

The org-ref manual

John Kitchen

2015-03-15 Sun

Contents

1	Introduction to org-ref	2
1.1	Basic usage of org-ref	4
1.1.1	Bibliography links	4
1.1.2	Citations	4
1.1.3	Pre/post text support	6
1.1.4	label links	7
1.1.5	ref links	7
1.1.6	Some other links	9
1.1.7	Controlling link messages and tooltips	9
1.2	org-ref customization of helm-bibtex	10
1.3	Some basic org-ref utilities	10
1.4	L ^A T _E X export	11
1.4.1	BibTeX users	11
1.4.2	biblatex users	12
1.5	Other exports	12
2	org-ref-ivy	12
3	Other libraries in org-ref	13
3.1	doi-utils	13
3.1.1	Bibtex key format	14
3.1.2	Troubleshooting doi-utils	14
3.2	org-ref-bibtex	15
3.2.1	Generate new bibtex files with adapted journal names	15
3.2.2	Modifying bibtex entries	15
3.2.3	Bibtex entry navigation	16
3.2.4	Hydra menus for bibtex entries and files	16
3.2.5	Formatted bibtex entry citations	16

3.3	org-ref-wos	17
3.4	org-ref-scopus	17
3.5	org-ref-isbn	18
3.6	org-ref-pubmed	18
3.7	org-ref-arxiv	19
3.8	org-ref-sci-id	19
3.9	x2bib	20
3.10	org-ref-latex	20
3.11	org-ref-pdf	20
3.12	org-ref-url-utils	20
4	Appendix	21
4.1	Customizing org-ref	21
4.2	Customizing how PDFs are opened	21
4.2.1	Using doc-view or pdf-tools	21
4.2.2	A note for Mendeley, JabRef and Zotero users	22
4.3	Setting up notes to work with multiple notes.org files	23
4.4	Customizing the notes format	23
4.5	Other things org-ref supports	24
4.5.1	org-completion	24
4.5.2	Storing org-links to labels	24
4.5.3	Storing links to bibtex entries	24
4.5.4	Indexes	24
4.5.5	Glossaries	25
4.6	Prelude users	26
4.7	Other forms of this document	26
4.7.1	Build notes	26
4.7.2	PDF	27
4.7.3	HTML	27
5	Index	27
6	References	29

1 Introduction to org-ref

Org-ref is a library for org-mode [1] that provides rich support for citations, labels and cross-references in org-mode. org-ref is especially suitable for org-mode documents destined for L^AT_EX export and scientific publication. org-ref is also extremely useful for research documents and notes. org-ref bundles

several other libraries that provide functions to create and modify bibtex entries from a variety of sources, but most notably from a DOI.

The basic idea of org-ref is that it defines a convenient interface to insert citations from a reference database (e.g. from a bibtex file(s)), and a set of functional org-mode links for citations, cross-references and labels that export properly to L^AT_EX, and that provide clickable functionality to the user. org-ref interfaces with helm-bibtex to facilitate citation entry, and it can also use reftex.

org-ref provides a fairly large number of utilities for finding bad citations, extracting bibtex entries from citations in an org-file, and functions for interacting with bibtex entries. We find these utilities indispensable in scientific writing.

org-ref is customizable.

org-ref has been in development since sometime in 2013. It was written to aid in the preparation of scientific manuscripts. We have published over a dozen scientific articles with it so far:

- [2]
- [3]
- [4]
- [5]
- [6]
- [7]
- [8]
- [9]
- [10]
- [11]
- [12]
- [13]
- [14]

Be sure to check out the supporting information files for each of these. The org source for the supporting information is usually embedded in the supporting information. You may also find these two perspective articles [15, 16] interesting, as well as this article on data sharing [17] we wrote.

There has been a lot of recent discussion on the org-mode mailing list about a citation syntax for org-mode. It is not clear what the impact of this on org-ref will be. The new syntax will more cleanly support pre/post text, and it will be separate from the links used in org-ref. It is not clear if the new syntax will support all of the citation types, especially those in biblatex, that are supported in org-ref. The new citations are likely to be clickable, and could share functionality with org-ref. The new citation syntax will not cover labels and cross-references though, so these links from org-ref will still exist. We anticipate a long life for org-ref.

Biblatex users should read this section.

If you want to export this document to another format like PDF or HTML, see Other forms of this document.

1.1 Basic usage of org-ref

1.1.1 Bibliography links

org-ref provides a bibliography link to specify which bibtex files to use in the document. You should use the filename with extension, and separate filenames by commas. This link is clickable; clicking on a filename will open the file. Also, if your cursor is on a filename, you will see a minibuffer message about whether the file exists or not. On export, the bibliography will appear at the position where the link is defined. Usually this link goes near the end of your document, e.g. like here.

There is also a nobibliography link, which is useful for specifying a bibliography file, but not listing a bibliography in the exported document.

There is also a bibliographystyle link that specifies the style. This link does nothing but export to a \LaTeX command. For \LaTeX export a bibliography and bibliographystyle link is *required* if you use bibtex. If you use biblatex see biblatex users.

1.1.2 Citations

org-ref uses the `6` to determine which bibtex files to get citations from, and falls back to the bibtex files defined in the variable `ref \LaTeX -default-bibliography` or `org-ref-default-bibliography` if no bibliography link is found. Note

that you **must** include a 6 in your document if you are exporting your document to pdf; `org-ref-default-bibliography` is not used by the L^AT_EX exporter.

