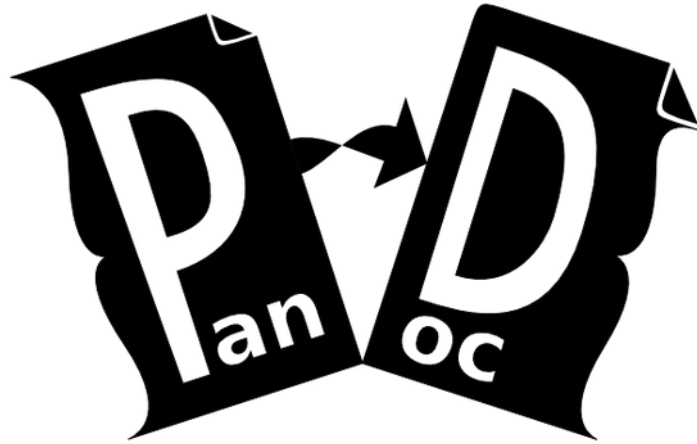


Document conversion with Pandoc



Tom Ryder

tom@sanctum.geek.nz

<https://sanctum.geek.nz/>

Plain-text universe—1/4

- I'm a **plain text** kind of guy.
- I spend a lot of time in a text terminal (xterm):
 - Writing commands in Bash, reading output
 - Reading documentation in `man(1)` pagers
 - Writing code, documents, and email in Vim
- I'm not too good for **WYSIWYG**—but I don't like straying too far from plain text.

text/plain

Plain-text universe—2/4

- But the world works in multiple **document formats**.
- My tech publisher wanted **DOCX** at first.
- Then they changed their minds to HTML.
- Poetry collections usually want **DOCX**.
- Except for the ones that want PDF...

Plain-text universe—3/4

- The formats themselves don't bother me.
 - Well, Microsoft Word does...
- But I don't like **composing** or **editing** documents like that, because it takes me away from my beloved **Vim-in-xterm**.



text/plain

Plain-text universe—4/4

- What I'd really like is to write *everything* in a plain-text format like [Markdown](#)...
- ...and then just **convert it** to whatever document type was demanded of me.

text/plain

Enter Pandoc

- **Pandoc** is the “swiss army knife” of document formatting, like **ffmpeg** is for media.
- Convert from very many formats to very many *other* formats, both **free** and **proprietary**.
- Markdown, **AsciiDoc**...
- RTF, **Microsoft Word** (.doc and .docx)...
- Man pages, **LaTeX**, PDF...



\pagenumbering{gobble}

Yorick

=====

Tom Ryder

The sexton tosses him the skull,
And all is silent. No ghosts here.
No hauntings, blood feuds; no revenge,
No prince called to deathly valour.

He scrapes the soil from an empty eye.
The dead jester, loving as he was,
Joked with the prince, laughed with him,
Consoled him when he wept, as men and children do,
But now is silent. He can speak,

Yorick

Tom Ryder

The sexton tosses him the skull,
And all is silent. No ghosts here.
No hauntings, blood feuds; no revenge,
No prince called to deathly valour.

He scrapes the soil from an empty eye.
The dead jester, loving as he was,
Joked with the prince, laughed with him,
Consoled him when he wept, as men and children do,
But now is silent. He can speak,



[Yorick

Tom Ryder

The sexton tosses him the skull,
And all is silent. No ghosts here.
No hauntings, blood feuds; no revenge,
No prince called to deathly valour.

He scrapes the soil from an empty eye.
The dead jester, loving as he was,
Joked with the prince, laughed with him,
Consoled him when he wept, as men and children do,
But now is silent. He can speak,

Dotfiles (Tom Ryder)

=====

This is my personal repository of configuration files and scripts for `~`, including most of the settings that migrate well between machines.

This repository began as a simple way to share Vim and tmux configuration, but a lot of scripts and shell configuration have been added over time, making it into a personal suite of custom Unix tools.

