runtexshebang: Lua script running LaTeX document files with TeX-style shebang

Lua script running LaTeX document files with a TeX-style shebang (`%#!`)

What is a TeX-style shebang (`%#!`)

In short, a TeX-style shebang (`%#!`) is a special kind of TeX comment that you include in your TeX/LaTeX document file to tell the operating system's shell how to run the file for the rest of the file:

```
%#!lualatex foo.tex
\documentclass{article}
\begin{document}
Hello, {\LaTeX} World!

Happy {\TeX}ing.
\end{document}
```

If you are using a TeX-style shebang, it must appear on the line that matched 20 lines or less in your LaTeX document, and it has to start with a TeX comment symbol (`%`) followed by a hash sign (`#`) and an exclamation mark (`!`), colloquially known as the bang, hence the name shebang for TeX/LaTeX.

Getting started

1. Install `runtexshebang.lua` in your TeX Live system.

The `runtexshebang` package is contained in TeX Live 2023 or higher. So, it can be installed through

```
$ sudo tlmgr install runtexshebang
```

If you manually install `runtexshebang.lua` in your TeX Live system, you can install it through

```
cp runtexshebang.lua
 ↪  /some/where/TEXMFDIST_or_TEXMFLOCAL/scripts/runtexshebang/runtexshebang.lua

cd TEXLIVE_BIN_DIRECTORY
ln -s ../../TEXMFDIST_or_TEXMFLOCAL/scripts/runtexshebang/runtexshebang.lua runtexshebang
```

2. Make a sample file with a TeX-style shebang.

Make the following LaTeX document.

```
%#!lualatex foo.tex
\documentclass{article}
\begin{document}
Hello, {\LaTeX} World!

Happy {\TeX}ing.
\end{document}
```

3. Run the sample file.

`runtexshebang foo.tex`

Then, it will run `lualatex foo.tex`. To see which command would be run without running it, use `runtexshebang --dry-run foo.tex`.

Examples

TeXworks

Settings

1. Open "Typesetting" tab in TeXworks Preferences.
2. Add the tool configuration "runtexshebang" in "Processing tools" as below.
   - Name: `runtexshebang`
   - Program: `runtexshebang`
   - Arguments: `$fullname`
   - Check "View PDF after running" (if necessary)

LaTeX document in the internal editor of TeXwork

`% !TEX program =` is a magic comment of TeXworks. You can set one processing tool as `% !TEX program = <your choice of tool configuration>`.

```
% !TEX program = runtexshebang
%#! lualatex --synctex=1 foo
```

LaTeX Workshop: Visual Studio Code

Settings

Nothing!

LaTeX document in Visual Studio Code

`% !TEX program =` is a magic comment of LaTeX Workshop. You need another magic comment to the target LaTeX document file as below.

```
% !TEX program = runtexshebang
% !TEX options = "%DOC%".tex
%#! lualatex --synctex=1 foo
```

TeXShop

Settings

There the "Engine" settings for the command lines "TeX+dvipdfmx / TeX+dvips+distiller" as follows.

- TeX: `runtexshebang`
- LaTeX: `runtexshebang`

LaTeX document in the internal editor of TeXShop

`% !TEX program =` is a magic comment of TeXShop.

```
% !TEX program = runtexshebang
%#! lualatex --synctex=1 foo
```

Applications

What does the PATH variable set in my editor?

```
%#! echo $PATH
```

How about general configuration information for the activated TeX Live version and its configuration in my editor?

```
%#! tlmgr conf
```

Otherwise

You can run any command line on your OSs. For example, you can use any build tools as follows: `make`, `rake`, `llmk`, `latexmk`, and so on.

```
%#! make foo.pdf

% #! rake foo.pdf
% #! llmk
% #! latexmk foo
%% and so on.
```

You can typeset a LaTeX document file in a Docker container as below.

```
%#! docker run -i --rm --workdir /data --mount type=bind,src=$(pwd)/,dst=/data/   bar/foo
 ↪  lualatex foo
```

Security

- **runtexshebang executes the `%#!` line of your document as a shell command line** (`/bin/sh -c` on Unix, `cmd.exe /c` on Windows). Running `runtexshebang foo.tex` is equivalent to typing the contents of its `%#!` line into your terminal yourself. Treat `.tex` files with a `%#!` line exactly like a `Makefile` or a `latexmkrc`: read the first lines before running them, and only use runtexshebang on documents you trust.
- **Editor integration widens the exposure.** Once you have registered runtexshebang as a typesetting tool or engine, every document that carries `% !TEX program = runtexshebang` and a `%#!` line will execute that line when you press "Typeset". Review the `%#!` line of documents you received from others. runtexshebang prints `runtexshebang: running: <command>` on stderr (the editor console) before executing it, and `runtexshebang --dry-run foo.tex` prints the line without executing it.
- **runtexshebang must never be added to `shell_escape_commands` in `texmf.cnf`** (or to any other restricted-shell-escape allow list). If it were, any document compiled with the default restricted shell escape could run arbitrary commands via `\write18{runtexshebang evil.tex}`. It is intentionally absent from that list and is not designed to be called from `\write18`. It is a build tool that the user invokes, like `make`, `latexmk`, `arara` or `llmk`.
- runtexshebang loads no Lua code (no `require`, no `dofile`) and reads no configuration file. Its behaviour depends only on its command-line arguments, the first 20 lines of the given file, and the environment (`PATH`, `COMSPEC`).
- The command is located through your `PATH`. On Windows, `cmd.exe` normally looks in the current directory first, so an executable placed next to the document could shadow a TeX Live program named on the shebang line. runtexshebang sets `NoDefaultCurrentDirectoryInExePath=1` before running the command to switch that lookup off.
- Tampering with `runtexshebang.lua` itself requires write access to the TeX Live tree, the same as for any other program shipped with TeX Live; that is covered by `tlmgr`'s signature verification, not by this script.

- The exit status of the executed command is passed through, so editors and `make` can detect failures. runtexshebang itself exits 1 on usage or file errors and 2 when no `%#!` line is found in the first 20 lines.
- Only lines that start with `%#!` in column 1 are recognised; `% #!` and `%#!` are ignored on purpose, which lets you keep alternative command lines in the file as plain comments. A UTF-8 byte order mark before `%#!` on the first line is ignored, and CRLF line endings are handled on every platform.

References

Commentary

- TeXworks、TeXShop、VSCode で TeX-style shebang してみた - Qiita (in Japanese)

Tools supporting `%#!`

- YaTeX: Yet Another TeX mode for Emacs
- llmk: Light LaTeX Make

Enjoy Happy TeXing!

License

This program is licensed under the conditions of the LaTeX Project Public License, either version 1.3c of this license or any later version.

Munehiro Yamamoto https://github.com/munepi