For simple citation needs, org-ref is simple to use. At the point you want to insert a citation, you select the "Org -> org-ref -> Insert citation" menu (or use the key-binding `C-c]` by default), select the reference(s) you want in the helm-bibtex buffer and press enter. The citation will be inserted automatically into your org-file. You "select" an entry by using the arrow keys (or `C-n` and `C-p`) to move up and down to the entry you want. You can also narrow the selection by typing a pattern to match, e.g. author name, title words, year, BibTeX key and entry types. For any other field (e.g. keywords), you will need to add it to the variable `bibtex-completion-additional-search-fields`. You can select multiple entries by pressing `C-SPC` to mark entries in the helm-bibtex buffer.

If the cursor is on a citation key, you should see a message in the minibuffer that summarizes which citation it refers to. If you click on a key, you should see a helm selection buffer with some actions to choose, including opening the bibtex entry, opening/getting a pdf for the entry, searching the entry in Web of Science, etc. . .

The default citation type is customizable, and set to "cite". If you want another type of citation type, then type `C-u` before pressing enter in the helm-bibtex selection buffer. You will be prompted for the type of citation you actually want.

Here is a list of supported citation types. You can customize this if you want. If you do not know what all these types are, you probably do not need them. The default cite is what you need. See <http://tug.ctan.org/macros/latex/contrib/natbib/natnotes.pdf> for the cite commands supported in bibtex , and <http://ctan.mirrorcatalogs.com/macros/latex/contrib/biblatex/doc/biblatex.pdf> for the commands supported in biblatex. For most scientific journals, only bibtex is supported.

`org-ref-cite-types`

If the cursor is on a citation, or at the end of the citation, and you add another citation, it will be appended to the current citation.

If you want to *replace* an existing key in a citation, put the cursor on the key, run the insert citation command, and type `C-u C-u` before pressing enter in the helm-bibtex selection buffer. The key will be replaced. Of course, you can just delete it yourself, and add a new key.

Finally, if you do not like the order of the keys in a citation, you can put your cursor on a key and use shift-arrows (left or right) to move the key

around. Alternatively, you can run the command `org-ref-sort-citation-link` which will sort the keys by year, oldest to newest.

You may want to bind a hydra menu to a key-binding or key-chord. For example:

```
(key-chord-define-global "kk" 'org-ref-cite-hydra/body)
```

This will allow you to quickly press `kk` while on a cite link to access functions that can act on the link.

Alternatively, there is now a keymap defined on cite links. By default this map is defined in the variable `org-ref-cite-keymap`. Note if you change any of these, you have to either restart emacs or run `org-ref-generate-cite-links`. I am a fan of the hyper key, so the following are defined by default:

H-b	Open the citation to the bibtex entry
H-u	Open the citation URL
H-p	Open the citation PDF (if it exists)
H-n	Open the citation notes
H-c	Open Web of Science citing articles
H-e	Email the bibtex entry and pdf
H-f	Copy a formatted entry string
H-g	Open Google Scholar
H-w	Copy the key at point
H-y	Paste the key in an existing link

There are navigation bindings:

C-left	Jump to previous key
C-right	Jump to next key
S-left	move key to the left
S-right	move key to the right

You can also sort

S-up	sort keys by ascending year
------	-----------------------------

1.1.3 Pre/post text support

org-ref has basic and limited support for pre/post text in citations. You can get pre/post text by using a description in a cite link, with pre/post text separated by `::`. For example, [See page 20 1, , for example]. It is not easy (maybe not possible) to extend this for the humanities style of citations

(e.g. harvard) with nested pre/post text on multiple citations. If anyone knows how to do it, pull requests are welcome! There is an ongoing effort in org-mode for a new citation syntax that may make this more feasible.

If you use helm-bibtex and would like pre/post text support enabled, you can add `org-ref-format-citation` to `bibtex-completion-format-citation-functions`:

```
(setf (cdr (assoc 'org-mode bibtex-completion-format-citation-functions)) 'org-ref-format-citation)
```

See also: Format of citations and Insert L^AT_EX cite commands. Note that pre/post prompt can also be switched off by setting the variable `bibtex-completion-cite-prompt-format` to nil.

1.1.4 label links

L^AT_EX uses labels to define places you can refer to. These can be labels in the captions of figures and tables, or labels in sections. We illustrate some uses here.

label links are "functional" if you put your cursor on the link, you will get a message in the minibuffer showing you the number of occurrences of that label in the buffer. That number should be one! It is most preferable to put a label link into a caption like this.

Table 1: Another simple table.

y
4
5

org-ref can help you insert unique labels with the command `org-ref-helm-insert-label-link`. This will show you the existing labels, and insert your new label as a link. There is no default key-binding for this.

1.1.5 ref links

A ref link refers to a label of some sort. For example, you can refer to a table name, e.g. Table `??`. You have to provide the context before the ref link, e.g. Table, Figure, Equation, Section, and so on.

Or you can refer to an org-mode label as in Table 3.

Note: You may need to set `org-latex-prefer-user-labels` to t if you refer to times by their "name" for the export to use the name you create.

```
(setq org-latex-prefer-user-labels t)
```

Table 2: A simple table.

x
1
2

Table 3: Another simple table.

$\frac{y}{3}$
2

You can also refer to an org-ref label link as in Table 1.

