WordPress to LaTeX with Pandoc and J: Using `TeXfrWpxml.ijs` (3)

Posted: 26 Feb 2012 02:38:53

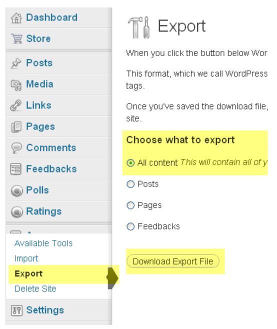
In this post I will describe how to use the J script `TeXfrWpxml.ijs` to generate LaTeX source from WordPress export XML. I am assuming you have worked through (Part 1) and (Part 2) and have:

WordPress to LaTeX

1. Successfully installed and tested Pandoc.
2. Installed and tested a version of J.
3. Set up appropriate directories (Part 2).
4. Know how to use LaTeX.

Item #4 is a *big* if. Inexperienced LaTeX users will probably not enjoy a lot of success with this procedure as the source generated by `TeXfrWpxml.ijs` requires manual edits to produce good results. However, if you're not a LaTeX guru, do not get discouraged. It's not difficult to create blog documents like `bm.pdf`.

Step 1: download WordPress Export XML How to download WordPress export XML is described here. Basically you go to your blog's dashboard, select **Tools**, choose **Export** and select the **All content** option.



Export

When you click the button below Wor

This format, which we call WordPress tags.

Once you've saved the download file, site.

Choose what to export

[✓ All content] *This will contain all of y*

Posts

Pages

Feedbacks

Tools > Export > All Content

When you press the **Download Export File** button your browser will download a single XML file that contains all your posts and comments. Remember where you save this file. I put my export XML here.

```
c:/pd/blog/wordpress/analyzethedatanotthedrivel.wordpress.xml
```

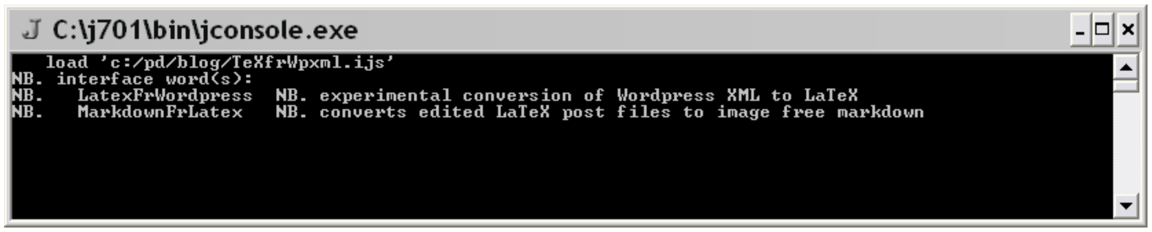
Step 2: download `TeXfrWpxml.ijs` Download `TeXfrWpxml.ijs` and remember where you save it. I put this script here.

```
c:/pd/blog/TeXfrWpxml.ijs
```

Step 3: start J and load `TeXfrWpxml.ijs` `TeXfrWpxml.ijs` was generated from JOD dictionaries. With JOD it's easy to capture root word dependencies and produce complete standalone scripts. `TeXfrWpxml.ijs` needs only the standard J load profile to run. It does not require any libraries or external references and should run on all Windows and Linux versions of J after 6.02. Loading this script is a simple matter of executing:

```
load 'c:/pd/blog/TeXfrWpxml.ijs'
```

The following shows this script running in a J 7.01 console. The console is the most stripped down J runtime.



```
   load 'c:/pd/blog/TeXfrWpxml.ijs'
NB. interface word(s):
NB.    LatexFrWordpress   NB. experimental conversion of Wordpress XML to LaTeX
NB.    MarkdownFrLatex    NB. converts edited LaTeX post files to image free markdown
```

Step 4: review directories and necessary $\LaTeX$ files The conversion script assumes proper directories are available up: see Part 2. The first time you run `TeXfrWpxml.ijs` it's a good idea to check that the directories and files the script is expecting are the ones you want to process. You can verify the settings by displaying `TEXFRWPDIR`, `TEXINCLUSIONS`, `TEXROOTFILE` and `TEXPREAMBLE`.

```
  TEXPREAMBLE
bmamble.tex
  TEXFRWPDIR
c:/pd/blog/wp2latex/
  TEXINCLUSIONS
inclusions
  TEXROOTFILE
bm.tex
  TEXPREAMBLE
```

```
bmamble.tex
```

If all these directories and files exist go to step (5).

