

Kuma にも使える Org の L^AT_EX エクスポート

しよーた

2015 年 1 月 29 日

目次

1	はじめに	2
2	L^AT_EX エクスポートの手順	2
3	L^AT_EX エクスポートの基本設定	2
3.1	org-latex-classes の詳細	3
3.2	org-latex-pdf-process の詳細	5
3.3	org-latex-packages-alist の詳細	5
3.4	org-latex-default-packages-alist の詳細	6
3.5	org-format-latex-header の詳細	8
4	L^AT_EX エクスポートの再設定	9
4.1	デフォルトの設定を確認する	9
4.2	org-latex-pdf-process を pTeX に変更する	10
4.3	ドキュメントクラスに jsarticle を追加する	10
4.4	ドキュメントクラスに jsbook を追加する	11
4.5	ドキュメントクラスに jsreport を追加する	11
4.6	デフォルトのドキュメントクラスを jsarticle に変更する	12
4.7	hyperref の設定	12
4.8	上文字、下文字の自動変換をオフにする	13
4.9	listings を使ってコードブロックの装飾する	13

1 はじめに

メインである「KumaROOT」のために、Org の L^AT_EX エクスポートについて調べていたら、いい感じの長さになってしまいました。

2 L^AT_EX エクスポートの手順

文章は、だいたい Org (の L^AT_EX エクスポート) + YaTeX で書くことにしています。具体的な手順は以下の通りです。YaTeX でのコンパイルには pTeX (=ptex2pdf) を使っています

Org ファイルの編集 `emacs kumaroot.org`

L^AT_EX エクスポート `C-c C-e l l`

L^AT_EX ファイルのバッファを開く `C-x b kumaroot.tex`

YaTeX コマンドでコンパイル `C-c C-t j`

YaTeX コマンドでプレビュー `C-c C-t p`

3 L^AT_EX エクスポートの基本設定

過去に設定した内容を見直し & テストしながら、基本的であろう項目を拾ってみました。

`ox-latex` L^AT_EX エクスポートを担っているパッケージ^{*1}

`org-export-latex-coding-system` `No document`

`org-export-latex-date-format` `No document`

`org-latex-default-class` L^AT_EX で使うクラスの設定。デフォルトは「`article`」。日本語を使う場合は「`jsarticle`」にする。ただし、後述する「`org-latex-classes`」に追加する必要がある。

`org-file-apps` ファイルを開く外部プログラムを設定できる。何も設定しないとシステムデフォルトのプログラムを使うようになっている。ox-latex というより、org の機

^{*1} 最近 `org-latex` → `ox-latex` に変更されたみたい。(require 'org-latex) しているサイトは内容が古いので、更新日を確認してみましょう。そしてより新しい情報を探しましょう

能。PDF を Preview.app で開くように設定する（もしかして不要なのかな？）。
デフォルトの設定を見ながら検討する。

`org-latex-classes` \LaTeX クラスの alist。Org ファイル中の「`#+LATEX_CLASS:`^{*2}」で指定できる。ない場合はデフォルト値が使われる。詳細は後述。

`org-latex-pdf-process` Org の \LaTeX エクスポート（`C-c C-e l p`）で PDF を作るための設定。内部では `pdflatex` が走ることになる。詳細は後述。

`org-latex-packages-alist` \LaTeX エクスポートした時に、ヘッダに挿入されるパッケージ群。`org-latex-default-packages-alist` の後に追記される。詳細は後述。

3.1 `org-latex-classes` の詳細

```
(class-name
  header-string
  (numberd-section . unnumberd-section)
  (...))
```

3.1.1 第1引数：HEADER-STRING

\LaTeX ファイルの挿入される文字列。`documentclass` や `usepackage` を書くことができる。「`org-latex-default-packages-alist`」や「`org-latex-packages-alist`」に入っているパッケージを呼び出すことができる。Org ファイル内の「`#+LATEX_HEADER:`」、`#+LATEX_HEADER_EXTRA:` キーワードの行で指定することもできる。

1. HEADER-STRING の制御

- 「`DEFAULT-PACKAGES`」 / 「`NO-DEFAULT-PACKAGES`」 (= `org-latex-default-packages-alist`)
- 「`PACKAGES`」 / 「`NO-PACKAGES`」 (= `org-latex-packages-alist`)
- 「`EXTRA`」 / 「`NO-EXTRA`」

2. その他の関連する変数

- `buffer-file-coding-system`
- `org-latex-inputenc-alist`
- `org-export-default-language`