Installation

```
$ mkdir -p ~/.local/src
$ git clone https://dev.sanctum.geek.nz/code/dotfiles.git ~/.local/src/dotfiles
$ cd ~/.local/src/dotfiles
$ git submodule init
$ git submodule update
$ make
$ make -n install # Check output carefully
$ make install
```

For the default `all` target, you'll need a POSIX-fearing userland, including `make(1)` and `m4(1)`.



index : dotfiles.git

Personal scripts and configuration files

[about](#) [summary](#) [refs](#) [log](#) [tree](#) [commit](#) [diff](#)

Dotfiles (Tom Ryder)

This is my personal repository of configuration files and scripts for `$HOME`, including most of the settings that migrate well between machines.

This repository began as a simple way to share Vim and tmux configuration, but a lot of scripts and shell configuration have been added over time, making it into a personal suite of custom Unix tools.

Installation

```
$ mkdir -p ~/.local/src
$ git clone https://dev.sanctum.geek.nz/code/dotfiles.git ~/.local/src/dotfiles
$ cd ~/.local/src/dotfiles
$ git submodule init
$ git submodule update
$ make
$ make -n install # Check output carefully
$ make install
```

For the default `all` target, you'll need a POSIX-fearing userland, including `make (1)` and `m4 (1)`.

\$

DOTFILES(7)

DOTFILES(7)

Dotfiles (Tom Ryder)

This is my personal repository of configuration files and scripts for ~~\$HOME~~, including most of the settings that migrate well between machines.

This repository began as a simple way to share Vim and tmux configuration, but a lot of scripts and shell configuration have been added over time, making it into a personal suite of custom Unix tools.

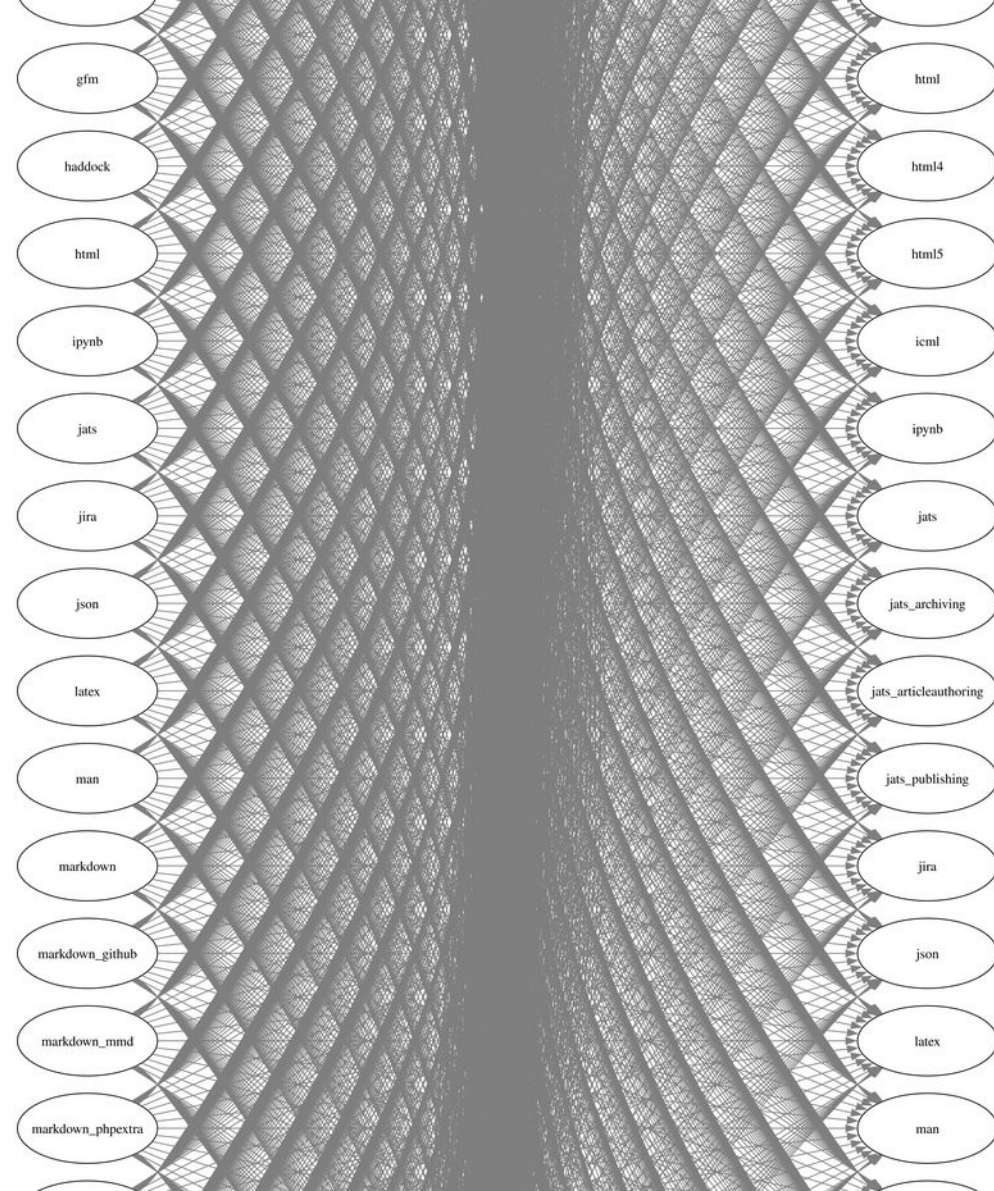
Installation

```
$ mkdir -p ~/.local/src
$ git clone https://dev.sanctum.geek.nz/code/dotfiles.git ~/.local
/src/dotfiles
$ cd ~/.local/src/dotfiles
$ git submodule init
$ git submodule update
$ make
$ make -n install # Check output carefully
$ make install
```

