

2014

## Decoder Ring: The CMS is Flat

Jerry Spiller

*Art Institute of Charleston*, [yeri.spiller@gmail.com](mailto:yeri.spiller@gmail.com)

Follow this and additional works at: <https://docs.lib.purdue.edu/atg>



Part of the [Library and Information Science Commons](#)

---

### Recommended Citation

Spiller, Jerry (2014) "Decoder Ring: The CMS is Flat," *Against the Grain*: Vol. 26: Iss. 2, Article 55.

DOI: <https://doi.org/10.7771/2380-176X.6740>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

# Decoder Ring — The CMS is Flat

Column Editor: **Jerry Spiller** (Art Institute of Charleston) <yeri.spiller@gmail.com>



**W**eb folks, do you ever get a nagging feeling that WordPress or another database-driven CMS might not be the best solution for a particular site you're building?

When I began to create the site for my Webcomic *Boggy Down* I knew its architecture could be pretty simple and streamlined. Aside from a few one-off static pages like an "About" page, almost all pages would need a recurring blog-like structure that simply housed an image for each page, with enough markup to keep them organized and accessible.

Considering the needs of the project, I found myself asking if the architecture for the underlying content really required the overhead of a database-driven CMS. Did the site need text and media content housed in a SQL database? Or different sets of admins with different levels of privileges? Did most of those admins need WYSIWYG editors? I asked myself those questions about *Boggy Down* and the answer to each one was "no." I had to question if I was looking at WordPress out of habit, despite the fact that another solution might be more appropriate. Simple can be freeing. Why house content in a database if I don't need to? Why have various users in a database if it's only me, and server access is all I need?

I know a lot of librarians working on Web projects face similar concerns, so I'll share the results of my search to find a more streamlined solution for a project that didn't necessarily benefit from MySQL in its stack. Not that WordPress hasn't given us a lot and isn't a great fit for many projects. Still, I'm sure a lot of folks working on presenting a body of images or other media content for digital libraries or special collections, or even reference and instruction, have looked at their back end and wanted something more lightweight.

We aren't alone. I feel that we are in the middle of the rise of the file-based CMS as an alternative. I tested out three contenders for *Boggy Down*: Jekyll, Pico, and Kirby. Kirby's developer echoed the needs of the Webcomic project's architecture: "There's no database behind it. All the administration stuff is done through my ftp client and nice little text files" (Allgeier). This is true of each of these file-based systems.

Jekyll, Pico, and Kirby are each made to use simple files with content marked up with Markdown or HTML as the basis of a site. The HTML doesn't even necessarily need to be fully formed HTML documents; you generally configure a wrapper in header and footer files (not unlike WordPress or many other systems), so the good semantic markup that describes and organizes the pieces of content is all you need.

## Jekyll

Jekyll is built on Ruby and is described by its developer **Tom Preston-Werner** as a "blog aware static site generator" (2013). It is released under the free and open source MIT license. Jekyll powers GitHub Pages, so it can handle large sites with multiple lower-level admins providing loads of content. In addition to Markdown and HTML, it also supports the lightweight markup language Textile as well as Ruby's Liquid templating. It can be extended to work as a blog with a little custom work, or use of its companion Octopress. Octopress provides a blog-ready HTML template for Jekyll and uses Git cloning to support multiple sites (Octopress 2011).

The main caveat I see for Jekyll is that if you aren't accustomed to regularly writing Ruby and maintaining your Ruby environment (I do not and was not) it probably requires too much configuration and too many updates to dependencies to be ready for quick use. If your Ruby is strong, I could see this being a great fit for large digital collections.

## Pico

Pico is a PHP-based CMS by **Gilbert Pellegrom** from **Dev7studios**. It is also released under the MIT license. Pico is built to employ Markdown exclusively, though Markdown files can contain HTML. Pico offers a truly flat directory structure, with .md files stored in a folder called *content*. Each content file typically starts with metadata, with available fields including title, date, author, title and description (Pico). Like our other choices, HTML structure, and CSS styling are completely up to you.

## Kirby

Kirby is a CMS created by designer and developer **Bastian Allgeier**. Like Pico, it is PHP-based. After stumbling across Kirby and experimenting with it a bit, it seemed to match up well with *Boggy Down*'s needs, being light and adaptable without a lot of overhead.

Coding for Kirby should seem instantly familiar to anyone used to writing PHP. Content files use Markdown, in a framework written by Allgeier called Kirbytext. Content files are .txt by default but could be changed to .md or another format in the config file. Kirby wraps the content up in custom, array-like objects based on its directory structure. When writing your PHP and HTML, these can be accessed through its jQuery-inspired API, manifesting a philosophy which might be described as "look for a thing, then use or modify the thing." See Figure I below.