^{*2} 大文字・小文字の区別はない。以下のサンプルコードでも適当に使ってるだけなので気にしないでください

3.1.2 第2引数 : sectioning structure

説明するより、ソースを読んだ方が早いと思う。以下に org-latex-classes デフォルト値を掲載。

```
Original value was
(("article"      ;; <-- the name of the class
;; HEADER-STRING : documentclass (+ default-packages)
"\documentclass[11pt]{article}"
;; SECTIONING : matching for Org headings and LaTeX sectioning
("\section{%s}" . "\section*{%s}")
("\subsection{%s}" . "\subsection*{%s}")
("\subsubsection{%s}" . "\subsubsection*{%s}")
("\paragraph{%s}" . "\paragraph*{%s}")
("\subparagraph{%s}" . "\subparagraph*{%s}"))

("report"
"\documentclass[11pt]{report}"
("\part{%s}" . "\part*{%s}")
("\chapter{%s}" . "\chapter*{%s}")
("\section{%s}" . "\section*{%s}")
("\subsection{%s}" . "\subsection*{%s}")
("\subsubsection{%s}" . "\subsubsection*{%s}"))

("book"
"\documentclass[11pt]{book}"
("\part{%s}" . "\part*{%s}")
("\chapter{%s}" . "\chapter*{%s}")
("\section{%s}" . "\section*{%s}")
("\subsection{%s}" . "\subsection*{%s}")
("\subsubsection{%s}" . "\subsubsection*{%s}")))
```

1. jsarticle の場合

基本的には、「article」部分をコピーして、必要な部分を変更する。

```
("jsarticle"      ;; <-- set to "jsarticle" (any name is OK)
;; HEADER-STRING : change documentclass to jsarticle
;;
: also set dvipdfmx driver
"\documentclass[12pt, dvipdfmx]{jsarticle}"
;; SECTIONING : leave them as it is
```

```
( "\\section{%s}" . "\\section*{%s}" )
( "\\subsection{%s}" . "\\subsection*{%s}" )
( "\\subsubsection{%s}" . "\\subsubsection*{%s}" )
( "\\paragraph{%s}" . "\\paragraph*{%s}" )
( "\\subparagraph{%s}" . "\\subparagraph*{%s}" )
```

3.2 org-latex-pdf-process の詳細

Its value is

```
("pdflatex -interaction nonstopmode -output-directory %o %f"
 "pdflatex -interaction nonstopmode -output-directory %o %f"
 "pdflatex -interaction nonstopmode -output-directory %o %f")
```

Org ファイルから直接 PDF を生成することもできる (C-c C-e l p / C-c C-e l o)。これはそのための設定。デフォルトの設定だと、pdflatex を使っている。3回も回しているとは知らなんだ。

リストになっているのは、途中で bibtex を入れたりできるように。Org 自身に適当なコンパイル方法を検知する仕組みが無いので、ユーザが好きにいじれるようにしてるみたい。

日本語だと pdflatex がうまく動かないので、一度 L^AT_EX ファイルにエクスポートして、pTeX を使ってコンパイルしていたが、これをきちんと設定すれば、楽になるのかも。

3.3 org-latex-packages-alist の詳細

A cell is of the **format**:

```
("options" "package" SNIPPET-FLAG)
```

第1引数 パッケージのオプション

第2引数 パッケージ名

第3引数 よくわからん

3.3.1 パッケージの追加方法

```
(add-to-list 'org-latex-packages-alist '("" "atbegshi"))
```

```
(add-to-list 'org-latex-packages-alist "\\AtBeginShipoutFirst{\\special{pdf:to

(add-to-list 'org-latex-packages-alist "\\hypersetup{setpagesize=false}")
(add-to-list 'org-latex-packages-alist "\\hypersetup{colorlinks=true}")
(add-to-list 'org-latex-packages-alist "\\hypersetup{linkcolor=blue}")

(add-to-list 'org-latex-packages-alist '(" " "listings"))
(add-to-list 'org-latex-packages-alist '(" " "color"))
(add-to-list 'org-latex-packages-alist '(" " "fancyvrb"))
```

このように「add-to-list」を使って、追加する。「hypersetup」などの先頭につけるバックスラッシュはエスケープする。すると以下のように L^AT_EX ファイルのヘッダに追加される。

```
\hypersetup{linkcolor=blue}
\hypersetup{colorlinks=true}
\hypersetup{setpagesize=false}
\usepackage{fancyvrb}
\usepackage{color}
\usepackage{listings}
\AtBeginShipoutFirst{\special{pdf:tounicode EUC-UCS2}}
\usepackage{atbegshi}
```

