10 minutes

An Orgmode Note Workflow

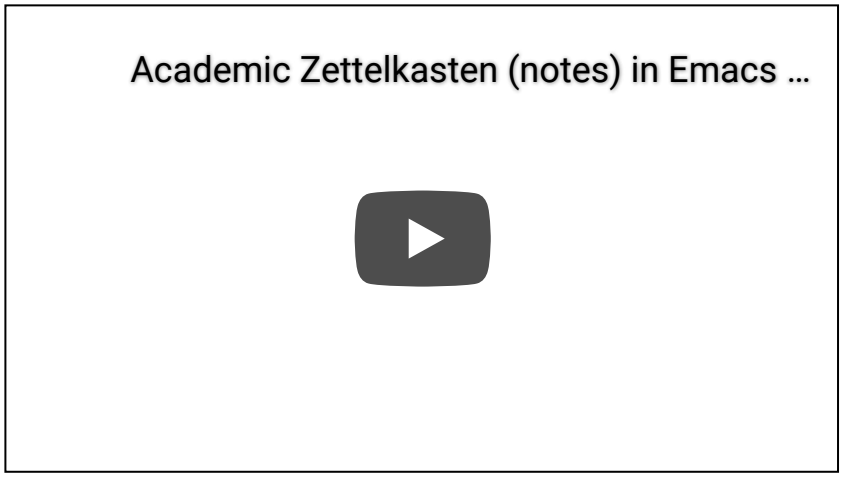
Rohit Goswami

Table of Contents

Background

One of the main reasons to use `orgmode` is definitely to get a better note taking workflow. Closely related to blogging or writing, the ideal note workflow is one which lets you keep a bunch of throwaway ideas and also somehow have access to them in a coherent manner. This will be a long post, and it is a work-in-progress, so, keep that in mind. Since this is mainly me[1] work-shopping my technique, the philosophy will come in a later post probably. This workflow is documented more sparsely in my config file here, in the `noteYoda` section[2]. Some parts of this post also include mini video clips for clarity[3].

The entire workflow will end up being something like this[4]:



Concept

While working through ideas, it actually was more useful to describe the workflow I want, and then implement it, instead of relying on the canned approaches of each package. So the basics of the ideology are listed below.

Reference Management

Reference management is one of the main reasons to consider a plain-text setup, and mine is no different. The options most commonly seen are:

Mendeley
: This is a great option, and the most mobile friendly of the bunch. Sadly, the price tiers aren't very friendly so I have to give it a hard pass.

Jabref
: This is fun, but really more of a per-project management system, but it works well for that. The fact that it is Java based was a major issue for me.

Zotero
: This is what I personally use and recommend. More on that in a later post.

Notes

The idea is to be able to create notes for all kinds of content. Specifically, papers or books, along with webpages. This then requires a separate system for each which is described by:

Search Engine

The search engine is key, both in terms of accessibility and scalability. It is assumed that there will be many notes, and that they will have a wide variety of content. The search interface must then simply allow us to narrow down our candidates in a meaningful manner.

Contextual Representation

This aspect of the workflow deals with representations, which should transcend the usage of tags or categories. In particular, it would be nice to be able to visualize the flow of ideas, each represented by a note.

Backlinks

In particular, by backlinks at this point we are referring to the ability to link to a `pdf` or a website with a unique key such that notes can be added or removed at will.

Storage

Not actually part of the workflow in the same way, since it will be handled at the system level, it is worth nothing, that in this workflow Zotero is used to export a master **bib** file and keeps it updated, while the notes themselves are version controlled[5].

The concepts above will be handled by the following packages.

Concept	Package	Note
Search	deft	Has a great interface
Context	org-roam	Allows the export of graphiz mindmaps
Backlinks	org-roam, org-ref, org-noter	Covers websites, bibliographies, and pdfs respectively

A key component in this workflow is actually facilitated by the fabulous `org-roam-bibtex` or ORB. The basic idea is to ensure meaningful templates which interpolate smoothly with `org-roam`, `org-ref`, `helm-bibtex`, and `org-capture`.

Basic Variables

Given the packages we will be using, some variable settings are in order, namely:

```
(setq
   org_notes (concat (getenv "HOME")
"/Git/Gitlab/Mine/Notes/")
   zot_bib (concat (getenv "HOME") "/GDrive/zotLib.bib")
   org-directory org_notes
```

```
    deft-directory org_notes
```

↓↓

Search

For the search setup, the `doom-emacs` `deft` setup, by adding `+deft` in my `init.el`, worked out of the box for me. For those who do not use `doom` [6], the following should suffice:

```
(use-package deft
  :commands deft
  :init
  (setq deft-default-extension "org"
        ;; de-couples filename and note title:
        deft-use-filename-as-title nil
```

↓↓

For more about the `doom-emacs` defaults, check the Github repo. The other aspect of interacting with the notes is via the `org-roam` interface and will be covered below.

