

The `totpages` package*

Wilhelm Müller
Herbststraße 66
D-28215 Bremen
muewi@acm.org

2005/09/19

Abstract

This package counts the total number of pages shipped out to the *dvi* file, which allows sentences like “This document contains 10 pages, the last one being no. 1243.” to be produced automatically. The counter itself may be shipped out to the *dvi* file, too, allowing page access by absolute page numbers, or it may be just a regular L^AT_EX counter.

It cooperates with the package `hyperref`, producing a hyperlink to the last page—though this is not of much use as hypertext readers usually allow direct access to the last page anyway.

Optionally, the number of sheets of paper needed to print the document can be computed.

Contents

1	Introduction	1234	4	Required packages	1236
2	Using <code>totpages</code>	1235	5	Acknowledgements	1236
3	Options	1235	6	The implementation	1237
3.1	Writing absolute page numbers to the <i>dvi</i> file . . .	1235	7	An example file	1242
3.2	Computing the number of sheets of paper	1235			

1 Introduction

Printing only part of a *dvi* file can be difficult or impossible if pages in different parts of the document have the same number—e.g., in the front matter and the first text pages (iii vs. 3).

T_EX provides an easy solution to this problem: whenever a page has been completed by the output routine and is being shipped out to the *dvi* file, it displays the values of `\count0` to `\count9`, with trailing zeroes suppressed, (e.g., `[1002.0.3]`) and writes them to the *log* and *dvi* files, cf. [Knuth, p. 119]. So, if you have a *dvi* driver which allows page selection by other counters than `\count0`, you will be able to refer to absolute page numbers when you make use of this package.

*The version number of this file is v2.00, last revised 2005/09/19.

2 Using `totpages`

`TotPages` Keeping a count of the absolute page number makes it possible to provide the *total* number of pages of the document (as opposed to the “number” on the last page provided by the `lastpage` package, cf. [Goldberg]).

References to the counter itself won’t make much sense since it is updated asynchronously whenever an actual shipout occurs. But by referencing the label `TotPages` (with `\ref{TotPages}`) you get the total number of pages the document had at the end of the previous run of L^AT_EX. If, for some reason, you want to know what page number L^AT_EX actually (would have) printed, you may use `\pageref{TotPages}`.¹

`\theTotPages` What is actually printed by `\ref{TotPages}` is the result of the macro `\theTotPages`, which normally is simply `\arabic{TotPages}`, but may be redefined to anything (sensible) somewhere before `\end{document}`.

If you want to use this package, you should load `totpages` as the *last* package in your document because it executes some code at `\end{document}` and has to be sure that its code is the last code to be executed there.²

You *must not* use `count1to` or `lastpage` together with this package. Package `count1to` provides additional functionality, but will not cooperate with `hyperref`, whereas any use of `lastpage` is covered by `totpages`.³ If you try to use one (or both) of these packages together with `totpages`, you will receive an error message at `\begin{document}`. I don’t think that something really bad will happen if you ignore it (unless you use `count1to` together with `hyperref`), but I won’t guarantee anything, either.

3 Options

3.1 Writing absolute page numbers to the *dvi* file

`dvi` With option `dvi` (the default), `\count1` is used for the absolute page counter.
`nodvi` This value is put into the *dvi* file and may be used by the *dvi* driver.

Option `nodvi` uses a regular L^AT_EX counter which will not show up elsewhere.

No matter what kind of counter is used, its name will always be `TotPages`, e.g., if you want to change its format to `roman`. (But who would want to do that?).

3.2 Computing the number of sheets of paper

`pagespersheet` The option `pagespersheet=n` allows to compute the number of physical sheets of
`nopagespersheet` paper needed when the document will be printed.⁴ To compute this value, you
 New feature have to specify the positive integer, *n*, telling how many pages will be printed
 Provide number of sheets on one physical sheet of paper. For example, when you use a duplex printer and
 needed for printing let your printer driver scale and rearrange document pages to print two docu-
 ment pages on one physical page, you should specify `pagespersheet=4`. Using

¹Since all references to this label are obviously forward references, you will have to run L^AT_EX at least twice to get correct results.

²See [Goldberg] for a discussion of problems.

³`lastpage` requires less resources, though.

⁴Thanks to Ido Tal for suggesting this feature, providing an initial realisation, testing it, and proof-reading the documentation.

`nopagespersheet` (or `pagespersheet=no`) turns this computation off; this is the default setting.