To help you insert ref links, use the "Org -> org-ref -> Insert ref" menu, or run the command `org-ref-helm-insert-ref-link`. There is no default key-binding for this.

ref links are functional. If you put the cursor on a ref link, you should see a message in the minibuffer with some context of the corresponding label. If you click on the ref link, the cursor will jump to the label.

A brief note about references to a section. You can make a ref link to a CUSTOM_ID. Section 1 has a label link in the headline. That works, but is not too pretty. Section ?? uses the CUSTOM_ID property. For this to work, you should set `org-latex-prefer-user-labels` to t.

Also note that "#+tblname:" and "#+label:" are deprecated in org-mode now, and "#+name:" is preferred.

1. Miscellaneous ref links

org-ref also provides these links:

pageref The page a label is on

nameref The name of a section a label is in

eqref Puts the equation number in parentheses

autoref A command from hyperref that automatically prefixes the reference number.

cref & Cref cleveref – Intelligent cross-referencing (crefrange is not supported)

Note for eqref, you must use a L^AT_EX label like this:

$$e^x = 4 \tag{1}$$

Then you can refer to Eq. (1) in your documents.

Autoref works like this: Table 3, item 1.

You can specify the default ref link type in ‘org-ref-default-ref-type’.

1.1.6 Some other links

org-ref provides clickable links for a

List of Tables

1	Another simple table.	7
2	A simple table.	8
3	Another simple table.	8
and		

List of Figures

. We have to put some text in the link, anything will do. These export as listoftables and listoffigures \LaTeX commands, and they are clickable links that open a mini table of contents with links to the tables and figures in the buffer. There are also interactive commands for this: `org-ref-list-of-tables` and `org-ref-list-of-figures`.

1.1.7 Controlling link messages and tooltips

Org-ref is setup to provide messages in the minibuffer when your cursor is on a link, and tooltips when your mouse hovers over a link. If this is distracting you can turn it off by putting this in your init file:

```
(setq org-ref-show-citation-on-enter nil)
```

Alternatively, you can turn this on and off interactively with the commands:

- `org-ref-mouse-messages-on`
- `org-ref-mouse-messages-off`

Org-ref can also be configured to show bad label,ref and cite links by setting this in your init files:

```
(setq org-ref-show-broken-links t)
```

This may be slow in large files, so you can turn it off by setting that variable to nil.

1.2 org-ref customization of helm-bibtex

org-ref adds a few new features to helm-bibtex. First, we add keywords as a searchable field. Second, org-ref modifies the helm-bibtex search buffer to include the keywords. Since keywords now can have a central role in searching, we add some functionality to add keywords from the helm-bibtex buffer as a new action.

We change the order of the actions in helm-bibtex to suit our work flow, and add some new actions as well. We define a format function for org-mode that is compatible with the usage defined in section 1.1.2. Finally, we add some new fallback options for additional scientific search engines.

1.3 Some basic org-ref utilities

The command **org-ref** does a lot for you automatically. It will check the buffer for errors, e.g. multiply-defined labels, bad citations or ref links, and provide easy access to a few commands through a helm buffer.

org-ref-clean-bibtex-entry will sort the fields of a bibtex entry, clean it, and give it a bibtex key. By default, this function does a lot of cleaning:

1. adds a comma if needed in the first line of the entry
2. makes sure the DOI field is an actual DOI, and not a URL
3. fixes bad year entries
4. fixes empty pages
5. escapes ampersand and percentage signs
6. generate a key according to your setup
7. runs your hook functions
8. sorts the fields in the entry
9. checks the entry for non-ascii characters

10. converts article title to title case (note: see below to convert titles in other entry types)

This function has a hook `org-ref-clean-bibtex-entry-hook`, which you can add functions to of your own. Each function must work on a bibtex entry at point. (Note: the default behavior can be changed by removing the relevant functions from the initial value of `org-ref-clean-bibtex-entry-hook`.)

```
(add-hook 'org-ref-clean-bibtex-entry-hook 'org-ref-replace-nonascii)
```

`org-ref-extract-bibtex-entries` will create a bibtex file from the citations in the current buffer.

1.4 L^AT_EX export

All org-ref links are designed to export to the corresponding L^AT_EX commands for citations, labels, refs and the bibliography/bibliography style. Once you have the L^AT_EX file, you have to build it, using the appropriate latex and bibtex commands. You can have org-mode do this for you with a setup like:

```
(setq org-latex-pdf-process
      '("pdflatex -interaction nonstopmode -output-directory %o %f"
        "bibtex %b"
        "pdflatex -interaction nonstopmode -output-directory %o %f"
        "pdflatex -interaction nonstopmode -output-directory %o %f"))
```

I have also had success with this setup:

```
(setq org-latex-pdf-process (list "latexmk -shell-escape -bibtex -f -pdf %f"))
```

1.4.1 BibT_EX users

In order to have your citations compiled by BibT_EX, you **must** include a 6. `org-ref` has a customization variable for setting the default bibliography, `org-ref-default-bibliography`, but this is **not used in L^AT_EX export**; you have to set it explicitly in your file. You will also need to include a 6, unless you are using a L^AT_EX style that includes that for you.

