

# The **parselines**\* package

a simple line parser for TeX (FC)

<florent.chervet at free.fr>

2010/10/23 – version 1.1

## Abstract

**parselines** provides an environment **parse lines** which applies a macro to each line of the input between `\begin{parse lines}` and `\end{parse lines}`.

There is also a macro: `\dofilebyline`: its purpose is to expand a user-defined macro `\do` with one parameter (the line) on each line of a given text file.

This packages requires  $\epsilon$ -TeX and no other package.

## Contents

<b>1</b>	<b>User interface</b>	<b>1</b>	<b>2.4 \dofilebyline</b>	<b>3</b>
1.1	The parse lines environment	1	<b>3 History</b>	<b>4</b>
1.2	\dofilebyline	2	[2010/10/23 v1.1]	4
<b>2</b>	<b>Implementation</b>	<b>2</b>	[2010/06/20 v1.0]	4
2.1	Identification	2	<b>4 References</b>	<b>4</b>
2.2	Collecting stuff inside an environment	2	<b>5 Index</b>	<b>4</b>
2.3	The parse lines environment	3		

## 1 User interface

### 1.1 The parse lines environment

```
\begin{<parse lines>}[<initialisation>]{<replacement text of a macro with 1 parameter>}
\end{<parse lines>}
```

Example (with xcolor) – inside the environment `\row` expands to the current row number (stored in `\count@`):

```
\definecolorseries{serie}{rgb}{last}{Goldenrod}{DarkSeaGreen}
\resetcolorseries[3]{serie}
\begin{parse lines}{\color{serie!+}{‘#1’}\par}
  This is a nice little package
  and I really don’t know
  What to do with it !
  May be tomorrow...
\end{parse lines}
```

“This is a nice little package”  
“and I really don’t know”  
“What to do with it !”  
“May be tomorrow...”

```
\begin{parse lines}{\row: #1\par}
  This is a nice little package
  and I really don’t know
  What to do with it !
  May be tomorrow...
\end{parse lines}
```

1: This is a nice little package  
2: and I really don’t know  
3: What to do with it !  
4: May be tomorrow...

This documentation is produced with the DocStrip utility.

→ To get the documentation, run (thrice): `pdflatex parselines.dtx`  
To get the index, run: `makeindex -s gind.ist parselines.idx`  
→ To get the package, run: `etex parselines.dtx`

The .dtx file is embedded into this pdf file thank to embedfile by H. Oberdiek.

## 1.2 `\dofilebyline`

`\dofilebyline{<file-name>}`  
`\dofilebyline*{<file-name>}`

Expands the macro `\do` on each line of a given file.

Each line is read using the TeX primitive `\read`. By default, `\endlinechar` is suppressed (`\endlinechar=-1`) in order not to add a space at the end of each line read. If you do not want this behaviour, use the `*` star form.

If the file does not exist, an error is displayed.

Example:

---

```
\def\do#1{\ifx\par#1\else \addto@hook\mytoken{#1}\fi}%
\dofilebyline{file name.tex}
```

---

In this example: `\ifx\par#1\else` will evaluate to false if `#1` is empty. Therefore, empty lines and double empty lines (equivalent to `\par`) are not captured into the token `\mytoken`.

## 2 Implementation

### 2.1 Identification

The package namespace is `parselin@`.

```
1 <*package>
2 \NeedsTeXFormat{LaTeX2e}% LaTeX 2.09 can't be used (nor non-LaTeX)
3   [2005/12/01]% LaTeX must be 2005/12/01 or younger
4 \ProvidesPackage{parselines}
5   [2010/08/13 v1.0 - a simple line parser for TeX]
```

### 2.2 Collecting stuff inside an environment

```
6 \ifdefined\globtoks \globtoks\parselin@tk
7 \else \newtoks\parselin@tk
8 \fi
9 \newcommand\CollectEnvir{}
10 \protected\def\CollectEnvir#1{%
11   \gdef\parselin@Finish{#1%
12     \expandafter{\the\parselin@tk}%
13     \global\let\parselin@Finish@undefined}%
14   \ifx\parselin@tk#1\else
15     \expandafter\def\expandafter\parselin@Finish
16       \expandafter{\parselin@Finish\parselin@tk{}}%
17     \if \relax
18       \expandafter\expandafter\expandafter\parselin@isatoken
19       \expandafter\meaning\expandafter#1\string\toks
20       \relax
21       \expandafter\gdef\expandafter\parselin@Finish
22         \expandafter{\expandafter\expandafter
23           \expandafter\def\expandafter\expandafter
24             \parselin@Finish}%
25     \fi\fi
26   \parselin@tk{}\def\parselin@stack{b}\parselin@Collect@Body
27 }% \CollectEnvir
28 \long\def\parselin@Collect@Body#1\end#2{%
29   \edef\parselin@stack{\parselin@PushBegins#1\begin\end\expandafter\@gobble\parselin@stack}%
30   \ifx\parselin@stack\@empty
```

```

31     \global\parselin@tk\expandafter{\the\parselin@tk#1}%
32     \aftergroup\parselin@Finish
33     \end{#2}%
34 \else
35     \parselin@tk\expandafter{\the\parselin@tk#1\end#2}%
36     \expandafter\parselin@Collect@Body % recurse
37 \fi
38}% \parselin@Collect@Body
39\long\def\parselin@PushBegins#1\begin#2{\ifx\end#2\else b\expandafter\parselin@PushBegins\fi}
40\expandafter\def\expandafter\parselin@isatoken
41 \expandafter#\expandafter1\string\toks#2\relax{\detokenize{#2}\relax}%