Bibliography

Since I will be using `org-ref`, it makes no sense to load or work with the `+biblio` module at the moment. Thus this section is actually `doom` agnostic. The basic tools of bibliographic management from the `emacs` end are the venerable `helm-bibtex` (repo here) and `org-ref` (repo here). In order to make this guide complete, I will also describe the Zotero settings I have.

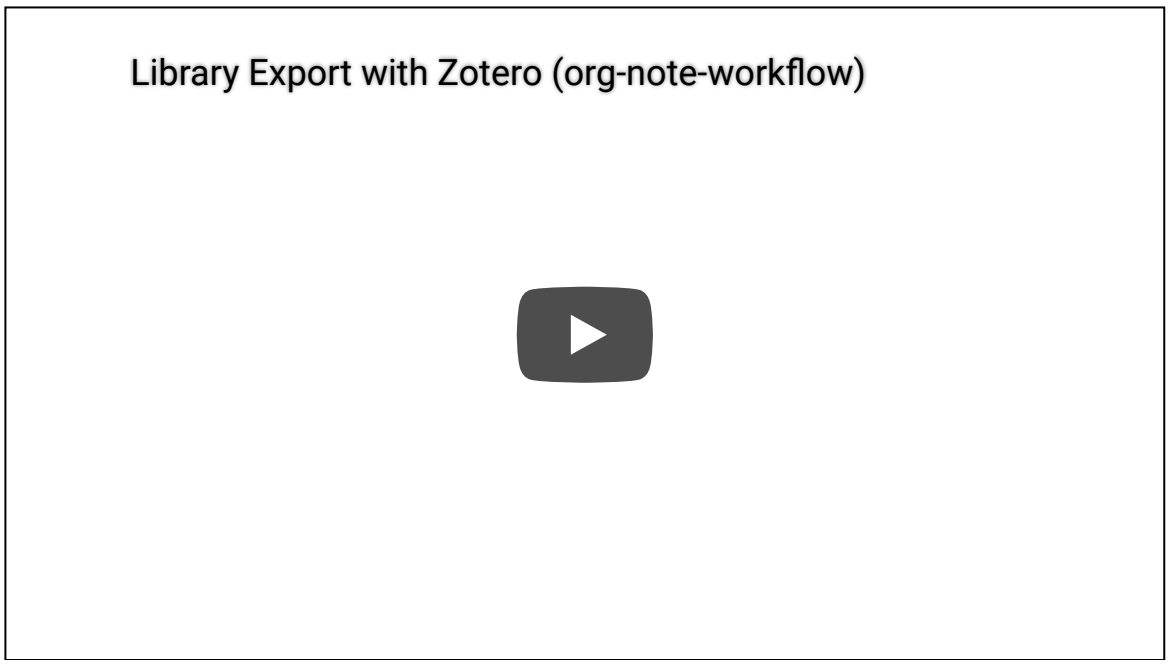
Zotero

Without getting too deep into the weeds here, the basic requirements are:

- Zotero
- The better bibtex extension

The idea is to then have one top level `.bib` file in some handy location which you will set up to sync automatically. To make life easier, there is a tiny recording of

the next steps.



Helm-Bibtex

This venerable package is really good at interfacing with a variety of externally formatted bibliographic managers.

```
(setq
 bibtex-completion-notes-path
"/home/haozeke/Git/Gitlab/Mine/Notes/"
 bibtex-completion-bibliography
"/home/haozeke/GDrive/zotLib.bib"
 bibtex-completion-pdf-field "file"
```

↓↓

`doom-emacs` users like me might want to wrap the above in a nice `after! org-ref` expression, but it doesn't really matter.

Explanation

To break-down aspects of the configuration snippet above:

- The template includes the `orb-process-file-field` function to allow selecting the `pdf` to be used with `org-noter`
- The `file` field is specified to work with the `.bib` file generated by Zotero

- `helm-bibtex` allows for any of the keys in a `.bib` file to be used in a template, and an overly expressive one is more useful

- The `ROAM_KEY` is defined to ensure that cite backlinks work correctly with `org-roam`
- As I prefer to have one notes file per `pdf`, I have only configured the `bibtex-completion-notes-template-multiple-files` variable

Org-Ref

As discussed above, this just makes citations much more meaningful in `orgmode`.

```
(use-package org-ref
    :config
    (setq
         org-ref-completion-library 'org-ref-ivy-cite
         org-ref-get-pdf-filename-function 'org-ref-get-
pdf-filename-helm-bibtex
```

↓↓

An essential aspect of this configuration is just that most of heavy lifting in terms of the notes are palmed off to `helm-bibtex`.