1.4.2 biblatex users

In the latex header you specify the style and bibliography file using `addbibresource`. Here is an example:

```
#+latex_header: \usepackage[citestyle=authoryear-icomp,bibstyle=authoryear, hyperref=t]{biblatex}
#+latex_header: \addbibresource{tests/test-1.bib}
```

Sometimes it is necessary to tell bibtex what dialect you are using to support the different bibtex entries that are possible in biblatex. You can do it like this globally.

```
(setq bibtex-dialect 'biblatex)
```

Or like this in a bibtex file:

```
% -*- mode:bibtex; eval: (bibtex-set-dialect 'biblatex); -*-
```

Make sure you invoke biblatex in the `org-latex-pdf-process`. Here is one way to do it.

```
(setq org-latex-pdf-process
      '("latexmk -shell-escape -bibtex -pdf %f"))
```

Finally, where you want the bibliography put this command:

```
\printbibliography
```

1.5 Other exports

There is some basic support for HTML and ascii export. Not all bibtex entry types are supported, but basic support exists for articles and books. For a markdown export, the cite links are exported as Pandoc style links. During HTML export, the references get the HTML class `org-ref-reference`, the bibliography headline has the class `org-ref-bib-h1` and the list of references has the class `org-ref-bib`.

2 org-ref-ivy

`org-ref` provides an alternative to `reftex` and `helm` with `ivy` as the backend completion engine for searching and entering citations.

You can set this backend in your init file like this

```
(setq org-ref-completion-library 'org-ref-ivy-cite)
(require 'org-ref)
```

There are some non-standard ivy features in org-ref ;)

You still use C-c | to search for a bibtex entry, and Ret to insert it as a citation. C-c | C-u Ret will prompt you for a different citation type.

1. You can mark entries with C-space like in helm, and pressing enter will insert the citations.
2. C-, will show you the only the marked entries, and C-. will show them all again.
3. C-up and C-down will move an entry up and down to reorder them.
4. In the selection buffer C-y will sort in increasing year, C-M-y will sort in decreasing year.
5. C-Ret will insert the current entry and move to the next one.

C-u C-c | will insert a ref link. You will see a list of labels to select. Pres Ret to enter a ref link, or C-u Ret to select a different type of ref.

C-u C-u C-c | will insert a label link. You should see a list of all the current labels to help you avoid duplicating them.

3 Other libraries in org-ref

These are mostly functions for adding entries to bibtex files, modifying entries or for operating on bibtex files. Some new org-mode links are defined.

3.1 doi-utils

This library adds two extremely useful tools for getting bibtex entries and pdf files of journal manuscripts. Add this to your emacs setup:

```
(require 'doi-utils)
```

This provides two important commands:

- doi-utils-add-bibtex-entry-from-doi

This will prompt you for a DOI, and a bibtex file, and then try to get the bibtex entry, and pdf of the article.

- `doi-utils-add-entry-from-crossref-query`

This will prompt you for a query string, which is usually the title of an article, or a free-form text citation of an article. Then you will get a helm buffer of matching items, which you can choose from to insert a new bibtex entry into a bibtex file.

This library also redefines the org-mode doi link. Now, when you click on this link you will get a menu of options, e.g. to open a bibtex entry or a pdf if you have it, or to search the doi in some scientific search engines. Try it out `doi:10.1021/jp511426q`.

3.1.1 Bibtex key format

The key is formatted according to the settings of `bibtex-autokey-*` variables. I use these settings. Look at the documentation of them to see how to get the format you want. The function (`bibtex-generate-autokey`) does this.

The settings I use are:

```
(setq bibtex-autokey-year-length 4
      bibtex-autokey-name-year-separator "-"
      bibtex-autokey-year-title-separator "-"
      bibtex-autokey-titleword-separator "-"
      bibtex-autokey-titlewords 2
      bibtex-autokey-titlewords-stretch 1
      bibtex-autokey-titleword-length 5)
```

3.1.2 Troubleshooting doi-utils

Occasionally weird things happen with a DOI. The first thing you should check is if the json data for the DOI can be retrieved. You can do that at the command line, or in a sh block like this:

```
curl -LH "Accept: application/citeproc+json" "http://doi.org/10.1021/jp511426q"
```

If you do not get json data, `doi-utils` will not be able to generate the bibtex entry.

Not all PDFs can be retrieved. `doi-utils` uses a set of functions to attempt this. Here is the list.

`doi-utils-pdf-url-functions`

You can check if a url for the PDF can be found like this:

```
(doi-utils-get-pdf-url "10.1021/jp511426q")
```

3.2 org-ref-bibtex

These are functions I use often in bibtex files.

3.2.1 Generate new bibtex files with adapted journal names

The variable `org-ref-bibtex-journal-abbreviations` contains a mapping of a short string to a full journal title, and an abbreviated journal title. We can use these to create new versions of a bibtex file with full or abbreviated journal titles. You can add new strings like this:

```
(add-to-list 'org-ref-bibtex-journal-abbreviations
  ("JIR" "Journal of Irreproducible Research" "J. Irrep. Res."))
```

`org-ref-bibtex-generate-longtitles` Generate a bib file with long titles as defined in `'org-ref-bibtex-journal-abbreviations'`

`org-ref-bibtex-generate-shorttitles` Generate a bib file with short titles as defined in `'org-ref-bibtex-journal-abbreviations'`

3.2.2 Modifying bibtex entries

`org-ref-stringify-journal-name` replace a journal name with a string in `'org-ref-bibtex-journal-abbreviations'`

`org-ref-set-journal-string` in a bibtex entry run this to replace the journal with a string selected interactively.

`org-ref-title-case-article` title case the title in an article entry.

`org-ref-title-case` title case the title for entries listed in `'org-ref-title-case-types'`.