```

## 2.3 The parse lines environment

```

42\newenvironment{parse lines}[2][]
43{%
44    #1%
45    \count@\z@\def\row{\number\count@}%
46    \def\parselin@ProcessLine##1{\advance\count@\@ne #2}%
47    \parselin@endlinechar{%
48        \def~##1~{%
49            \ifx\parselin@@@@@##1\relax
50            \else \parselin@ProcessLine{##1}\expandafter~%
51            \fi}%
52    }%
53    \catcode\endlinechar\active
54    \CollectEnvir\parselin@tk
55 }
56{%
57    \parselin@endlinechar{%
58        \parselin@tk\expandafter{\the\parselin@tk\parselin@@@@@~}%
59    }\the\parselin@tk
60}%
61\long\def\parselin@endlinechar#1{\begingroup
62    \lccode'\~\endlinechar \lowercase{\endgroup#1}%
63}% \parselin@endlinechar
64\def\parselin@@@@@{\parselin@@@@@}

```

## 2.4 \dofilebyline

```

65\newread\parselin@read
66\newrobustcmd*\dofilebyline{\let\parselin@Finish\@empty
67    \@ifstar
68        \parselin@dobyline
69        {\edef\parselin@Finish{\endlinechar\the\endlinechar\relax}%
70        \endlinechar\m@ne
71        \parselin@dobyline}%
72}% \dofilebyline
73\newrobustcmd\parselin@dobyline[1]{% #1 = file name
74    \IfFileExists{#1}
75        \parselin@do@byline
76        {\PackageError{parselines}
77            {No file #1 found !
78            \MessageBreak Your command \string\dobyline\space was ignored}\@eha
79        }%
80}% \parselin@dobyline
81\def\parselin@do@byline{%
82    \openin\parselin@read=\@filef@und%
83    \loop
84        \ifeof\parselin@read\else
85            \read\parselin@read to\@tempa
86            \expandafter\do\expandafter{\@tempa}%
87        \repeat
88    \parselin@Finish

```

```
89}% \parselin@do@byline
```

```
90</package>
```

## 3 History

### [2010/10/23 v1.1]

- `\row` counter added
- Better documentation.

### [2010/06/20 v1.0]

- First version.

## 4 References

[1] *The docstrip program*; 2009/09/25 v2.5d; [CTAN:macros/latex/base/](http://CTAN:macros/latex/base/).

## 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; plain numbers refer to the code lines where the entry is used.

Symbols		M	
<code>\@eha</code> .....	78	<code>\meaning</code> .....	19
<code>\@filef@und</code> .....	82	N	
<code>\@undefined</code> .....	13	<code>\newenvironment</code> .....	42
<code>\~</code> .....	62	<code>\newread</code> .....	65
A		<code>\newtoks</code> .....	7
<code>\active</code> .....	53	<code>\number</code> .....	45
<code>\aftergroup</code> .....	32	O	
B		<code>\openin</code> .....	82
<code>\begin</code> .....	<i>1</i>	P	
C		<code>\PackageError</code> .....	76
<code>\catcode</code> .....	53	<code>\parselin@@@@@</code> .....	49, 58, 64
<code>\CollectEnvir</code> .....	9, 10, 27, 54	<code>\parselin@Collect@Body</code> .....	26, 28, 36, 38
<code>\count@</code> .....	45, 46	<code>\parselin@do@byline</code> .....	75, 81, 89
D		<code>\parselin@dobyline</code> .....	68, 71, 73, 80
<code>\detokenize</code> .....	41	<code>\parselin@endlinechar</code> .....	47, 57, 61, 63
<code>\dobyline</code> .....	78	<code>\parselin@Finish</code> .	11, 13, 15, 16, 21, 24, 32, 66, 69, 88
<code>\dofilebyline</code> .....	2, 66, 72	<code>\parselin@isatoken</code> .....	18, 40
E		<code>\parselin@ProcessLine</code> .....	46, 50
<code>\endlinechar</code> .....	53, 62, 69, 70	<code>\parselin@PushBegins</code> .....	29, 39
G		<code>\parselin@read</code> .....	65, 82, 84, 85
<code>\globtoks</code> .....	6	<code>\parselin@stack</code> .....	26, 29, 30
I		<code>\parselin@tk</code> ....	6, 7, 12, 14, 16, 26, 31, 35, 54, 58, 59
<code>\ifeof</code> .....	84	<code>\protected</code> .....	10
<code>\IfFileExists</code> .....	74	R	
L		<code>\read</code> .....	85
<code>\lccode</code> .....	62	<code>\repeat</code> .....	87
<code>\loop</code> .....	83	<code>\row</code> .....	45
<code>\lowercase</code> .....	62	T	
		<code>\toks</code> .....	19, 41