Enhancing `write` register in LaTeX documents

C. V. RADHAKRISHNAN

KEYWORDS: PDF, pdfTeX, LaTeX

QUICK LINKS

1. Prolog

The `io` registers in TeX are often a matter of severe limitation. When we try to typeset a document that loads a lot of packages, it becomes a non-trivial task to satisfy register requirements of various packages that compete with each other. It is further complicated, if refining the typeset copy involves usage of more custom packages which want to write out a few streams for reading during subsequent TeX compilations. LaTeX needs a few registers by default, some of the packages might need a few more streams thereby reducing the available number of registers to less than ten. At times, we have been forced to comment out a few `write` registers which are seemingly unused for the document in question to make available necessary registers for our custom packages. But, this is not an elegant way, also, it requires developer intervention during the production process. The package namely, `rvwrite.sty`, is written to solve this nagging problem.

2. The theory

We make use of only one register to write out all our streams, meaning, all the items of information will be written to one file with necessary mark-up and delimiters to separate at a latter stage. The final separation and write out process depend on the literate programming method developed by Eitan Gurari and as provided in his package, `ProTex.sty` which is part of any standard TeX distribution. The written out file is further processed by TeX with the help of `ProTex.sty` to segregate all information and collect in separate files for our final usage.

3. The implementation

The process involves the following stages:

1. Mark-up all strings to be written out with special tags.
2. Run LaTeX in the usual way which will generate `\jobname-write.tex` along with other auxiliary files.
3. Typeset `\jobname-write.tex` with LaTeX, this will result in several output files as you would expect.

3.1. New functions

`\newrvwrite` Every new stream has to be initialized with this function. eg.,

```
\newrvwrite{mex}
```

This will facilitate a new auxiliary stream, `\jobname-mex.tex`.

`\rvwrite` Macro to mark-up strings meant for writing out. This has two arguments, first being the file type and the second the strings. eg.,

```
\rvwrite{mex}{$\beta$}
```

will write `$\beta$` to `\jobname-mex.tex` without expanding any of the strings even if control sequence.

`\ervwrite` Macro to mark-up strings that contain TeX control sequences which need to be expanded before writing. eg.,

```
\ervwrite{mex}{Section: \thesection}
```

will write the contents after expanding `\thesection` to the corresponding counter number.

3.2. The usage

Package can be loaded with the following command:

```
\usepackage{rvwrite}
```

There are no extra options for the package, nor is there any specific sequence for loading, it can be loaded anywhere in the preamble.

As explained in the previous section, to define a new output stream, `\newrvwrite` shall be used, the usage is provided below:

```
\newrvwrite{info}
```

This command will finally result in a file, `\jobname-info.tex` which may be used as such or renamed to suit user's requirements. Each stream may be initialized with `\newrvwrite` macros to facilitate safe writing of output at the end of the process. Even if you miss to initialize, the first instance of writing will trigger initialization, but it will be more safer to initialize the stream in the beginning.

3.3. Examples

3.3.1. Source file

Given below is the source of a test file, `test.tex` for favor of your testing:

```
\documentclass[a4paper]{article}
 \usepackage{pxfonts}
 \usepackage{rvwrite}
 \usepackage{lipsum}

\begin{document}

 \lipsum[1]
 \newrvwrite{info}
 \ervwrite{info}{\dlrchr\string\gamma\dlrchr}
 \rvwrite{info}{$\beta$}
 \ervwrite{info}{JOB: \jobname-\thepage}

 \newrvwrite{tbl}
 \ervwrite{tbl}{\string\def\string\thetable{9}}
 \rvwrite{tbl}{{\itshape Indian \TeX\ Users Group}}
 \rvwrite{tbl}{\LaTeX is a macro package written in \TeX\ language}
 \lipsum[2]

\end{document}
```

3.3.2. Composite stream

The verbatim source of the composite stream written to `test-write.tex` is given below:

```
\documentclass{article}
 \input ProTex.sty
 \AlProTex{tex,<<<>>>,list,title,|,[]}
 \begin{document}

 \<test-info\><<<
 $\gamma$
 >>>
 \<test-info\><<<
 $\beta $
 >>>
 \<test-info\><<<
 JOB: test-1
 >>>
 \<test-tbl\><<<
 \def\thetable{9}
 >>>
 \<test-tbl\><<<
 {\itshape Indian \TeX \ Users Group}
 >>>
 \<test-tbl\><<<
 \LaTeX is a macro package written in \TeX \ language
 >>>
\OutputCode\<test-info\>
\OutputCode\<test-tbl\>

\end{document}
```

3.3.3. Various output files

Two files will be created (1) `test-info.tex` and (2) `test-tbl.tex`, the sources are provided below:

test-info.tex

```
$\gamma$
$\beta $
JOB: test-1
```

test-tbl.tex

```
\def\thetable{9}
{\itshape Indian \TeX \ Users Group}
\LaTeX is a macro package written in \TeX \ language
```

4. Download

The package, documentation and test files can be downloaded from the URL:

`http://download.river-valley.com/cvr/rvwrite.tar.gz`

Bugs, suggestions and feature requests may be posted at http://www.cvr.cc.