

The **parselines*** package

a simple line parser for TeX

FC

2010/11/20 – version 1.3

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 ε -TeX and no other package.

Contents

| | | | | | |
|----------|---|----------|-------------------------|----------------------------|----------|
| 1 | User interface | 1 | 2.4 | \dofilebyline | 3 |
| 1.1 | The parse lines environment | 1 | 3 | History | 4 |
| 1.2 | \dofilebyline | 2 | [2010/11/20 v1.3] | | 4 |
| 2 | Implementation | 2 | [2010/11/16 v1.2] | | 4 |
| 2.1 | Identification | 2 | [2010/10/23 v1.1] | | 4 |
| 2.2 | Collecting stuff inside an environ- ment | 2 | [2010/06/20 v1.0] | | 4 |
| 2.3 | The parse lines environment | 3 | 4 | Index | 4 |

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 plain TeX register `\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...”

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. Oberdieck.

```

\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...

```

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   [2010/11/20 v1.3 - 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
30 \expandafter\@gobble\parselin@stack}%
31 \ifx\parselin@stack\@empty
32 \global\parselin@tk\expandafter{\the\parselin@tk#1}%
33 \aftergroup\parselin@Finish
34 \end{#2}%
35 \else
36 \parselin@tk\expandafter{\the\parselin@tk#1\end#2}%
37 \expandafter\parselin@Collect@Body % recurse
38 \fi
39}% \parselin@Collect@Body
40\long\def\parselin@PushBegins#1\begin#2{%
41 \ifx\end#2\else b\expandafter\parselin@PushBegins\fi}
42\expandafter\def\expandafter\parselin@isatoken
43 \expandafter#\expandafter1\string\toks#2\relax{\detokenize{#2}\relax}%

```

2.3 The parse lines environment

```

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

```

2.4 \dofilebyline

```

67\newread\parselin@read
68\protected\def\dofilebyline{\let\parselin@Finish\@empty
69 \@ifstar
70 \parselin@dobyline
71 {\edef\parselin@Finish{\endlinechar\the\endlinechar\relax}%
72 \endlinechar\m@ne
73 \parselin@dobyline}%
74}% \dofilebyline
75\protected\def\parselin@dobyline#1{% #1 = file name
76 \IfFileExists{#1}
77 \parselin@do@byline
78 {\PackageError{parselines}
79 {No file #1 found !
80 \MessageBreak Your command \string\dobyline\space was ignored}\@eha
81 }%
82}% \parselin@dobyline
83\def\parselin@do@byline{%
84 \openin\parselin@read=\@filef@und%

```

```

85 \loop
86   \ifeof\parselin@read\else
87     \read\parselin@read to\@tempa
88     \expandafter\do\expandafter{\@tempa}%
89 \repeat
90 \parselin@Finish
91}% \parselin@do@byline
92</package>

```

3 History

[2010/11/20 v1.3]

- Replacement of `\newrobustcmd` by `\protected\def`: [parselines](#) does not require etoolbox...

[2010/11/16 v1.2]

- Fix in CTAN archive (`.ins` file).

[2010/10/23 v1.1]

- `\row counter` added
- Improved documentation.

[2010/06/20 v1.0]

- First version.

4 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 | | | |
|----------------------------------|----------------|---|--|
| <code>\@eha</code> | 80 | <code>\IfFileExists</code> | 76 |
| <code>\@filef@und</code> | 84 | L | |
| <code>\@undefined</code> | 13 | <code>\lccode</code> | 64 |
| <code>\~</code> | 64 | <code>\loop</code> | 85 |
| | | <code>\lowercase</code> | 64 |
| A | | M | |
| <code>\active</code> | 55 | <code>\meaning</code> | 19 |
| <code>\aftergroup</code> | 33 | N | |
| B | | <code>\newenvironment</code> | 44 |
| <code>\begin</code> | 1 | <code>\newread</code> | 67 |
| C | | <code>\newtoks</code> | 7 |
| <code>\catcode</code> | 55 | <code>\number</code> | 47 |
| <code>\CollectEnvir</code> | 9, 10, 27, 56 | O | |
| <code>\count@</code> | 47, 48 | <code>\openin</code> | 84 |
| D | | P | |
| <code>\detokenize</code> | 43 | <code>\PackageError</code> | 78 |
| <code>\dobyline</code> | 80 | <code>\parselin@@@@</code> | 51, 60, 66 |
| <code>\dofilebyline</code> | 2, 68, 74 | <code>\parselin@Collect@Body</code> | 26, 28, 37, 39 |
| E | | <code>\parselin@do@byline</code> | 77, 83, 91 |
| <code>\endlinechar</code> | 55, 64, 71, 72 | <code>\parselin@dobyline</code> | 70, 73, 75, 82 |
| G | | <code>\parselin@endlinechar</code> | 49, 59, 63, 65 |
| <code>\globtoks</code> | 6 | <code>\parselin@Finish</code> | |
| I | | | 11, 13, 15, 16, 21, 24, 33, 68, 71, 90 |
| <code>\ifeof</code> | 86 | <code>\parselin@isatoken</code> | 18, 42 |
| | | <code>\parselin@ProcessLine</code> | 48, 52 |

| | | | |
|--|----------------|----------------------------|--------|
| <code>\parselin@PushBegins</code> | 29, 40, 41 | R | |
| <code>\parselin@read</code> | 67, 84, 86, 87 | <code>\read</code> | 87 |
| <code>\parselin@stack</code> | 26, 29, 30, 31 | <code>\repeat</code> | 89 |
| <code>\parselin@tk</code> 6, 7, 12, 14, 16, 26, 32, 36, 56, 60, 61 | | <code>\row</code> | 47 |
| <code>\protected</code> | 10, 68, 75 | T | |
| | | <code>\toks</code> | 19, 43 |