`org-ref-sentence-case-article` sentence case the title in an article entry.

`org-ref-replace-nonascii` replace nonascii characters in a bibtex entry. Replacements are in `'org-ref-nonascii-latex-replacements'`. This function is a hook function in `org-ref-clean-bibtex-entry`.

The non-ascii characters are looked up in a list of cons cells. You can add your own non-ascii replacements like this. Note backslashes must be escaped doubly, so one `\` is `\\` in the cons cell.

```
(add-to-list 'org-ref-nonascii-latex-replacements
  '("æ" . "{\\\\"ae}"))
```

These functions are compatible with `bibtex-map-entries`, so it is possible to conveniently apply them to all the entries in a file like this:

```
(bibtex-map-entries 'org-ref-title-case-article)
```

3.2.3 Bibtex entry navigation

`org-ref-bibtex-next-entry` bound to M-n

`org-ref-bibtex-previous-entry` bound to M-p

3.2.4 Hydra menus for bibtex entries and files

- Functions to act on a bibtex entry or file

`org-ref-bibtex-hydra/body` gives a hydra menu to a lot of useful functions like opening the pdf, or the entry in a browser, or searching in a variety of scientific search engines.

`org-ref-bibtex-new-entry/body` gives a hydra menu to add new bibtex entries.

`org-ref-bibtex-file/body` gives a hydra menu of actions for the bibtex file.

You will want to bind the hydra menus to a key. You only need to bind the first one, as the second and third can be accessed from the first hydra. You can do that like this before you require `org-ref-bibtex`:

```
(setq org-ref-bibtex-hydra-key-binding "\C-cj")
```

Or this if you like key-chord:

```
(key-chord-define-global "jj" 'org-ref-bibtex-hydra/body)
```

3.2.5 Formatted bibtex entry citations

`org-ref` has some limited capability to make formatted bibliography entries from a bibtex entry or citation link. This is generally a hard problem, and the first solution is not a replacement for a dedicated citation processor like BibTeX. Two variable determine the behavior of formatted citations:

- Formats are from ‘org-ref-formatted-citation-formats’ is an a-list of (backend . formats). formats is an alist of (entry-type . format-string).
- The variable ‘org-ref-formatted-citation-backend’ determines which set of format strings is used. The default is "text", and "org" format strings are also defined.

So, if you click on a citation link, there should be a menu option to copy a formatted citation, which will copy the citation string to the clipboard.

If you are on a bibtex entry, the ‘org-ref-bibtex-hydra/body’ has an option to copy a formatted citation for the entry your cursor is in.

Finally, depending on your org-ref backend, there may be an action in the org-ref-insert-link command.

For the second approach see [./citeproc/readme.org](http://citeproc.readme.org). It is more advanced in someways, but it is currently limited to a few formats, and is a long-term work in progress.

3.3 org-ref-wos

This is a small utility for Web of Science/Knowledge (WOK) (<http://apps.webofknowledge.com>).

```
(require 'org-ref-wos)
```

wos Convenience function to open WOK in a browser.

wos-search Search WOK with the selected text or word at point

There is also a new org-mode link that opens a search:

3.4 org-ref-scopus

This is a small utility to interact with Scopus (<http://www.scopus.com>). Scopus is search engine for scientific literature. It is owned by Elsevier. You must have a license to use it (usually provided by your research institution).

```
(require 'org-ref-scopus)
```

Interactive functions:

scopus Convenience function to open Scopus in a browser.

scopus-basic-search Prompts for a query and opens it in a browser.

scopus-advanced-search Prompts for an advanced query and opens it in a browser.

Some new links: Open a basic search in Scopus:

Open an advanced search in Scopus: `scopus-advanced-search:au-id(7004212771)`. See <http://www.scopus.com/search/form.url?display=advanced&clear=t> for details on the codes and syntax, and http://help.elsevier.com/app/answers/detail/a_id/2347/p/8150/incidents.c\protect\T1\textdollarportal_account_name/26389.

A functional link to a document in Scopus by its "EID": `eid:2-s2.0-72649092395`. Clicking on this link will open a hydra menu to open the document in Scopus, find different kinds of related documents by keywords, authors or references, and to open a page in Scopus of citing documents.

There is also a `scopusid` link for authors that will open their author page in Scopus: `scopusid:7004212771`

3.5 org-ref-isbn

(require 'org-ref-isbn)

This library provides some functions to get bibtex entries for books from their ISBN.

- `isbn-to-bibtex`
- `isbn-to-bibtex-lead`

It also provides some variables for customizing the bibtex entry.

- `org-ref-isbn-clean-bibtex-entry-hook`
- `org-ref-isbn-exclude-fields`
- `org-ref-isbn-field-name-replacements`

3.6 org-ref-pubmed

PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites. This library provides some functions to initiate searches of PubMed from Emacs, and to link to PubMed content.

```
(require 'org-ref-pubmed)
```

This library provides a number of new org-mode links to PubMed entries. See <http://www.ncbi.nlm.nih.gov/pmc/about/public-access-info/#p3> for details of these identifiers. These links open the page in PubMed for the identifier.

pmcid:PMC3498956

pmid:23162369

nihmsid:NIHMS395714

Also, you can retrieve a bibtex entry for a PMID with

- `pubmed-insert-bibtex-from-pmid`

There are some utility functions that may be helpful.

pubmed Open PubMed in a browser

pubmed-advanced Open PubMed at advanced search page.

pubmed-simple-search Prompts you for a simple query and opens it in PubMed.