`\TotPerSheet` When `pagespersheet` is set to valid number, its value is available in the macro `\TotPerSheet` (in exactly the same form as you wrote it), otherwise this macro is undefined.

`\TotSheets` The number of sheets, as computed at the end of the previous \LaTeX run, will be contained (as decimal digits) in the macro `\TotSheets`.⁵ When `pagespersheet` is used for the first time, the value of this macro will be set to 0000 (four zeroes), which is still a number but different from the representation of an actual zero. When you say `nopagespersheet` (or never use `pagespersheet` at all), `\TotSheets` will be undefined. When, at the end of the document, the value of `\TotSheets` differs from the one at `\begin{document}` (the initial 0000, or as read from *aux*), a warning will be issued similar to the one you get when labels differ.

4 Required packages

This package needs the `everyshi` [Schröder 2] and `keyval` [Carlisle] packages.

5 Acknowledgements

This package uses ideas inspired by Martin Schröder's `count1to` package [Schröder 1], Jeff Goldberg's `lastpage` package [Goldberg], and the `lastpage` and `keyval` support contained in Sebastian Rahtz's `hyperref` package [Rahtz], and implements options suggested by and based on ideas of Ido Tal [Tal].

References

- [Carlisle] David Carlisle. The `keyval` package. CTAN:tex-archive/macros/latex/required/graphics/keyval.dtx. \LaTeX 2 ϵ package.
- [Goldberg] Jeff Goldberg. The `lastpage`-package. CTAN:tex-archive/macros/latex/contrib/lastpage/. \LaTeX 2 ϵ package.
- [Knuth] Donald E. Knuth. *The \TeX Book*, volume A of *Computers and Typesetting*. Addison-Wesley, 1986. Tenth printing, revised, June 1990.
- [Rahtz] Sebastian Rahtz. Hypertext marks in \LaTeX . CTAN:tex-archive/macros/latex/contrib/hyperref/. \LaTeX 2 ϵ package.
- [Schröder 1] Martin Schröder. The `count1to` package. CTAN:tex-archive/macros/latex/contrib/ms/count1to.dtx. \LaTeX 2 ϵ package.

⁵The value of `pagespersheet` must, of course, coincide with the options you use when you print the document.

[Schröder 2] Martin Schröder. The everyshi package. CTAN:
 tex-archive/macros/latex/contrib/ms/everyshi.dtx.
 L^AT_EX 2_ε package.

[Tal] Ido Tal. Private communication.

6 The implementation

We need the everyshi and keyval packages.

```

1 (*package)
2 \ProvidesPackage{totpages}
3     [2005/09/19 v2.00 Totpages Package (muewi)]
4 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
5 \RequirePackage{everyshi}[1994/12/09]
6 \RequirePackage{keyval}[1998/06/05]
```

`\ifTotPagesToDvi` The user can decide if the absolute page number should go to the *dvi* file or not, the default being yes:

```
7 \newif\ifTotPagesToDvi\TotPagesToDvittrue
```

`\ifPagesPerSheet`
`\TotPerSheet` The user may ask to compute the number of sheets needed for printing. To do this, we need some variables. (The number of pages per sheet is saved in a macro instead of a counter since counters are a scarce resource in some T_EX implementations.)

```
8 \newif\ifPagesPerSheet\PagesPerSheetfalse
9 \newcommand{\TotPerSheet}{}
```

We want to parse package options with keyval. This code has been stolen (more or less) from `hyperref`:

First, we define the keys. In case of the boolean keys, `dvi` and `nodvi`, we ignore any value, just taking into account their presence:

```
10 \define@key{Tot}{dvi}[x]{\TotPagesToDvittrue}
11 \define@key{Tot}{nodvi}[x]{\TotPagesToDvifalse}
```

For the numeric key, a value of `no` may be used to disable the feature.

```

12 \define@key{Tot}{nopagespersheet}[x]{\PagesPerSheetfalse}
13 \define@key{Tot}{pagespersheet}{%
14   \lowercase{\def\Tot@temp{#1}}%
15   \ifx\Tot@temp\@empty
16     \else
17       \def\Tot@no{no}
18       \ifx\Tot@temp\Tot@no
19         \else
20           \edef\TotPerSheet{#1}
21           \PagesPerSheettrue
22         \fi
23       \fi
24 }
```

`\Tot@optionsWithKV` Now comes the actual option processing. We won't use the code from `hyperref` literally since it will produce a warning when something like `pagespersheet=2` is specified as a global option (although it's processed correctly!). Nevertheless,

this code has been derived from the code in `hyperref`, which, in turn, contains the following attribution: “This section was written by David Carlisle.”

```
25 \def\Tot@optionsWithKV#1{%
26   \let\@tempc\relax
27   \let\Tot@tempa\@empty
```