Explanation

To break-down aspects of the configuration snippet above:

- The `org-ref-get-pdf-filename-function` simply uses the `helm-bibtex` settings to find the `pdf`
- The default bibliography and notes directory are set to the same location as all the `org-roam` files, to encourage a flat hierarchy
- The `org-ref-notes-function` simply ensures that, like the `helm-bibtex` settings, I expect one file per `pdf`, and that I would like to use my `org-roam` template instead of the `org-ref` or `helm-bibtex` one

Note that for some reason, the format specifiers for `org-ref` are **not** the keys in `.bib` but are instead the following[7]:

`.bib` but are instead, the following:

```
In the format, the following percent escapes will be
expanded.
%l   The BibTeX label of the citation.
%a   List of author names, see also `reftex-cite-
punctuation'.
%2a  Like %a, but abbreviate more than 2 authors like
Jones et al.
```

↓↓

Indexing Notes

This part of the workflow builds on the concepts best known as the Zettelkasten method. More details about the philosophy behind `org-roam` is here.

Org-Roam

The first part of this interface is essentially just the `doom-emacs` configuration, adapted for those who don't believe in the dark side below.

```
(use-package org-roam
  :hook (org-load . org-roam-mode)
  :commands (org-roam-buffer-toggle-display
             org-roam-find-file
             org-roam-graph
             org-roam-insert
```

↓↓

Once again, for more details, check the Github repo.

Org-Roam-Bibtex

The configuration required is:

```
(use-package org-roam-bibtex
 :after (org-roam)
 :hook (org-roam-mode . org-roam-bibtex-mode)
 :config
 (setq org-roam-bibtex-preformat-keywords
  '("=key=" "title" "url" "file" "author-or-editor"
```

↓↓

Where most of the configuration is essentially the template again. Like `helm-bibtex`, ORB allows taking arbitrary keys from the `.bib` file.

Org Noter

The final aspect of a `pdf` workflow is simply ensuring that every `pdf` is associated with notes. The philosophy of `org-noter` is best described here. Only minor tweaks should be required to get this working with `interleave` as well.

```
(use-package org-noter
  :after (:any org pdf-view)
  :config
  (setq
   ;; The WM can handle splits
   org-noter-notes-window-location 'other-frame
```

↓↓

Evidently, from my configuration, it appears that I decided to use org-noter over the more commonly described interleave because it has better support for working with multiple documents linked to one file.

Org-Protocol

I will only cover the bare minimum relating to the use of `org-capture` here, because eventually I intend to handle a lot more cases with orca. Note that this part of the workflow has more to do with using `org-roam` with websites than `pdf` files.

Templates

This might get complicated but I am only trying to get the bare minimum for `org-protocol` right now.

```
;; Actually start using templates
(after! org-capture
  ;; Firefox and Chrome
  (add-to-list 'org-capture-templates
               '("P" "Protocol" entry ; key, name, type
                 (file+headline +org-capture-notes-file
```

↓↓

Conclusions

At this point, many might argue that since by the end, only one template is called, defining the rest were pointless. They would be right, however, this is just how my configuration evolved. Feel free to cannibalize this for your personal benefit. Eventually I plan to expand this into something with `org-journal` as well, but not right now.

Comments

The older commenting system was implemented with utteranc.es as seen below.

14 Comments - *powered by utteranc.es*

pepegar commented on 12 May 2020

This is an excellent post, thank you very much.

I've trying to use the zettelkasten method lately with org-roam myself, but this post helped me better understand how to integrate with other tools like zotero & ivy-bibtex for references!

Keep up with the good content, cheers!

notuntoward commented on 31 May 2020

Great post.

I don't see it here, but do you also somehow also version control your Zotero database?

In my current system, I have one gigantic org file and one gigantic .bib file, both in git -- and git has actually saved me more often on the .bib file. I've considered using Zotero and then version controlling an exported bib file, but that has problems. If I could "git" Zotero, I'd avoid them.

HaoZeke commented on 6 Jun 2020 Owner

Hi @notuntoward
Excellent point. I don't actually version control my Zotero database, it'd be too large. I do use a web storage provider and the Zotero account service though...
Also, thanks for the kind words @pepegar!

alvarmaciel commented on 10 Jun 2020

Hi!
Excellent post. I'm form Argentina, Spanish speaker. So If you let me, I will make a translation of it. To learn and to make this technology accessible to Spanish speakers

HaoZeke commented on 10 Jun 2020 Owner

Hi @alvarmaciel. Thanks, I'd be happy to let you translate it, as long as you link back to this post! I will add a link to your translated version to this post as well when you are done. :)

👍 1

alvarmaciel commented on 10 Jun 2020

Great! @HaoZeke and of course I will link back to this post.