There is a new org-mode link to PubMed searches:

3.7 org-ref-arxiv

This library provides an org-mode link to <http://arxiv.org> entries: `arxiv:cond-mat/0410285`, and a function to get a bibtex entry and pdfs for arxiv entries:

```
(require 'org-ref-arxiv)
```

- `arxiv-add-bibtex-entry`

- `arxiv-get-pdf`

3.8 org-ref-sci-id

```
(require 'org-ref-sci-id)
```

This package just defines two new org-mode links for <http://www.orcid.org>, and <http://www.researcherid.com>. Here are two examples:

orcid:0000-0003-2625-9232

researcherid:A-2363-2010

3.9 x2bib

(require 'x2bib)

If you find you need to convert some bibliographies in some format into bibtex, this library is a starting point. This code is mostly wrappers around the command line utilities at <http://sourceforge.net/p/bibutils/home/Bibutils>. I thankfully have not had to use this often, but it is here when I need it again.

ris2bib Convert an RIS file to a bibtex file.

medxml2bib Convert PubMed XML to bibtex.

clean-entries Map over a converted bibtex file and "clean it".

3.10 org-ref-latex

This provides some org-ref like capabilities in L^AT_EX files, e.g. the links are clickable with tooltips.

3.11 org-ref-pdf

org-ref-pdf allows Emacs to get bibliography information from pdf files that contain a DOI. You must have **pdftotext** installed where Emacs can find it. This library is known to not work on Windows very well.

- Drag and drop a PDF onto a bibtex file to add a bibtex entry
- If you have a pdf file open in Emacs (from, for example, an email attachment), use **org-ref-pdf-to-bibtex** to try to extract and write the bibliography information to a bib file, (defaulting to **org-ref-default-bibliography**) and save the pdf to **org-ref-pdf-directory**.

3.12 org-ref-url-utils

Allows you to drag-n-drop a webpage from a browser onto a bibtex file to add a bibtex entry (as long as it is from a recognized publisher that org-ref knows about). This library does not work well on Windows.

4 Appendix

4.1 Customizing org-ref

You will probably want to customize a few variables before using org-ref. One way to do this is through the Emacs customization interface: `(customize-group "org-ref")`.

Also see: `(customize-group "org-ref-bibtex")`.

Here is my minimal setup:

```
(setq reftex-default-bibliography '("~/Dropbox/bibliography/references.bib"))

(setq org-ref-bibliography-notes "~/Dropbox/bibliography/notes.org"
      org-ref-default-bibliography '("~/Dropbox/bibliography/references.bib")
      org-ref-pdf-directory "~/Dropbox/bibliography/bibtex-pdfs/")
```

You can also specify different completion backends. The default is 'org-ref-helm-bibtex'.

org-ref-helm-bibtex The default backend that uses helm-bibtex

org-ref-reftex A backend that uses reftex

org-ref-helm-cite An alternative helm completion backend (does not use helm-bibtex)

org-ref-ivy-cite uses ivy for the backend

To use one of these, add a line like this before you "require" org-ref.

```
(setq org-ref-completion-library 'org-ref-ivy-cite)
```

4.2 Customizing how PDFs are opened

4.2.1 Using doc-view or pdf-tools

There are a few different ways in which PDFs can be opened from org-ref. By default, org-ref uses the function `org-ref-open-pdf-at-point`, which looks for the corresponding file in the directory specified in `org-ref-library-path`. If the file was found, it opens it externally with `org-open-file`. To open the PDF from within Emacs, using doc-view or pdf-tools, you will need to modify the function slightly and assign it to the variable `org-ref-open-pdf-function`, as in the example below.

```
(defun my/org-ref-open-pdf-at-point ()
  "Open the pdf for bibtex key under point if it exists."
  (interactive)
  (let* ((results (org-ref-get-bibtex-key-and-file))
        (key (car results))
        (pdf-file (funcall org-ref-get-pdf-filename-function key)))
    (if (file-exists-p pdf-file)
        (find-file pdf-file)
        (message "No PDF found for %s" key))))

(setq org-ref-open-pdf-function 'my/org-ref-open-pdf-at-point)
```

An alternative approach adapted from <https://github.com/jkitchin/org-ref/issues/184> to use pdfview is described here. First, install org-pdfview (on MELPA).

Add this to your init file:

```
;; PDF links for org-mode
(with-eval-after-load 'pdf-tools
  (use-package org-pdftools
    :config
    ;; https://lists.gnu.org/archive/html/emacs-orgmode/2016-11/msg00169.html
    ;; Before adding, remove it (to avoid clogging)
    (delete '("\\.pdf\\'" . default) org-file-apps)
    ;; https://lists.gnu.org/archive/html/emacs-orgmode/2016-11/msg00176.html
    (add-to-list 'org-file-apps
      '("\\.pdf\\'" . (lambda (file link)
                        (org-pdftools-open link))))))
```

4.2.2 A note for Mendeley, JabRef and Zotero users

If `bibtex-completion-pdf-field` is defined, the function below should work with JabRef and Zotero. For more information, see <https://github.com/tmalsburg/helm-bibtex#pdf-files>.

```
(defun my/org-ref-open-pdf-at-point ()
  "Open the pdf for bibtex key under point if it exists."
  (interactive)
  (let* ((results (org-ref-get-bibtex-key-and-file))
        (key (car results))
        (pdf-file (car (bibtex-completion-find-pdf key))))
```

```

      (if (file-exists-p pdf-file)
          (org-open-file pdf-file)
          (message "No PDF found for %s" key))))

(setq org-ref-open-pdf-function 'my/org-ref-open-pdf-at-point)

