox
185 \ifvoid\@begindvibox
186 \else
187   \global\setbox\ps@begindvibox\vbox{%
188     \unvbox\@begindvibox
189   }%
190 \fi
191 \let\ps@org@AtBeginDvi\AtBeginDvi
192 \def\AtBeginDvi#1{%

```

```

193 \global\setbox\ps@begindvibox\ vbox{%
194   \unvbox\ps@begindvibox
195   #1%
196 }%
197 \ps@org@AtBeginDvi{#1}%
198 }

```

`\ps@begindvi` Macro `\ps@begindvi` is called the similar way as `\@begindvi`. If the first page is printed, then `\AtBeginDvi` should work as usual. Otherwise the contents of box `\ps@begindvibox` is set on the first selected page.

```

199 \def\ps@begindvi{%
200   \ifx\ps@next\@empty
201     \global\let\ps@begindvi\@empty
202   \else
203     \global\let\ps@begindvi\ps@begindvi@do
204   \fi
205 }
206 \def\ps@begindvi@do{%
207   \ifx\ps@next\@empty
208     \setbox\@cclv\ vbox{%
209       \unvbox\ps@begindvibox
210       \box\@cclv
211     }%
212     \global\let\ps@begindvi\@empty
213   \fi
214 }
215 </package>

```

## 3 Installation

### 3.1 Download

**Package.** This package is available on CTAN<sup>3</sup>:

[CTAN:macros/latex/contrib/oberdiek/pagesel.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/pagesel.pdf](#) Documentation.

**Bundle.** All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

*TDS* refers to the standard “A Directory Structure for  $\text{\TeX}$  Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 3.2 Bundle installation

**Unpacking.** Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```

chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/

```

---

<sup>3</sup>[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

### 3.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain- $\text{\TeX}$ :

```
tex pagesel.dtx
```

**TDS.** Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
pagesel.sty → tex/latex/oberdiek/pagesel.sty
pagesel.pdf → doc/latex/oberdiek/pagesel.pdf
pagesel.dtx → source/latex/oberdiek/pagesel.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

### 3.4 Refresh file name databases

If your  $\text{\TeX}$  distribution (`te $\text{\TeX}$` , `mik $\text{\TeX}$` , ...) relies on file name databases, you must refresh these. For example, `te $\text{\TeX}$`  users run `texhash` or `mktextlsr`.

### 3.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk pagesel.pdf unpack_files output .
```

**Unpacking with  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain- $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{pagesel.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

**Generating the documentation.** You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf $\text{\LaTeX}$` :