Kirby's sensible folder and object hierarchies, along with organized and easy to follow documentation from **Allgeier**, really set it over the top for me. I found creating and referencing parent-child (and sibling) structure in Kirby more intuitive than in Pico. **Allgeier** even provides an extremely useful reference cheat sheet (Kirby 2013a). Documentation also includes growing list of tutorials for extending Kirby, including writing blog templates, configuring your RSS feed, hiding your backend content files, and creating "art directed" posts with their own unique styling (Kirby 2013b). I found the tutorials easy to follow and implement. *Boggy Down*'s main content area, the comics, is just a blog made with the help of **Allgeier**'s blog tutorial, with the main blog page customized to return only the single most recent post as a page.

Kirby's default metadata might seem less robust than Pico's, but it is easily extensible. You can add any field you like. **Allgeier** offers, "If you need more fields for your template — maybe you want to show a date on that page or a subtitle — simply add more fields to the text file. [...] After each field you have to add four dashes and that's it" (Kirby 2013c). From the PHP, Kirby makes these new fields, which you can think of as custom variables, available from methods with corresponding names:

`$page->yourvar()`

*continued on page 93*

Kirby Cheat Sheet		getkirby.com/docs
<b>main variables</b> <code>\$site</code> , <code>\$pages</code> , <code>\$page</code> available in all templates and snippets	<code>\$pages-&gt;findByName(\$dirname)</code> finds a page by its full dirname incl. number	<code>\$site-&gt;pagination()-&gt;countPages()</code> returns the number of available pages
	<code>\$pages-&gt;findByName(\$title)</code> finds a page by its title	<code>\$site-&gt;pagination()-&gt;countItems()</code> returns the number of items/page objs in a set of pages
<b>\$site</b> contains all site relevant information	<code>\$pages-&gt;filterBy(\$key, \$value, \$split=false)</code> filters a set of pages by a key and a value. if you pass a splitting character as third arg. it will try to split a single value. ie. a comma separated list of tags and search within its elements. returns the filtered set of pages.	<code>\$site-&gt;pagination()-&gt;pageURL(\$page)</code> returns the url for a given page
<code>\$site-&gt;yourvar()</code> the site object can contain any number of custom variables, defined in <code>content/site.txt</code>	<code>\$pages-&gt;first()</code> returns the first page obj in a set of pages	<code>\$site-&gt;pagination()-&gt;firstPage()</code> returns the number of the first page
<code>\$site-&gt;modified()</code> the timestamp of the last content update	<code>\$pages-&gt;last()</code> returns the last page obj in a set of pages	<code>\$site-&gt;pagination()-&gt;isFirstPage()</code> checks if we are on the first page
<code>\$site-&gt;pages()</code> equals the global <code>\$pages</code> variable	<code>\$pages-&gt;count()</code> counts the num. of page objs in a pages set	<code>\$site-&gt;pagination()-&gt;firstPageURL()</code> returns the url for the first page
<b>\$site-&gt;url()</b> the url object provides full access to the current url in the browser address field	<code>\$pages-&gt;visible()</code> returns all visible pages from a pages set	<code>\$site-&gt;pagination()-&gt;lastPage()</code> returns the number of the last page
<code>\$site-&gt;url()-&gt;path()</code> get the full path as string	<code>\$pages-&gt;countVisible()</code>	<code>\$site-&gt;pagination()-&gt;isLastPage()</code> checks if we are on the last page
		<code>\$site-&gt;pagination()-&gt;lastPageURL()</code> returns the url for the last page

Figure I: Kirby's organized and highly usable documentation set a high bar — Image: <http://getkirby.com/docs>

## Decoder Ring from page 92

In the case of the subtitle example above, that would be:

\$page->subtitle()

The worst thing I could say about Kirby is that it prescribes a directory structure and naming scheme to a certain extent. Visible content should be in a subdirectory of Kirby's *content* folder, named according to *number-name* scheme such as "01-blog" or "02-comics." Unnumbered folders are "invisible," meaning they are still Web-accessible but do not appear in auto-generated navigation. Subdirectories follow this same scheme, with the numbers starting over from 01 in each folder.

Again, I found this hierarchy sensible and didn't consider it limiting to the project at hand. Admittedly, it does mean that media content like my image files are spread out in each individual post subdirectory. If you need to house all media together in one directory or in another structure, as I could see being useful for a digital library admin who didn't want to reinvent a wheel, you could get around this limitation with Kirby's custom metadata fields.

After considering your projects needs, I recommend experimenting with each of these flat file systems to see if it's right for the project and the people working on it. All are available on GitHub (Kirby's licensing is based on the honor system). Jekyll is very capable and probably fairly simple to use and maintain if you develop in Ruby. If you're more comfortable in PHP, Pico and Kirby are very similar on first, and even second, glance. I found Kirby easier to work with and extend. Plus, if I had problems Kirby's documentation cleared things up faster than Pico's did.