Step 5: make sure you are online The first time you run the converter it will attempt to download all the images referenced in your blog. This is where `wget.exe` gets executed. Obviously to download anything you must be connected to the Internet.

Step 6: run `LatexFrWordpress` Run the verb `LatexFrWordpress`. The monadic version of this verb takes a single argument: the complete path and file name of the export XML file you downloaded in step (1).

```
xml=: 'c:/pd/blog/wordpress/analyzethedatanotthedrivel.wordpress.xml'

LatexFrWordpress xml
```

As the verb runs you will see output like:

```
   LatexFrWordpress xml
Whats In it for Facebook?
downloading: c:/pd/blog/wp2latex/inclusions/demotivational-posters-facebook-you.jpg
1 downloaded; 0 not downloaded; 0 skipped
Fake Programming
downloading: c:/pd/blog/wp2latex/inclusions/672169130_vajvn-M.png
1 downloaded; 0 not downloaded; 0 skipped
Laws or Suggestions
downloading: c:/pd/blog/wp2latex/inclusions/i-B5mfdRF-M.jpg
1 downloaded; 0 not downloaded; 0 skipped
Lens Lust

... many lines omitted ...

downloading: c:/pd/blog/wp2latex/inclusions/i-mNK4RHL-M.png
1 downloaded; 0 not downloaded; 0 skipped
WordPress to LaTeX with Pandoc and J: LaTeX Directories (Part 2)
0 downloaded; 0 not downloaded; 1 skipped
+-++
|1||
+-++
```

When the verb terminates you should have a directory `c:/pd/blog/wp2latex` full of `*.tex` files: one file for each blog post. Now the hard work starts.

Step 7: editing LaTeX posts The conversion from WordPress XML to LaTeX produces files that require manual edits. The more images, video, tables and other elements in your posts the more demanding these edits will become. My blog has about one image per post. Most of these images are wrapped by text. LaTeX has a mind of its own when it comes to *floating figures* and getting illustrations to behave requires far more parameter tweaking than it should. This is a longstanding weakness of LaTeX that pretty much everyone bitches about. My advice is start at the front of your document and work through it post by post. The files generated by `LatexFrWordpress` do not attempt to place figures for you but they do bolt in ready-made figure templates as comments that you can experiment with. Each post file is also set up for separate LaTeX compilation. You don't have to compile your entire blog to tweak one post. The one good thing about this edit step is once you have sorted out your old posts you do not have to revisit them unless you make major global document changes. The next time you run `LatexFrWordpress` it will only bring down new posts and images.

Step 8: compile your LaTeX blog I use batch files and shell scripts to drive LaTeX compilations. I processed my blog with this batch file.

```
echo off
rem process blog posting (bm.tex) root file
title Running Blog Master/LaTeX ...

rem first pass for aux file needed by bibtex
lualatex bm

rem generate/reset bbl file
bibtex bm
makeindex bm

rem resolve all internal references - may
rem comment out when debugging entire document
lualatex bm
lualatex bm

rem display pdf - point to prefered PDF reader
title Blog Master/LaTeX complete displaying PDF ...
"C:\Program Files\SumatraPDF\SumatraPDF.exe" bm.pdf
```

The presence of Unicode APL, see this post, forced me to use luaLaTeX. I needed some very nonstandard APL fonts. **See bm.pdf — also available on the *Download this Blog* page** — to judge the effectiveness of my edits. Producing nice figure laden typeset blog documents is work but, as I will describe in the next post, *producing image free eBooks is a simple and far less laborious variation on this process.*

From the blog: Analyze the Data not the Drivel
John D. Baker — revised: August 14, 2020