・・・なんと、よくわからない順番にソートされる。これについては後で調べる。

「add-to-list」の他に、「add-to-order-list」というのがあるみたいなので使ってみる。また、すでにあるリストから削除する関数がないかも探してみる。

3.3.2 設定するときの注意点

- すべてのファイルに使いたいパッケージであること
- 「org-format-latex-header」の設定とコンフリクトしないこと
- 「org-latex-default-packages-alist」のパッケージとコンフリクトしないこと

、

3.4 org-latex-default-packages-alist の詳細

このリストにあるパッケージは Org-mode を動かすために必要な最低限のパッケージなので、基本的には変更しないこと。ただし、あるパッケージを使いたいときに、ここに

あるパッケージとコンフリクトするようなら修正するようにする。

3.4.1 リストにあるパッケージ一覧

`inputenc`, `fontenc` for basic font and character selection

`fixltx2e` Important patches of L^AT_EX itself

`graphicx` for including images

`longtable` For multipage tables

`float`, `wrapfig` for figure placement

`rotating` for sideways figures and tables

`ulem` for underline and strike-through

`amsmath` for subscript and superscript and math environments

`textcomp`, `marvosymb`, `wasysym`, `amssymb` for various symbols used for interpreting the entities in ‘org-entities’. You can skip some of these packages if you don’t use any of their symbols.

`hyperref` for cross references

3.4.2 デフォルト設定

Emacs24.1 で多少変更されたいらしい。

```
Value:
(("AUTO" "inputenc" t)
 ("T1" "fontenc" t)
 (" " "fixltx2e" nil)
 (" " "graphicx" t)
 (" " "longtable" nil)
 (" " "float" nil)
 (" " "wrapfig" nil)
 (" " "rotating" nil)
 ("normalem" "ulem" t)
 (" " "amsmath" t)
 (" " "textcomp" t)
 (" " "marvosym" t)
 (" " "wasysym" t)
 (" " "amssymb" t)
 (" " "hyperref" nil)
 "\\tolerance=1000")
```

3.5 org-format-latex-header の詳細

あとでちゃんと読もうかな。

3.5.1 Documentation:

The document header used for processing L^AT_EX fragments. It is imperative that this header make sure that no page number appears on the page. The package defined in the variables ‘org-latex-default-packages-alist’ and ‘org-latex-packages-alist’ will either replace the placeholder "[PACKAGES]" in this header, or they will be appended. Documentation: The document header used for processing L^AT_EX fragments. It is imperative that this header make sure that no page number appears on the page. The package defined in the variables ‘org-latex-default-packages-alist’ and ‘org-latex-packages-alist’ will either replace the placeholder "[PACKAGES]" in this header, or they will be appended.

3.5.2 デフォルト値

```
"\\documentclass{article}\\n
\\usepackage[usenames]{color}\\n
[PACKAGES]\\n
[DEFAULT-PACKAGES]\\n
\\pagestyle{empty}      % do not remove\\n

% The settings below are copied from fullpage.sty\\n
\\setlength{\\textwidth}{\\paperwidth}\\n
\\addtolength{\\textwidth}{-3cm}\\n
\\setlength{\\oddsidemargin}{1.5cm}\\n
\\addtolength{\\oddsidemargin}{-2.54cm}\\n
\\setlength{\\evensidemargin}{\\oddsidemargin}\\n
\\setlength{\\textheight}{\\paperheight}\\n
\\addtolength{\\textheight}{-\\headheight}\\n
\\addtolength{\\textheight}{-\\headsep}\\n
\\addtolength{\\textheight}{-\\footskip}\\n
\\addtolength{\\textheight}{-3cm}\\n
\\setlength{\\topmargin}{1.5cm}\\n
\\addtolength{\\topmargin}{-2.54cm}"
```


4 L^AT_EX エクスポートの再設定

基本設定の項目が分かったので、もう一度設定を見直してみます。

4.1 デフォルトの設定を確認する

org-latex の設定をすべてコメントアウトして、エクスポートしてみたとき L^AT_EX ファイルのヘッダは以下のようにになっている。

```
\documentclass[11pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{fixltx2e}
\usepackage{graphicx}
\usepackage{longtable}
\usepackage{float}
\usepackage{wrapfig}
\usepackage{rotating}
\usepackage[normalem]{ulem}
\usepackage{amsmath}
\usepackage{textcomp}
\usepackage{marvosym}
\usepackage{wasysym}
\usepackage{amssymb}
\usepackage{hyperref}
\tolerance=1000
\author{しょーた{}}
\date{\today}
\title{\LaTeXエクスポートのテスト用{}}
\hypersetup{
  pdfkeywords={},
  pdfsubject={},
  pdfcreator={Emacs 24.4.1 (Org mode 8.2.10)}}
\begin{document}

\maketitle
\tableofcontents
```