👍 1 🎉 1

alvarmaciel commented on 17 Jun 2020

it's done, the spanish translation of this article can be found here https://alvarmaciel.gitlab.io/cyberiada/post/20-06-10-una-metodologia-de-toma-de-notas-con-orgmode/

👍 1 🎉 1

japhir commented on 16 Jul 2020

Hey, this is awesome! I currently adapted it to my emacs config, but everyone who's using org-roam using doom makes it more attractive by the day ;-).

One main question though: when you say: "At this point, many might argue that since by the end, only one template is called, defining the rest were pointless.", it isn't entirely obvious to me which of the many configs you end up using mainly! I think it's probably the org-roam-bibtex one, but I'm not sure. Could you elaborate?

And what would the workflow then be for PDF's/references?
something like this?

- find nice article to add to references online
- hit Zotero plugin to add to bibliography.bib (with betterbibtex) and automatically download the pdf
- call capture template from emacs, find the correct citekey, creating a new org-roam file with an org-noter heading in it

👍 1

japhir commented on 16 Jul 2020

Also, I think there are two parts of the code that aren't working when you're not using doom in your use-package lines:

this part in your org-roam config complains about the `+org-roam-open-buffer-maybe-h()` and `+org-roam-open-buffer-on-find-file` not working

```
(add-hook 'find-file-hook
  (defun +org-roam-open-buffer-maybe-h ()
    (and +org-roam-open-buffer-on-find-file
         (memq 'org-roam-buffer--update-mayb
         (not (window-parameter nil 'window
```

```
(not (window-parameter nil 'window-
(not (eq 'visible (org-roam-buffer-
(with-current-buffer (window-buffer
  (org-roam-buffer--get-create)))))
```

and this bit is complaining that the packages don't exist:

```
;; Since the org module lazy loads org-protocol
```

👍 1

japhir commented on 16 Jul 2020

and there's probably an unbalanced parenthesis after the first few lines of the `:config` in the org-roam section, where you start with `(setq ...` but then the new line starts with `(add-hook 'find-file-hook .`

HaoZeke commented on 17 Jul 2020 Owner

Hi @japhir, thanks for the detailed queries and comments! The last step is a little disjointed in that my workflow is:

- Keep a topical notes file
- This also (sometimes) involves setting up a capture template to save web-pages directly to the same file (in a different heading)
- Add citations in that file and then use `orb` to open and link pdf notes

I'm not fully convinced that's the best way, but for the moment I haven't had time to fine tune it further. I'll put out a write up on the full workflow when it's more polished ^_^

Excellent catch regarding the missing packages, I'm not sure about the parenthesis though I'll update that section from the current `doom` upstream repo.

MarkusSagen commented on 8 Aug 2020

I Believe the config for the org-roam section is incorrect...
I got it working by adding a closing parenthesis before the hooks.
Thank you for a great setup and tutorial!

```
(use-package org-roam
```

```
  :hook (org-load . org-roam-mode)
  :commands (org-roam-buffer-toggle-display
             org-roam-find-file
             org-roam-graph
             org-roam-insert
             org-roam-switch-to-buffer
             org-roam-dailies-date
             org-roam-dailies-today
             org-roam-dailies-tomorrow
             org-roam-dailies-yesterday)
  :preface
  ;; Set this to nil so we can later detect whet
  ;; directory for it, and default to `org-direc
  (defvar org-roam-directory nil)
  :init
  :config
  (setq org-roam-directory (expand-file-name (or
                                              org
```

👍 1 🎉 1

HaoZeke commented on 9 Aug 2020 Owner

Thanks for the great catch (and the appreciaton) @MarkusSagen! I'll update it with your suggestion. It's too bad there's no way to automatically sync these code-blocks with my configuration.

tshu-w commented 3 weeks ago

After iterating, I have a similar configuration to yours, except I'm not using org-roam-bibtex and org-noter, but it seems I can configure it to do what org-roam-bibtex describes(https://github.com/tshu-w/.emacs.d/blob/master/lisp/lang-org.el#L855), am I missing something?

1. Rohit Goswami that is, from the landing page; obviously ↩
2. This is a reference to my fantastic pet, named Yoda ↩
3. Recorded with SimpleScreenRecorder, cut with LosslessCut, uploaded to YouTube, and embedded with a Hugo shortcode ↩
4. The video uses `org-ref-notes-function-many-files` as the `org-ref-notes-function` so the template looks a little different ↩
5. For some strange reason a lot of online posts suggested Dropbox for syncing notes, which makes no sense to me, it is always better to have version control

and ignore rules ↩

6. Therefore clearly proving that the cookies of the dark side have no power in the holy text editor war ↩
7. Where these are from the org-ref documentation ↩

#tools #emacs #workflow #orgmode

2097 Words

Posted: 2020-05-10 15:01 +0000