`\Tot@tempopt` To cope with global options of the form `key=value`, we have to look for a definition of `key`, but remove `key=value` from the list of unprocessed options. `TeX`'s argument parser will help us here:

```
28 \def\Tot@tempopt##1=##2=##3\Tot@tempopt{%
```

`\Tot@tempopt` should be called with an option string followed by `==\Tot@tempopt`. `##1` will always contain the keyword, `##2` and `##3` will vary:

option string	##1	##2	##3
<code>key=value</code>	<code>key</code>	<code>value</code>	<code>=</code>
<code>key=</code>	<code>key</code>	(empty)	<code>=</code>
<code>key</code>	<code>key</code>	(empty)	(empty)

Now we can check the existence of the key...

```
29 \@ifundefined{KV@#1@##1}%
30 {}%
31 {%
```

... and, if found, add the complete string to the list of options to be parsed here and remove it from the list of unused options:

```
32 \edef\Tot@tempa{\Tot@tempa,\CurrentOption,}%
33 \expandtwoargs\@removeelement\CurrentOption
34 \@unusedoptionlist\@unusedoptionlist
35 }%
36 }%
```

Add any global options that are known to KV to the start of the list being built in `\Tot@tempa` and mark them used by removing them from the unused option list.

```
37 \@for\CurrentOption:=\@classoptionslist\do{%
38 \expandafter\Tot@tempopt\CurrentOption==\Tot@tempopt
39 }%
```

Now stick the package options at the end of the list and wrap in a call to `\setkeys`.

```
40 \edef\Tot@tempa{%
41 \noexpand\setkeys{#1}{%
42 \Tot@tempa\@optionlist{\@currname.\@current}}%
43 }%
44 }%
```

Do it. And drop unused macros

```
45 \Tot@tempa
46 \let\Tot@no\relax
47 \let\Tot@temp\relax
48 \let\Tot@tempa\relax
49 \let\Tot@tempopt\relax
50 \let\Tot@optionsWithKV\undefined
51 }
```

```
52 \Tot@optionsWithKV{Tot}%
```

Unknown package options will be flagged as an error by `\setkeys`, so we will turn off L^AT_EX's message. (Don't ask me why this has to be wrapped in `\AtEndOfPackage`. If it isn't, the warning will still appear.)

```
53 \AtEndOfPackage{\let\unprocessedoptions\relax}
```

Finally, we check if the value given for `pagespersheet` makes sense. This code is not really perfect since using really strange things for the option value will cause havoc.

```
54 \ifPagesPerSheet
55   \ifnum 1 > \TotPerSheet
56     \PackageError{totpages}{%
57       The number of pages per sheet must be positive.}%
58     You cannot print less than one TeX page per sheet of paper.\MessageBreak
59     The option pagespersheet has been disabled.}
60   \PagesPerSheetfalse
61 \fi
62 \fi
```

`\theTotPages` If the counter is to go to the *dvi* file, we have to use a fixed count register: `\count1`. (For this to work, we have to copy a few internals from *latex.ltx*.) Otherwise, we'll just use a conventional L^AT_EX counter.

```
63 \ifTotPagesToDvi
64   \countdef\c@TotPages=1 \c@TotPages=0
65   \let\c1@TotPages\@empty
66   \newcommand\theTotPages{\arabic{TotPages}}
67 \else
68   \newcounter{TotPages}
69 \fi
```

To save a bit of space, some control sequences for the options are removed. Some more will be removed at `\begin{document}`.

```
70 \let\ifTotPagesToDvi\undefined
71 \let\TotPagesToDvitruel\undefined
72 \let\TotPagesToDvifalse\undefined
```

When shipout occurs, we step the counter.

```
73 \EveryShipout{\stepcounter{TotPages}}
```