このまま、YaTeX 環境でのコンパイル (= pTeX) は可能だが、「dvipdfmx:warning: (error messages)」という警告がたくさん出る。L^AT_EX ファイルを開き、ドキュメントクラスのオプションでドライバを dvipdfmx に指定するとこのエラーはでなくなる。

```
\documentclass[11pt, dvipdfmx]{article}
```

直接出力 (= pdf_latex) も試してみたが、「org-latex-compile: PDF file ./testing.pdf wasn't produced: [package error]」というログが Messages バッファに残り、うまくいっていない。

4.2 org-latex-pdf-process を pTeX に変更する

```
(use-package ox-latex
  :config
  (setq org-latex-pdf-process
    ("ptex2pdf -l -ot -synctex=1 -file-line-error"
     "ptex2pdf -l -ot -synctex=1 -file-line-error"
     "ptex2pdf -l -ot -synctex=1 -file-line-error"
    )
  )
)
```

「Warning: "ptex2pdf -l -ot -synctex=1 -file-line-error" is a malformed function」とう警告が表示され、そもそもの設定がうまくできない。setq ではできんのか？後で調べる。

4.3 ドキュメントクラスに jsarticle を追加する

```
\documentclass[12pt, dvipdfmx]{jsarticle}
```

文字サイズは少し大きく (= 12pt) して、ドライバには dvipdfmx を指定する。sectioning については、article のデフォルトを使う。

```
(add-to-list 'org-latex-classes
  '("jsarticle"
    "\\documentclass[dvipdfmx,12pt]{jsarticle}"
    ("\\section{%s}" . "\\section*{%s}")
    ("\\subsection{%s}" . "\\subsection*{%s}")
    ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
  )
)
```

```
( "\\paragraph{%s}" . "\\paragraph*{%s}")
( "\\subparagraph{%s}" . "\\subparagraph*{%s}")
))
```

4.4 ドキュメントクラスに jsbook を追加する

```
\documentclass[12pt, dvipdfmx]{jsbook}
```

文字サイズは少し大きく (= 12pt) して、ドライバには dvipdfmx を指定する。sectioning については、book のデフォルトを使う。

```
(add-to-list 'org-latex-classes
  ("jsbook"
    "\\documentclass[dvipdfmx,12pt]{jsbook}"
    ("\\part{%s}" . "\\part*{%s}")
    ("\\chapter{%s}" . "\\chapter*{%s}")
    ("\\section{%s}" . "\\section*{%s}")
    ("\\subsection{%s}" . "\\subsection*{%s}")
    ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
  )
)
```

4.5 ドキュメントクラスに jsreport を追加する

```
\documentclass[12pt, dvipdfmx, report]{jsbook}
```

文字サイズは少し大きく (= 12pt) して、ドライバには dvipdfmx を指定する。「jsreport」というクラスファイルはないが、jsbook に report オプションをつければいいらしい。sectioning については、article のデフォルトを部分に chapter を付け加えた。

```
(add-to-list 'org-latex-classes
  ("jsreport"
    "\\documentclass[dvipdfmx,12pt,report]{jsbook}"
    ("\\chapter{%s}" . "\\chapter*{%s}")
    ("\\section{%s}" . "\\section*{%s}")
    ("\\subsection{%s}" . "\\subsection*{%s}")
    ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
    ("\\paragraph{%s}" . "\\paragraph*{%s}")
  )
)
```

```
)  
)
```

4.6 デフォルトのドキュメントクラスを jsarticle に変更する

すべての Org ファイルに「#+latex_class: jsarticle」を付けるのはめんどくさいので、デフォルトに設定する。

```
(setq org-latex-default-class "jsarticle")
```

4.7 hyperref の設定

```
\usepackage{hyperref}  
\hypersetup{  
  setpagesize=false ,      %% <-- この1行がものすごく大事  
  pdfkeywords={},  
  pdfsubject={},  
  pdfcreator={Emacs 24.4.1 (Org mode 8.2.10)}}
```

hyperref パッケージと jsarticle は仲が良くなくて、そのままコンパイルするとページの幅がおかしくなってしまう。これは setpagesize=false とすることで解決する。hyperref パッケージの設定は、hypersetup の中で行うことができる。