For the default **all** target, you'll need a POSIX-fearing userland, in-

❖

So many **formats**...



Installation

- Implemented in Haskell (!)
- But you don't have to compile it.
- It's packaged pretty much everywhere, so:

```
$ sudo apt install pandoc,  
$ sudo dnf install pandoc...
```
- It's **pretty big**; on Debian GNU/Linux amd64, **165 MiB installed** (!)
- It's worth the download and the disk space!



License

- It's **free software**, of course.
- GNU General Public License v2
- (The Linux license!)



Usage—1/2

```
$ pandoc in.md -o out.docx
```

```
$ pandoc in.md -o out.pdf
```

```
$ pandoc in.md -o out.html
```

```
...
```



Usage—2/2

- By default: reads from stdin, writes to stdout.
- You'll need to specify the input (-f) and output (-t) formats, though.
- Just use a source filename and -o. It's easier.

```
$ awk '...' | pandoc -f plain -t html5 | tidy
```



Version control—1/2

- You can use [Git](#) to track documents produced with WYSIWYG editors.
- You can track commits, authorship, change times, tag releases, clone...it all works.
- However, you can't easily **diff** the documents.
- Git was designed for small, plain-text files.



Version control—2/2

- If your “source” material is plain text, Git becomes that much more useful.
- You can see changes quickly with `git diff`, or any other diff tool.
- You aren’t dependent on the “Track Changes” feature of your word processing software.



Makefiles—1/3

- One of my favourite tricks is using make to publish my work in multiple formats.
- I write a short Makefile calling pandoc with a few formats.
- I make a change to the .md file, and then just type make to update the rendered documents.



