WordPress has been great to me, but it is slowly becoming a pain to keep updating plugins, security issues, slow performance and the annoying block editor. I am also always looking for additional activities I can do with Emacs. Hugo takes a lot of the pain of managing site away as you can focus on the content. Emacs provides me with excellent editing functionality.

**Convert the content to Markdown or Org Mode**

The first step is to export the WordPress posts database to a CSV file. Several plugins are available that help you with this task. Alternatively, you can [link directly to the database](https://lucidmanager.org/data-science/analyse-site-structure/) and extract the data with the RMySQL package. I have used the [WP All Export plugin](https://wordpress.org/plugins/wp-all-export/) to export the data. We need at least the following fields:

* Title
* Slug
* Date
* Content
* Categories
* Tags

The content files for Hugo are either Markdown or Org Mode. I prefer to use Org Mode as it provides me with access to the extensive functionality that Emacs has to offer. Org Mode is comparable to RMarkdown. You can write and execute code snippets in Org Mode, just like in RMarkdown. Org-Mode has several other advantages because it also has a fully-featured task and project management system. This software also has superior editing options compared to anything that RStudio has to offer. In this code, you set your preferred file type with the export variable.

Screenshot of Emacs with R through the Emacs Speaks Statistics package.

The Content field in the WordPress database contains HTML code. The code below reads the exported CSV file and saves each content field as an HTML file. The mighty Pandoc software undertakes the conversion from HTML to Org Mode or Markdown, depending on the export variable, using the post slug as the file name. Any draft posts or pages in the export file will have NA as the file name.

## Export WP to Hugo

## Read exported WP content

**library**(tibble)

**library**(readr)

**library**(dplyr)

**library**(stringr)

posts <- **read\_csv**("Posts-Export-2020-July-17-2245.csv")

## Convert to Org Mode or Markdown

export <- ".org" # ".org" or ".md"

**for** (i in **1**:**nrow**(posts)) {

filename <- **paste0**(posts$Slug[i], ".html")

**writeLines**(posts$Content[i], filename)

pandoc <- **paste0**("pandoc -o content/post/", posts$Slug[i], export, " ", filename)

**system**(pandoc)

}

## Clean folder

**file.remove**(**list.files**(pattern = "\*.html"))

The next step is to add the front matter for Hugo. The front matter for this export will contain the title, date and the original URL so that we can create a redirect to the new address.

**Export WordPress to Hugo site**

Now that we have some content, we need to provide the context in the front matter so that Hugo can build a site. Hugo knows several types of front matter, i.e. TOML, YAML, JSON and Org-Mode. This code provides either org Mode or TOML front matter for markdown files, depending on how you set the export variable.

## Create Org Mode files

baseurl <- "<https://lucidmanager.org>"

## Create front matter

**if**(export == ".org") {

fm <- **tibble**(title = **paste**("#+title:", posts$Title),

date = **paste**("#+date:", **as.POSIXct**(posts$Date, origin = "1970-01-01")),

lastmod = **paste**("#+lastmod:", **Sys.Date**()),

categories = **paste**("#+categories[]:", **str\_replace\_all**(posts$Categories, " ", "-")),

tags = **paste**("#+tags[]:", **str\_replace\_all**(posts$Tags, " ", "-")),

draft = "#+draft: true") %>%

**mutate**(categories = **str\_replace\_all**(categories, "\\|", " "),

tags = **str\_replace\_all**(tags, "\\|", " "))

} else {

fm <- **tibble**(opening = "+++",

title = **paste0**('title = "', posts$Title, '"'),

date = **paste0**('date = "', **as.POSIXct**(posts$Date, origin = "1970-01-01"), '"'),

lastmod = **paste0**('lastmod = "', **Sys.Date**(), '"'),

categories = **paste0**('categories = ["', posts$Categories, '"]'),

tags = **paste0**('tags = ["', posts$Tags, '"]'),

draft = 'draft = "true"',

close = "+++") %>%

**mutate**(categories = **str\_replace\_all**(categories, "\\|", '", "'),

tags = **str\_replace\_all**(tags, "\\|", '", "'))

}

## Load Hugo files an append front matter

**for** (f in **1**:**nrow**(posts)) {

filename <- **paste0**("content/post/", posts$Slug[f], export)

post <- **c**(**paste**(fm[f, ]), "", **readLines**(filename))

## Repoint images

post <- **str\_replace\_all**(post, **paste0**(baseurl, "/wp-content"), "/images")

## R Code highlighting

post <- **str\_replace\_all**(post, "``` \\{.\*", "

")

post <- str\_replace\_all(post, "```", "

"

)

## Remove remaining WordPress artefacts

post <- **str\_remove\_all**(post, ':::|\\{.wp.\*|.\*\\"\\}')

## Write to disk

**writeLines**(post , filename)

}