

# Tutorial: Writing papers for ACPD using emacs org-mode

Arne Babenhauserheide

February 28, 2014

Emacs Org mode is an excellent tool for reproducible research,<sup>1</sup> but research is only relevant if people learn about it.<sup>2</sup> To reach people with scientific work, you need to publish your results in a Journal, so I show here how to publish in ACPD with Emacs Org mode.<sup>3</sup>

## Contents

<b>1</b>	<b>Requirements</b>	<b>2</b>
<b>2</b>	<b>Basic Setup</b>	<b>2</b>
2.1	Emacs . . . . .	2
2.2	The working folder . . . . .	4
<b>3</b>	<b>The org-mode document</b>	<b>5</b>
3.1	Use the L <sup>A</sup> T <sub>E</sub> X class . . . . .	6
3.2	Delayed table of contents . . . . .	7
3.3	Delayed maketitle . . . . .	7
3.4	Define minted style . . . . .	7
3.5	Intro and conclusions . . . . .	7
3.6	Appendix . . . . .	7
3.7	Bibliography . . . . .	8
3.8	Babel evaluate without confirmation . . . . .	8

---

<sup>1</sup>General methods for using Emacs org-mode in scientific publishing have been described by Schulte et al. (2012).

<sup>2</sup>Research, or rather *science* not only means to learn new things and to uncover secrets, but just as importantly to share what you learn. Fun fact: The German word for science is “Wissenschaft”, built from the words “wissen” (knowledge) and “schaft” (from schaffen: create), so it more exactly captures the essence of scientific work than the word “science”, that is based on the latin word “scientia” which just means knowledge. It isn’t enough to just learn. Creating knowledge requires telling it to others, so they can build upon it.

<sup>3</sup>I chose ACPD as target for this article, because it is an Open Access journal, and because I want to publish in it (which makes it a rather natural choice for a tutorial).

## 1 Requirements

To use this tutorial, you need

- a fairly recent version of org-mode (8.0 or later - not yet shipped with emacs 24.3, so you will need to install it separately) and naturally
- Emacs. Also you need to download the
- copernicus latex package. And it can't hurt to have a look at the latex-instructions from ACP. I used them to create my setup.
- lineno.sty. This is required by copernicus, but not included in the package - and neither in the texlive version I use.

## 2 Basic Setup

### 2.1 Emacs

The first step in publishing to ACPD is to activate org-mode and latex export and to create a latex-class in Emacs. To do so, just add the following to your `~/emac` (or `~/emacs.d/init.el`) and eval it (for example by moving to the closing parenthesis and typing `C-x C-e`):

```

(require 'org)
(require 'org-latex)
(require 'ox-latex)
(setq org-latex-packages-alist
      (quote ((" " "color" t) (" " "minted" t) (" " "parskip" t)))
      org-latex-pdf-process
      (quote (
"pdflatex -interaction nonstopmode -shell-escape -output-directory %o %f"
"bibtex $(basename %b)"
"pdflatex -interaction nonstopmode -shell-escape -output-directory %o %f"
"pdflatex -interaction nonstopmode -shell-escape -output-directory %o %f")))
      (add-to-list 'org-latex-classes
        '("copernicus_discussions"
          "\\documentclass{copernicus_discussions}
[NO-DEFAULT-PACKAGES]
[PACKAGES]
[EXTRA]"
          ("\\section{%s}" . "\\section*{%s}")
          ("\\subsection{%s}" "\\newpage" "\\subsection*{%s}" "\\newpage")
          ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
          ("\\paragraph{%s}" . "\\paragraph*{%s}")
          ("\\subparagraph{%s}" . "\\subparagraph*{%s}"))
        ))
)

```

This allows you to use `#+Latex_Class: copernicus_discussions` in your org-mode file to set the PDF to export for ACPD.

Also you will likely want to use `refTeX` for nice `bibtex` integration. To get it, add the following to your `~/.`emacs or `~/.`emacs.d/init.el:

```

(require 'reftex-cite)
(defun org-mode-reftex-setup ()
  (interactive)
  (and (buffer-file-name) (file-exists-p (buffer-file-name))
    (progn
      ; Reftex should use the org file as master file. See C-h v TeX-master for infos.
      (setq TeX-master t)
      (turn-on-reftex)
      ; enable auto-revert-mode to update reftex when bibtex file changes on disk
      (global-auto-revert-mode t) ; careful: this can kill the undo
      ; history when you change the file
      ; on-disk.

      (reftex-parse-all)
      ; add a custom reftex cite format to insert links
      ; This also changes any call to org-citation!
      (reftex-set-cite-format
        '((?c . "\\citet{%l}" ; natbib inline text
          (?i . "\\citep{%l}" ; natbib with parens
            )))
      (define-key org-mode-map (kbd "C-c ") 'reftex-citation)
      (define-key org-mode-map (kbd "C-c (") 'org-mode-reftex-search))

      (add-hook 'org-mode-hook 'org-mode-reftex-setup)

```

The first line adds reftex-citations with `C-c /`, the rest sets some reftex-defaults and adds a menu which allows you to chose using `\citep{}` instead of `\cite{}` (this is what ACPD requires).

For nice Sourcecode highlighting, you should also install Pygmentize and then add the following to your `.emacs.d`:

```

(add-to-list 'org-latex-packages-alist '(" " "minted"))
(add-to-list 'org-latex-packages-alist '(" " "color"))
(setq org-latex-listings 'minted)

; add emacs lisp support for minted
(setq org-latex-custom-lang-environments
  '(emacs-lisp "common-lispcode"))

```

## 2.2 The working folder

As next step, unzip the copernicus latex package in the folder you want to use for writing your article (do use a dedicated folder for that: `org-mode` leaves around some files. And remember to use a version-tracking system like Mercurial, so you can always take snapshots of your current state.