```

Mendeley users should set to:

```

(setq org-ref-open-pdf-function 'org-ref-get-mendeley-filename)

```

4.3 Setting up notes to work with multiple notes.org files

Adapted from <https://github.com/jkitchin/org-ref/issues/443>. This setup lets helm-bibtex find the note file for an entry.

```

;; Tell org-ref to let helm-bibtex find notes for it
(setq org-ref-notes-function
  (lambda (thekey)
    (let ((bibtex-completion-bibliography (org-ref-find-bibliography)))
      (bibtex-completion-edit-notes
       (list (car (org-ref-get-bibtex-key-and-file thekey)))))))

```

4.4 Customizing the notes format

A user asked about customizing note formats in <https://github.com/jkitchin/org-ref/issues/225>. The issue is that helm-bibtex command creates notes as defined in ‘bibtex-completion-notes-template-one-file’ and org-ref creates the notes as defined in ‘org-ref-open-bibtex-notes’. They wanted some consistency so they got the same results either way they open the notes. @jagrg suggested this approach:

```

(defun my/org-ref-notes-function (candidates)
  (let ((key (helm-marked-candidates)))
    (funcall org-ref-notes-function (car key))))

(helm-delete-action-from-source "Edit notes" helm-source-bibtex)
;; Note that 7 is a magic number of the index where you want to insert the command. You
(helm-add-action-to-source "Edit notes" 'my/org-ref-notes-function helm-source-bibtex 7)

```

4.5 Other things org-ref supports

4.5.1 org-completion

Most org-ref links support org-mode completion. You can type `C-c C-l` to insert a link. You will get completion of the link type, type some characters and press tab. When you select the type, press tab to see the completion options. This works for the following link types:

- bibliography
- bibliographystyle
- all cite types
- ref

4.5.2 Storing org-links to labels

If you are on a label link, or on a table name, or on an org-mode label you can "store" a link to it by typing `C-c l`. Then you can insert the corresponding ref link with `C-c C-l`. This will insert a ref link or custom_id link as needed. This usually works, but it is not used by me too often, so it is not tested too deeply.

4.5.3 Storing links to bibtex entries

If you have a bibtex file open, you type `C-c C-l` with your cursor in a bibtex entry to store a link to that entry. In an org buffer if you then type `C-c l`, you can enter a cite link.

4.5.4 Indexes

org-ref provides links to support making an index in L^AT_EX. (<http://en.wikibooks.org/wiki/LaTeX/Indexing>).

index creates an index entry.

printindex Generates a temporary index of clickable entries. Exports to the L^AT_EX command.

You will need to use the makeidx package, and use this in the L^AT_EX header.


```
#+LATEX_HEADER: \usepackage{makeidx}
#+LATEX_HEADER: \makeindex
```

You will have to incorporate running `makeindex` into your PDF build command.

This is not supported in anything but \LaTeX export.

4.5.5 Glossaries

`org-ref` provides some support for glossary and acronym definitions.

`gls` a reference to a term

`glspl` plural reference to a term

`Gsl` capitalized reference to a term

`Glspl` capitalized plural reference to a term

`gslink` Link to alternative text in the link description.

`glssymbol` The symbol term

`glsdesc` The description of the term

`acrshort` Short version of the acronym

`acrfull` The full definition of the acronym

`acrlong` The full definition of the acronym with (abbrv).

There are two useful commands:

`org-ref-add-glossary-entry` Add a new entry to the file

`org-ref-add-acronym-entry` Add a new acronym to the file

Here is an example of glossary link for an acronym and an actual TLA. Each link has a tool tip on it that shows up when you hover the mouse over it. These links will export as the \LaTeX commands need by the glossaries package.

You will need to incorporate running the command `makeglossaries` into your PDF build command. You also need use the glossaries \LaTeX package.

Here is a minimal working example of an org file that makes a glossary.

```

#+latex_header: \usepackage{glossaries}
#+latex_header: \makeglossaries

#+latex_header_extra: \newglossaryentry{computer}{name=computer,description={A machine}}

A gls:computer is good for computing. Gls:computer is capitalized. We can also use a b

\printglossaries

```

This is not supported in anything but L^AT_EX export.

4.6 Prelude users

An issue related to using org-ref with prelude was reported, along with a solution that loads org-ref before loading prelude. The solution was to create a file like `~/emacs.d/personal/preload/preload.el` with these contents.

```

(package-initialize)

(setq reftex-default-bibliography '("~/bibliography/references.bib"))
(setq org-ref-bibliography-notes "~/bibliography/notes.org"
      org-ref-default-bibliography '("~/bibliography/references.bib")
      org-ref-pdf-directory "~/bibliography/bibtex-pdfs/")

(require 'org-ref)

(provide 'preload)

;;; preload.el ends here

```

4.7 Other forms of this document

4.7.1 Build notes

Before building this file you need to require the following libraries so the links will be resolved.

```

(require 'org-id)
(require 'org-ref-wos)