```
pdfflatex pagesel.dtx
makeindex -s gind.ist pagesel.idx
pdfflatex pagesel.dtx
makeindex -s gind.ist pagesel.idx
pdfflatex pagesel.dtx
```



## 4 History

[1999/03/01 v0.9]

- The first version was built as a response to a question of Dirk Kuypers<sup>4</sup>, published in the newsgroup `de.comp.text.tex`:  
“Re: pdf<sub>l</sub>atex nur fuer bestimmte Seiten?!?”<sup>5</sup>

[1999/04/05 v1.0]

- Documentation added in dtx format.
- Copyright: LPPL (`CTAN:macros/latex/base/lppl.txt`)
- Options `odd`, `even` added.
- `\nofiles` added, bug fix for `hyperref`.
- Abort loading of package, if nothing to do.

[1999/04/13 v1.1]

- `\nofiles` bug fix removed because of `hyperref` 6.55.
- First CTAN release.

[2003/06/05 v1.2]

- `\deadcycles` is decremented for omitted pages.
- LPPL 1.2.

[2006/02/20 v1.3]

- Code is not changed.
- New DTX framework.
- LPPL 1.3

[2006/03/02 v1.4]

- Support for `\AtBeginDvi` added.

[2006/03/07 v1.5]

- Job is aborted after last selected page.

[2007/04/11 v1.6]

- Line ends sanitized.

[2007/04/12 v1.7]

- Hard coded box number 255 replaced by macro `\@cc1v`.

[2008/08/11 v1.8]

- Code is not changed.
- URL updated from `www.dejanews.com` to `groups.google.com`.

---

<sup>4</sup>Dirk Kuypers’s email address: `dk@comnets.rwth-aachen.de`

<sup>5</sup>Url: `http://groups.google.com/group/de.comp.text.tex/msg/6b68c7b3439fb658`

## 5 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.

Symbols	
\@@end .....	125
\@begindvibox .....	185, 188
\@cclv .....	10, 123, 208, 210
\@empty .....	21,
	22, 68, 70, 84, 110, 128, 129,
	151, 162, 180, 200, 201, 207, 212
\@ifundefined .....	5
\@ne .....	154, 171
\@spaces .....	165, 174
\@ .....	32, 33, 42, 52
A	
\advance .....	13
\aftergroup .....	125
\AtBeginDvi .....	191, 192
B	
\box .....	123, 210
C	
\c@ps@count .....	78
\copy .....	10
\count@ .....	12, 13, 14
\csname .....	150, 155, 161
\currentgrouplevel .....	171
\currentiflevel .....	154
\CurrentOption .....	26
D	
\deadcycles .....	12, 14, 124
\DeclareOption ..	18, 19, 20, 21, 22, 24
E	
\END .....	26, 31
\end .....	166, 175
\endcsname .....	150, 155, 161
\endinput .....	7, 73
\EveryShipout .....	112
\ExecuteOptions .....	23
I	
\ifnum .....	67,
	88, 97, 101, 114, 134, 140, 154, 171
\ifodd .....	135, 141
\ifvoid .....	185
\ifx .....	32, 33, 42, 52, 68,
	70, 83, 84, 129, 150, 161, 200, 207
M	
\maxdimen .....	35, 36, 43, 53, 82
N	
\NeedsTeXFormat .....	2
\newbox .....	184
\newcommand .....	9,
	17, 30, 50, 58, 63, 82, 87, 96, 109
\newcounter .....	78
\noindent .....	20, 83
O	
\output .....	115
P	
\PackageInfo .....	72
\PackageWarningNoLine .....	6
\ProcessOptions .....	64
\ProvidesPackage .....	3
\ps@begindvi .....	146, 199
\ps@begindvi@do .....	203, 206
\ps@begindvibox .....	184, 187, 193, 194, 209
\ps@checklast .....	47, 50
\ps@checkoption .....	26, 30
\ps@cleanup@if .....	116, 151, 153, 156
\ps@first .....	27, 35, 38, 43, 45
\ps@group@message .....	117, 162, 170
\ps@ifinset .....	96, 110
\ps@last .....	27, 36, 39, 53, 55
\ps@lastpage ..	82, 86, 88, 89, 114, 120
\ps@makevoid .....	9, 131, 137, 142
\ps@message@ignore ..	122, 163, 172, 180
\ps@next .....	110,
	128, 131, 137, 142, 147, 200, 207
\ps@nofiles .....	20, 21, 22, 77, 83
\ps@oddpages .....	17, 67, 134, 140
\ps@org@AtBeginDvi .....	191, 197
\ps@pagetest .....	60, 87, 93, 109
\ps@ReturnAfterElseFi ...	80, 98, 102
\ps@ReturnAfterFi .....	80, 100, 104
\ps@store .....	27, 58
\ps@testlist .....	
	59, 60, 63, 68, 84, 92, 129, 132
R	
\renewcommand .....	18, 19
\RequirePackage .....	76
S	
\setbox .....	10, 123, 187, 193, 208
\setcounter .....	79
\space .....	165, 166, 174, 175
\stepcounter .....	113
T	
\typeout .....	118, 164, 173
U	
\unvbox .....	188, 194, 209
V	
\value .....	97, 101, 114, 135, 141
\vbox .....	187, 193, 208
\voidb@x .....	10, 123
X	
\x .....	27, 28, 66, 70