しかし、デフォルトの hyperref の中身は、ox-latex.el にハードコーディングされていて追加／変更できないので、以下のように、hypersetup を2回呼び出すことにする。

```
\usepackage{hyperref}  
\hypersetup{  
  pdfkeywords={},  
  pdfsubject={},  
  pdfcreator={Emacs 24.4.1 (Org mode 8.2.10)}}  
\hypersetup{  
  setpagesize=false  
}
```

とりあえずテストしたい場合は、編集している Org ファイルの先頭に「#+latex_header:」もしくは「#+latex_header_extra:」を使って定義するとよい。

latex_header と latex_header_extra の違いを調べるために、以下の順番で hyper-setup を定義してみた。

```
#+latex_header: \hyperref{setpagesize=false}
#+latex_header_extra: \hyperref{colorlinks=true}
#+latex_header: \hyperref{linkcolor=blue}
```

すると、`latex_header > latex_header_extra` の順に書かれることが分かった。いまいちどういう時に順番を考えたらいいのか思いつかないけれど。

```
\usepackage{hyperref}
\hypersetup{setpagesize=false}      %% latex_header:
\hypersetup{linkcolor=blue}         %% latex_header:
\hypersetup{colorlinks=true}        %% latex_header_extra:
\tolerance=1000
\authorしょうた{}
\date{\today}
\title{\LaTeXエクスポートのテスト用{}}
\hypersetup{
  pdfkeywords={},
  pdfsubject={},
  pdfcreator={Emacs 24.4.1 (Org mode 8.2.10)}}

```

出力場所は、デフォルト出力の `hypersetup` の上になるが、コンパイルには影響しないのでこれで良しとする。

4.8 上文字、下文字の自動変換をオフにする

```
(setq org-use-sub-superscripts nil)
(setq org-export-with-sub-superscripts nil)
```

Org ファイル中の「`^` (ハット)」「`_` (アンダースコア)」以降の数文字は、自動的に上文字、下文字に変換されてしまいます。便利なのかもしれませんが、意図しない箇所も変換されてしまうのはやっぱり不便なのでオフにします。エクスポートするときも同じ理由でオフにしておきます。

上付き・下付きにしたい場合は、文字`^{上付き}`、文字`_ {下付き}`、のように中括弧 (`{}`) で囲みます。Org ファイル中で「`C-c C-x \`」すればプレビューできます。

4.9 listings を使ってコードブロックの装飾する

```
(setq org-latex-listings t)
(add-to-list 'org-latex-packages-alist '(" " "listings"))
(add-to-list 'org-latex-packages-alist '(" " "color"))
(add-to-list 'org-latex-packages-alist '(" " "fancyvrb"))
```

listings パッケージの初期設定は lstset を使う。hypersetup のときと同じように、Org ファイルの先頭に書いておく。

```
#+latex_header: \lstset{language=[LaTeX]TeX}
#+latex_header: \lstset{basicstyle=\small}
#+latex_header: \lstset{stringstyle=\ttfamily}
#+latex_header: \lstset{commentstyle=\ttfamily}
#+latex_header: \lstset{showstringspaces=false}
#+latex_header: \lstset{frame=shadowbox}
#+latex_header: \lstset{rulesepcolor=\color{black}}
#+latex_header: \lstset{fancyvrb=true}
```

5 簡単なテスト方法

L^AT_EX エクスポートの設定のテスト^{*3}のために、必要な emacs 設定ファイルを編集して、ロードして、確認して、とやっていると結構疲れます。

Org 文書中に記述できるソースコードのブロックは、なんと内容を実行する機能がついています。なので、テストの際はそれを使うと格段に捗ります。あるので、それを使うと

```
/#+BEGIN_SRC emacs-lisp :exports results :results silent
(setq org-latex-listings t)
(add-to-list 'org-latex-packages-alist '(" " "listings"))
(add-to-list 'org-latex-packages-alist '(" " "color"))
/#+END_SRC
```

上のコードを Org 文書の最初の方に書いておきます。エクスポートした L^AT_EX 文書のヘッダには、「listings」「color」パッケージが追加されています。

```
\usepackage{hyperref}
\tolerance=1000
\usepackage{color}      %% <-- ここに追加されている
```

^{*3} Org の機能なので L^AT_EX エクスポート以外でも使えます

```
\usepackage{listings} %%  
\hypersetup{setpagesize=false}  
\hypersetup{linkcolor=blue}  
\hypersetup{colorlinks=true}  
\authorしょうた{}
```

思った通りの動きが確認できたのち、emacs 設定ファイルに移動させれば OK です。