This will give you the following files:

- authblk.sty
- copernicus.bst
- copernicus\_discussions.cls
- natbib.sty
- pdfscreen.sty
- pdfscreencop.sty

Ensure that all of them are in your folder, not in a subfolder. If necessary copy them there.

Also get lineno.sty and copy it into your folder.

If you want to use unicode-symbols in your text, add uniinput.sty, too.

### **3 The org-mode document**

Using the ACPD style requires some deviations from the standard org-mode export process. Luckily org-mode is flexible to adapt to them. Setup your document as follows:

```

##title: YOUR TITLE
##Options: toc:nil ~:nil
##BIND: org-latex-title-command ""
##Latex_Class: copernicus_discussions
##LaTeX_CLASS_OPTIONS: [acpd, hvmath, online]

# Nice code-blocks
##BEGIN_SRC elisp :noweb no-export :exports results
  (setq org-latex-minted-options
    '(("bgcolor" "mintedbg") ("frame" "single") ("framesep" "6pt")
      ("mathescape" "true") ("fontsize" "\\footnotesize")))
  nil
##END_SRC

##BEGIN_ABSTRACT
Abstract
##END_ABSTRACT
##TOC: headlines 2

##Latex: \runningtitle{SHORT TITLE}
##Latex: \runningauthor{SHORT AUTHOR}
##Latex: \correspondence{AUTHOR NAME\ EMAIL}
##Latex: \affil{YOUR UNIVERSITY}
##Latex: \author[2,*]{SECOND AUTHOR}
##Latex: \author[1]{THIRD AUTHOR SAME INSTITUTE}
##Latex: \affil[2]{SECOND UNIVERSITY}
##Latex: \affil[*]{now at: THIRD UNIVERSITY}

##Latex: \received{}
##Latex: \pubdiscuss{}
##Latex: \revised{}
##Latex: \accepted{}
##Latex: \published{}
##Latex: %% These dates will be inserted by ACPD
##Latex: \firstpage{1}

##Latex: \maketitle

##Latex: \introduction
# * Introduction

* Second section

* Discussion

##Latex: \conclusions
# * Conclusions

##Latex: \appendix

# use acknowledgements for multiple
##BEGIN_acknowledgement
Foo Bar Baz.
##END_acknowledgement

##Latex: \bibliographystyle{copernicus}
##Latex: \bibliography{ABSOLUTE_PATH_TO_YOUR_BIBTEX_FILE_WITHOUT_.bib_SUFFIX}{}

# Local Variables:
# org-confirm-babel-evaluate: nil
# org-export-allow-bind-keywords: t
# End:

```

Let's look at this in more detail.

### 3.1 Use the $\LaTeX$ class

As first step, we set the  $\LaTeX$  class. In the options we select the journal (acpd) and such - you can find the detailed options in the latex-instructions from ACP.

```

##+Latex_Class: copernicus_discussions
##+LaTeX_CLASS_OPTIONS: [acpd, hvmath, online]

```

## 3.2 Delayed table of contents

The table of contents is set to be shown after the Abstract by setting the *toc:nil* option and later explicitly calling *#+TOC: headlines 2*. In org-mode this is really straightforward.

## 3.3 Delayed maketitle

Delaying `\maketitle` is a bit more convoluted than delaying the TOC. First we add the local variable *org-export-allow-bind-keywords: t* at the bottom to allow file-local custom bindings for functions in the file, then we inactivate the title-command with *#+BIND: org-latex-title-command /"/* and finally we add `\maketitle` where we need it.

## 3.4 Define minted style

This defines the variables `minted` uses for beautiful code-blocks. Without this, your code-blocks will just look like inline text.

```
#+BEGIN_SRC elisp :noweb no-export :exports results
  (setq org-latex-minted-options
    '(("bgcolor" "mintedbg") ("frame" "single") ("framesep" "6pt")
      ("mathescape" "true") ("fontsize" "\\footnotesize")))
  nil
#+END_SRC
```

## 3.5 Intro and conclusions

The Introduction and the conclusions have their own commands in ACPD, because they use them to add bookmarks. You can also use the commands to specify another name.

We call the commands with *#+L<sup>A</sup>T<sub>E</sub>X:* (just like some others) which allows us to explicitly add arbitrary L<sup>A</sup>T<sub>E</sub>X-code.

## 3.6 Appendix

The appendix should be used sparingly. It changes the numbering of the pages.

```
#+Latex: \appendix
```

### 3.7 Bibliography

The bibliography allows referring to entries from your general bibtex-file. Ensure that you use the correct absolute path to that file. For more information, see the org-tutorial page for biblatex.

### 3.8 Babel evaluate without confirmation

This allows us to just run all code snippets which we embedded in the document when we export the file. If we do not set this local variable, we have to acknowledge each source block before it runs (the block with local variables also contains the variable which allows binding functions on a per-file basis, as explained above).

```
# Local Variables:  
# org-confirm-babel-evaluate: nil  
# org-export-allow-bind-keywords: t  
# End:
```

## 4 Conclusion

With this setup, you can publish your paper with ACPD using org-mode for the actual writing, which has a much lower overhead than  $\text{\LaTeX}$  and offers quite a few unique features for more efficient working - from easy referencing over inline math preview to planning and code-evaluation directly in your file.

## References

Schulte, E., Davison, D., Dye, T., and Dominik, C.: A Multi-Language Computing Environment for Literate Programming and Reproducible Research, *Journal of Statistical Software*, 46, 1–24, URL <http://www.jstatsoft.org/v46/i03>, 2012.