Makefiles—2/3

```
.SUFFIXES:  
.SUFFIXES: .md .docx .pdf
```

```
ALL = yorick.docx yorick.pdf  
all: $(ALL)
```

```
.md.docx:  
    pandoc $< --output=$@  
.md.pdf:  
    pandoc $< --output=$@
```

```
clean:  
    rm -f -- $(ALL)
```



Makefiles—3/3

- Add the rendered documents to `.gitignore`.
- Keep the `Makefile` in the Git repository too, same as you would for software!
- Turns out lots of the software source code approach works just fine for writing prose...and even poetry.



When to Pandoc

- **Presenting** to Microsoft-Word-only users
- Document has a simple, hierarchical structure: headings, lists, links...
- Document needs tracking and version comparison (use version control)
- Converting to multiple formats including web is desirable
- You prefer plain text to WYSIWYG



When *not* to Pandoc

- **Collaborating** with Microsoft-Word-only users (use LibreOffice)
- Look-and-feel matters and must be maintained (use design software)
- Branding concerns (use design software)
- Complex or unique typesetting (use LaTeX)
- Fixed templates
- LUG presentations about Pandoc...



Tips—1/3

- Put sentences on new lines in your source, with two newlines for paragraph breaks:

A treatise on quick brown foxes
=====

Tom Ryder

The quick brown fox jumps over the lazy dog.
-Truer words were never spoken.
+In this essay, I will explain these words.
Their impact is felt all over civilization.



Tips—2/3

- Use Markdown or ASCIIDoc for your “source” documents.
 - Converting from e.g. DOCX to PDF totally works, though.
- Make self-contained, complete HTML5 documents:
`--self-contained --metadata title='Title'`



Tips—3/3

- **Keep it simple**, don't fight it!
- If you *need* to get things pixel-perfect, you might want proper typesetting software like [LaTeX](#) instead.



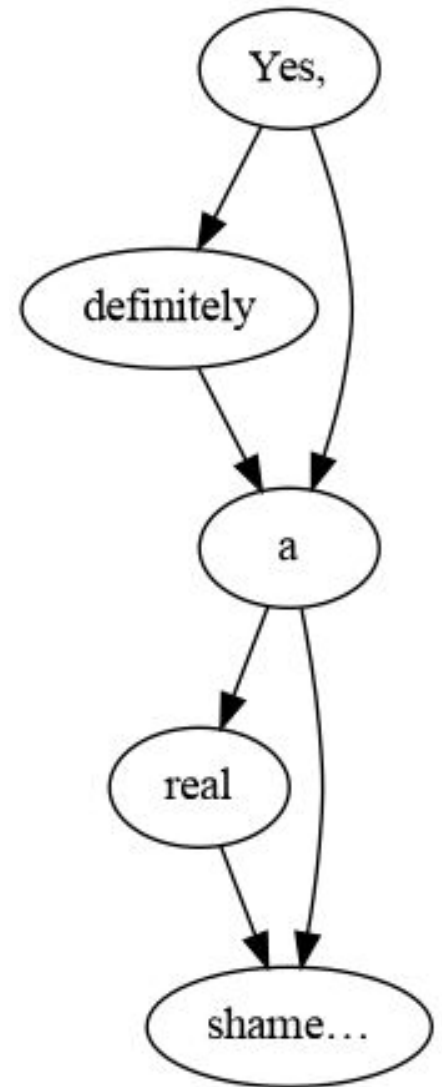
Teaser—1/2

- So now I know I can write anything in plain text and quickly format it as a document.
- More of my world can stay in plain text. Great!
- Shame I can't draw **diagrams** like that.



Teaser—2/2

```
digraph demo {  
    "Yes," -> "definitely"  
    "Yes," -> "a"  
    "definitely" -> "a"  
    "a" -> "real"  
    "a" -> "shame..."  
    "real" -> "shame..."  
}
```



Questions?

- Pandoc
- Markdown
- “GitHub-flavored” Markdown
- AsciiDoc

Email: tom@sanctum.geek.nz

Website: <https://sanctum.geek.nz/>

Fediverse: [@tejr@mastodon.sdf.org](https://mstdn.social/@tejr)