Nothing outside this package should mess with that counter, of course.—At the beginning of the document, we should be able to find out what packages are used and warn about some we aren't sure we can cooperate with:

```
74 \AtBeginDocument{%
75   \@ifpackageloaded{count1to}{%
76     \PackageError{totpages}{%
77       Can't use both, count1to and totpages}{%
78       You should read the documentation of these packages and\MessageBreak
79       decide which one is more appropriate for your intentions.}
80   }{}%
81   \@ifpackageloaded{lastpage}{%
82     \PackageError{totpages}{%
83       Can't use both, lastpage and totpages}{%
84       You should read the documentation of these packages and\MessageBreak
85       decide which one is more appropriate for your intentions.}
86   }{}%
```

```

\TotPages@putlabel We define a command which will be executed at the end of the document, but if
anybody did it already, we simply hope they did it right.
87 \ifx\undefined\TotPages@putlabel
If the package hyperref is used, we think we know what to do to produce a correct
hyperlink label. (This code was obtained from the lastpage support of hyperref
itself.)
88 \@ifpackageloaded{hyperref}{%
There are two different ways to produce a warning with hyperref:
89 \ifx\undefined\Hy@WarningNoLine
90 \let\Hy@WarningNoLine\hyper@warn
91 \fi
92 \ifHy@pageanchor
93 \else
94 \Hy@WarningNoLine{%
95 The \string\ref{TotPages} link will not provide hyperlinks\MessageBreak
96 with disabled option 'pageanchor'%
97 }%
98 \fi
When \nofiles was requested, we can't do much since no aux is to be written,
so we won't define anything.
99 \if@files
100 \def\TotPages@putlabel{%
101 \addtocounter{page}{-1}%
102 \begingroup
103 \let\@number\@firstofone
104 \ifHy@pageanchor
105 \ifHy@hypertexnames
106 \ifHy@plainpages
107 \def\Hy@temp{\arabic{page}}%
108 \else
109 \let\textlatin\@firstofone
110 \edef\Hy@temp{\thepage}%
111 \fi
112 \else
113 \def\Hy@temp{\the\Hy@pagecounter}%
114 \fi
115 \fi
116 \immediate\write\@mainaux{%
117 \string\newlabel
118 {TotPages}{\theTotPages}{\thepage}{}%
119 \ifHy@pageanchor page.\Hy@temp\fi}{}%
120 }%
121 \endgroup
122 \addtocounter{page}{1}%
123 }%
124 \fi
Otherwise, we just put a label here, but \label may refer to some random counter
just stepped some time ago, so we have to produce it ourselves.
125 }{%
126 \if@files
127 \def\TotPages@putlabel{%

```

```

128     \addtocounter{page}{-1}%
129     \immediate\write\@mainaux{%
130         \string\newlabel{TotPages}{\theTotPages}{\thepage}}%
131     }%
132     \addtocounter{page}{1}%
133 }%
134 \fi
135 }%
136 \fi

```

`\TotSheets` When `pagespersheet` is disabled, any use of `\TotSheets` will be flagged as an error. Otherwise, the initial value will be 0000. When the value computed during `\TotPages@putSheets` the previous run has been read from the *aux* file, we just assume it is right and check at the end of the document if it is still the same.

```

137 \ifPagesPerSheet
138 \ifx\undefined\TotSheets
139 \newcommand{\TotSheets}{0000}
140 \fi
141 \def\TotPages@putSheets{%
142     \addtocounter{TotPages}{\TotPerSheet}
143     \addtocounter{TotPages}{-1}
144     \divide \value{TotPages} by \TotPerSheet
145     \edef\Tot@temp{\arabic{TotPages}}
146     \ifx\Tot@temp\TotSheets \else
147         \PackageWarning{totpages}{%
148             The number of sheets may have changed (\TotSheets => \Tot@temp).\MessageBreak
149             You should run your document through the formatter again.}
150     \fi
151     \if@filesw
152         \immediate\write\@mainaux{%
153             \string\gdef\string\TotSheets{\Tot@temp}}%
154     \fi
155 }%
156 \else
157 \let\TotSheets\undefined
158 \fi

```

Before we write the value to the *aux* file we call a `\clearpage` to force all pending floats to be output. To do this we have to be sure to be the *last* macro called by `\AtEndDocument`—but this is nearly impossible to assure, so we try to do our very best and set our entry for `\AtEndDocument` during `\begin{document}` processing. When `\nofiles` is specified, though, we won't have to do anything at the end of the document.

```

159 \if@filesw
160 \AtEndDocument{%

```

The label `TotPages` is put onto the last page of the document and holds the total number of pages of the document, i. e., the value of `TotPages`.

```

161     \clearpage
162     \TotPages@putlabel
163 }%
164 \fi

```

Finally, we may have to compute the number of sheets. This is done even when no files are written to allow checking for changes.

```

165 \ifPagesPerSheet
166   \AtEndDocument{\TotPages@putSheets}
167 \fi
168 \let\ifPagesPerSheet\undefined
169 \let\PagesPerSheettrue\undefined
170 \let\PagesPerSheetfalse\undefined
171 }
172 </package>