```

```
(require 'org-ref-scopus)
(require 'org-ref-pubmed)
```

4.7.2 PDF

You may want to build a pdf of this file. Here is an emacs-lisp block that will create and open the PDF.

```
(let ((org-export-with-broken-links t)
      (org-latex-pdf-process
       '("pdflatex -interaction nonstopmode -shell-escape -output-directory %o %f"
         "bibtex %b"
         "makeindex %b"
         "pdflatex -interaction nonstopmode -shell-escape -output-directory %o %f"
         "pdflatex -interaction nonstopmode -shell-escape -output-directory %o %f")))
  (org-open-file (org-latex-export-to-pdf)))
```

4.7.3 HTML

You may want to build an html version of this file. Here is an emacs-lisp block that will create and open the html in your browser. You will see the bibliography is not perfect, but it is pretty functional.

```
(let ((org-export-with-broken-links t))
  (browse-url (org-html-export-to-html)))
```

5 Index

This is a functional link that will open a buffer of clickable index entries:

arxiv, 17

biblatex, 4

bibliography, 2

bibliographystyle, 2

bibtex

clean entry, 8

conversion, 18

- cite, 3
 - replace, 4
 - shift, 4
- completion, 22
- customization, 19
- doi, 11
- export
 - ascii, 10
 - html, 10
 - LaTeX, 9
- glossary, 23
- helm-bibtex, 8
- index, 22
- isbn, 16
- label, 5
- link
 - completion, 22
 - storing, 22
- list of figures, 7
- list of tables, 7
- natbib, 4
- orcid, 18
- pubmed, 17
- ref, 6
 - eqref, 7
 - nameref, 7
 - pageref, 7
- researcher id, 18

6 References

References

- [1] Carsten Dominik. *The Org Mode 8 Reference Manual - Organize your life with GNU Emacs*. Samurai Media Limited, 2014.
- [2] Alexander P. Hallenbeck and John R. Kitchin. Effects of O₂ and SO₂ on the capture capacity of a primary-amine based polymeric CO₂ sorbent. *Industrial & Engineering Chemistry Research*, pages 10788–10794, 2013.
- [3] Prateek Mehta, Paul A. Salvador, and John R. Kitchin. Identifying potential BO₂ oxide polymorphs for epitaxial growth candidates. *ACS Appl. Mater. Interfaces*, 6(5):3630–3639, 2014.
- [4] Zhongnan Xu and John R. Kitchin. Relating the electronic structure and reactivity of the 3d transition metal monoxide surfaces. *Catalysis Communications*, 52:60–64, 2014.
- [5] Zhongnan Xu and John R. Kitchin. Probing the coverage dependence of site and adsorbate configurational correlations on (111) surfaces of late transition metals. *J. Phys. Chem. C*, 118(44):25597–25602, 2014.
- [6] Spencer D. Miller, Vladimir V. Pushkarev, Andrew J. Gellman, and John R. Kitchin. Simulating temperature programmed desorption of oxygen on Pt(111) using DFT derived coverage dependent desorption barriers. *Topics in Catalysis*, 57(1-4):106–117, 2014.
- [7] Matthew T. Curnan and John R. Kitchin. Effects of concentration, crystal structure, magnetism, and electronic structure method on first-principles oxygen vacancy formation energy trends in perovskites. *The Journal of Physical Chemistry C*, 118(49):28776–28790, 2014.
- [8] Jacob R. Boes, Gamze Gumuslu, James B. Miller, Andrew J. Gellman, and John R. Kitchin. Estimating bulk-composition-dependent H₂ adsorption energies on Cu_xPd_{1-x} alloy (111) surfaces. *ACS Catalysis*, 5:1020–1026, 2015.
- [9] Zhongnan Xu, Jan Rossmeisl, and John R. Kitchin. A linear response DFT+U study of trends in the oxygen evolution activity of transition metal rutile dioxides. *The Journal of Physical Chemistry C*, 119(9):4827–4833, 2015.

- [10] Zhongnan Xu and John R. Kitchin. Relationships between the surface electronic and chemical properties of doped 4d and 5d late transition metal dioxides. *The Journal of Chemical Physics*, 142(10):104703, 2015.
- [11] Jacob R. Boes, Mitchell C. Groenenboom, John A. Keith, and John R. Kitchin. Neural network and Reaxff comparison for Au properties. *Int. J. Quantum Chem.*, 116(13):979–987, 2016.
- [12] Jacob R. Boes and John R. Kitchin. Neural network predictions of oxygen interactions on a dynamic Pd surface. *Molecular Simulation*, pages 1–9, 2017.
- [13] Zhongnan Xu, Paul A. Salvador, and John R. Kitchin. First-principles investigation of the epitaxial stabilization of oxide polymorphs: TiO_2 on $(\text{Sr}_{\text{Ba}})\text{TiO}_3$. *ACS Applied Materials & Interfaces*, 9(4):4106–4118, 2017. PMID: 28004912.
- [14] Feiyang Geng, Jacob R. Boes, and John R. Kitchin. First-principles study of the Cu-Pd phase diagram. *Calphad*, 56:224–229, 2017.
- [15] John R. Kitchin. Examples of effective data sharing in scientific publishing. *ACS Catalysis*, 5(6):3894–3899, 2015.
- [16] John R. Kitchin. Data sharing in surface science. *Surface Science*, 647:103–107, 2015.
- [17] John R. Kitchin, Ana E. Van Gulick, and Lisa D. Zilinski. Automating data sharing through authoring tools. *International Journal on Digital Libraries*, pages 1–6, 2016.