And then there's the fact that I was making a Webcomic, and what comics creator could resist a name that recalls the King, Jack Kirby?

### References

Allegeier, Bastian. "About." <http://bastianallegeier.com/about>.

Preston-Werner, Tom. "Welcome." *Jekyll*. <http://jekyllrb.com/docs/home/>.

Octopress. 2011. "Octopress 2.0 Surfaces." <http://octopress.org/>.

Pico. "Introducing Pico." <http://pico.devstudios.com/>.

Kirby. 2013. "The Kirby Cheat Sheet." <http://getkirby.com/docs>.

---. "Tutorials." <http://getkirby.com/tutorials>.

---. "Adding Content." <http://getkirby.com/docs/content>. 🐾

## Back Talk from page 94

ways imagined that paradise will be a kind of library." I think this is 100% true and that is why we should all try to help others find this paradise before they cross the veil between this and whatever is coming next. 🌧

## against the grain people profile

### Jennifer Lohmann

Adult Services Manager, Southwest Regional Library  
Durham County Library, 3605 Shannon Rd., Durham, NC 27707

Phone: (919) 560-8594 • Fax: (919) 560-0542  
<jlohmanna@dconc.gov> • <jenniferlohmanna@gmail.com>  
[durhamcountylibrary.org](http://durhamcountylibrary.org) • [jenniferlohmanna.com](http://jenniferlohmanna.com)

**BORN AND LIVED:** I've lived all over the place! I grew up in southern Idaho and Salt Lake City, went to college in Chicago, taught English as a Foreign Language in Shanghai, and then moved to Durham, North Carolina. My first three books are set in Chicago, a city I dearly love.

**IN MY SPARE TIME:** Between writing and working full-time, I hardly feel like I have spare time! I read a lot, both for work (I run two book clubs) and for fun. I like to cook and eat, so I like going out to restaurants and I read a lot of cookbooks. I run and enjoy hiking. I'm interested in quilting, but have only made two baby quilts (though I've bought fabric for several other quilts).

**FAVORITE BOOKS:** I've been telling everyone I meet to read *The Black Count* by **Tom Reiss**. I'm also a big fan of *Wolf Hall* and *Bring Up the Bodies* by **Hilary Mantel**. For romance novels, I love *The Outsider* by **Penelope Williamson**, anything **Carla Kelly** writes, *Simply Love* by **Mary Balogh**, and **Jennifer Crusie's** *Faking It*. I'm sure if you asked me on a different day, you'd get a different list. Asking a librarian this question is a bit like asking a mother to pick her favorite child.

**PHILOSOPHY:** Perfect is the opposite of good. **Anne Klinefelter**, the director of the **UNC Law Library**, told me that my first year of graduate school. Honestly, if I waited for perfect, I would never get anything done (can you tell that I'm a big picture person, instead of a detail-oriented person?).

**MOST MEMORABLE CAREER ACHIEVEMENT:** I have two careers, so I'll answer for both of them. Winning the **RWA Librarian of the Year Award** was fantastic. And getting the phone call from **Megan Long** for my first book was a wonderful feeling. I had been having a really terrible month of May. When **Megan** told me the production schedule and how quickly I'd have to turn the book over, she said, "Do you think you can do that?" My exact response was, "I've spent the past week force feeding a cat. I can do anything." It took **Megan** a couple seconds to respond with anything more coherent than a "um, okay."

**HOW/WHERE DO I SEE THE INDUSTRY IN FIVE YEARS:** eBooks and self-publishing are changing the market a lot (which we all know). Even though libraries and publishers are fighting out the role of libraries and eBooks, with brick-and-mortar stores struggling and the number of books available online is skyrocketing, I think authors and publishers will start placing greater value on the unique role of the library and librarians in getting books into the hands of readers (there are many authors and publishers who recognize this already). 🐾

## Rumors from page 86



Speaking of **ATG** profiles, I noticed that **Anneliese Taylor** (p.30) noted that in her spare time, she enjoys anything that gets her away from a computing device. Ain't it the truth?

And reading the *Forbes* interview with **Anthea Stratigos** (see above) I was interested to see that she and her husband, **Greg Chagaris**, make a point of going computer-less when they vacation. This may be a growing trend?

One last thing before I go, what do you think is the sexiest part about being a librarian? Got this from **Ileana Jacks** who found it on Instagram.

Now I gotta go! But be sure and visit the **ATG archive** on the **Purdue University Press** website not to mention the proceedings as well. <http://docs.lib.purdue.edu/atg/>

Happy Spring! 🌸



Check out **Lindsay Barnett's** new couch. Do you think Charlie was watching the Academy Awards?