```

7 An example file

And, at last, there is a small example file using features of package `totpages`.

```

173 (*example)
174 \documentclass[pagespersheet=2]{article}
175 \usepackage[dvips]{hyperref}
176 %%%% Replace preceding line by the following one for pdflatex:
177 %%\usepackage[pdftex]{hyperref}
178 \usepackage[dvi]{totpages}
179 \renewcommand{\theTotPages}{\roman{TotPages}}
180 \begin{document}
181 \setcounter{page}{12345}
182 This is the number of pages (in Roman numerals): ‘‘\ref{TotPages}’’.
183 On the last page, the number ‘‘\pageref{TotPages}’’ is printed.
184
185 \clearpage
186
187 a second page\ldots
188
189 \clearpage
190
191 and a last one\ldots\ldots
192
193
194 When printed on a duplex printer in duplex mode, \TotSheets{} sheets of
195 paper will be needed.
196 \end{document}
197 </example>

```

Change History

v1.00	General: Initial Version 1234	of sheets needed; changed option processing to use keyval;
v1.10	General: Fixed hook into hyperref 1234	corrected a problem when we wanted to write to <code>aux</code> although
v2.00	General: Incorporated changes by Ido Tal to compute number	<code>\nofiles</code> was requested; moved <code>\StopEventually</code> to correct place 1234

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	N
<code>\@classoptionslist</code>	<code>\NeedsTeXFormat</code> 4
<code>\@expandtwoargs</code>	<code>\newcommand</code> 9, 66, 139
<code>\@for</code>	<code>\newcounter</code> 68
<code>\@ifpackageloaded</code> 75, 81, 88	<code>\newif</code> 7, 8
<code>\@ifundefined</code>	<code>\newlabel</code> 117, 130
<code>\@mainaux</code> 116, 129, 152	<code>nodvi</code> <u>9</u> , 1235
<code>\@ptionlist</code>	<code>nopagespersheet</code> <u>11</u> , 1235
<code>\@removeelement</code>	
<code>\@tempc</code>	P
<code>\@unprocessedoptions</code>	<code>\PackageError</code> 56, 76, 82
<code>\@unusedoptionlist</code>	<code>\PackageWarning</code> 147
	<code>pagespersheet</code> <u>11</u> , 1235
A	<code>\PagesPerSheetfalse</code> 8, 12, 60, 170
<code>\AtBeginDocument</code>	<code>\PagesPerSheettrue</code> 21, 169
<code>\AtEndDocument</code> 160, 166	<code>\ProvidesPackage</code> 2
<code>\AtEndOfPackage</code>	
	R
C	<code>\RequirePackage</code> 5, 6
<code>\c@TotPages</code>	
<code>\cl@TotPages</code>	S
<code>\clearpage</code> 161, 185, 189	<code>\setkeys</code> 41
<code>\countdef</code>	<code>\stepcounter</code> 73
<code>\CurrentOption</code> 32, 33, 37, 38	
	T
D	<code>\theTotPages</code> <u>63</u> , 118, 130, 179, 1235
<code>\define@key</code> 10–13	<code>\Tot@no</code> 17, 18, 46
<code>dvi</code> <u>9</u> , 1235	<code>\Tot@optionsWithKV</code> <u>25</u> , 52
	<code>\Tot@temp</code>
E 14, 15, 18, 47, 145, 146, 148, 153
<code>\EveryShipout</code>	<code>\Tot@tempa</code> 27, 32, 40, 42, 45, 48
<code>\expandafter</code>	<code>\Tot@tempopt</code> <u>28</u> , 38, 49
	<code>TotPages</code> <u>62</u> , 1235, 1241
H	<code>\TotPages@putlabel</code> <u>87</u> , 162
<code>\Hy@WarningNoLine</code> 89, 90, 94	<code>\TotPages@putSheets</code> <u>137</u> , 166
<code>\hyper@warn</code>	<code>\TotPagesToDvifalse</code> 11, 72
	<code>\TotPagesToDvittrue</code> 7, 10, 71
I	<code>\TotPerSheet</code> <u>8</u> , 20, 55, 142, 144, 1236
<code>\ifnum</code>	<code>\TotSheets</code> <u>137</u> , 194, 1236
<code>\ifPagesPerSheet</code> <u>8</u> , 54, 137, 165, 168	
<code>\ifTotPagesToDvi</code> <u>7</u> , 63, 70	W
	<code>\write</code> 116, 129, 152