

Static Site Generator Usage Notes

What is a static site generator, and why use one?

A static site generator is a software tool that (typically) takes content, applies a layout template, applies styling, and produces plain-old HTML and CSS. In most cases, a generator would take content written in a formatting language like markdown or mediawiki, and produce html from that. A website produced in this way is static, as in the content that is seen on the user's screen is real HTML served up from the file system on the remote server - unlike a CMS (content management system) which uses scripts to query a database to generate, and deliver content on demand.

The advantage of using a static site generator is that:

1. better security: no database to hack, no monthly CMS and plugin security updates to worry about
2. won't break: updates won't break designs/functionality since the only updates are the ones made on the content by the author
3. no CMS-specific knowledge needed for theming / admin
4. LAMP dependency removed (good for presentations, easy deployment, etc.)
5. potentially better a11y: complete control over markup...no surprise injections of ugly html
6. source documents and content can be version controlled using a code repository like github

The disadvantages of using a static site generator are:

1. workflow: it isn't as easy as single-button publishing. Such platforms or tools may exist - more research needed.
2. interactivity: we lose helpful CMS widgets like a comment system, user accounts, analytic widgets. Some of this interactivity can be offloaded to 3rd party services - like LiveFyre for comments.
3. content is written in HTML markdown syntax or some other content format. Know-how is required. (There may be plugins for WYSIWIG that might help)
4. to build the site, knowledge of the static site generator is required (i.e. know-how required for set up, templating)
5. possible lack of content conversion tools? (i.e. joomla or drupal -> docpad)

Current Recommendation

- Static site generator - Use Docpad because of wider adoption and developer support. See this thread for the start of the discussion: <http://lists.idrc.ocad.ca/pipermail/fluid-work/2014-January/009251.html>
- Templating - Use Handlebars which maintains validity of HTML content, see this thread <http://lists.idrc.ocad.ca/pipermail/fluid-work/2014-January/009252.html>
- CSS Framework - Use Foundation as per this recommendation: [Fluid Project Recommendation for Use of CSS Frameworks](#)

Areas of Further Research

- Wysiwyg editor for markdown
- A GUI for Docpad content authoring
- direct publishing from local docpad to github pages
- Find out if there are migration tools (i.e. Joomla to Docpad)

Wordpress or Docpad

Until the above 4 "Areas of Research" are investigated, Wordpress should be considered for websites requiring:

1. Ease of publishing
2. User accounts / privileges and widgets.

IDRC websites using static sites

- Fluid Project - using docpad (in progress)
- Floe Project - plain HTML, no generator
- Fluid Infusion Docs - docpad
- Inclusive Learning Design Handbook - docpad, work yet to be scheduled.

Docpad Installing and Configuring

1. Follow the installation guide at the following link: <http://docpad.org/docs/install>
2. Create an empty directory where you will begin building your new site
3. From inside this new directory, execute `docpad run` in the console.
 - a. Use the "No Skeleton" option since we will choose our own CSS framework configuration / build and other plugins
 - b. Docpad will keep running as a web server and automatically build and deploy new files

Example Directory Structure

```
./out/                (docpad automatically deploys here)
./out/css/
./out/html/
./out/images/
./out/lib/

./src/                (contents of this directory gets compiled and deployed to ./out/)
./src/documents/     (css, html, markdown, any content that can change over time goes in ./src/documents/)
./src/documents/css/
./src/documents/html/
./src/documents/images/

./src/files/         (content that is reasonably static goes here. Note ./src/files/ contents are deployed
sibling to contents of ./src/documents/)
./src/files/lib/     (static libraries go in here. i.e. Infusion, Foundation, jQuery, etc.)

./src/layouts/      (content templates go here)
```

Foundation

1. Build and download a custom Foundation framework. For building websites, we recommend the following components to be omitted from the Foundation build:
 - a. Progress Bars
 - b. Pricing Tables
 - c. Flex Video
 - d. All Foundation JavaScript components
2. Extract Foundation into the following directory: `./src/files/lib/Foundation`

Infusion

- something in the infusion build causes DocPad's web server to stop functioning after deploying Infusion. Tracking this down...

Resources

- [Beginner Guide http://docpad.org/docs/begin](http://docpad.org/docs/begin) - Start here
- [Tweaking docpad to behave more like a site http://takitapart.com/posts/organizing-docpad/](http://takitapart.com/posts/organizing-docpad/) - do this after going through the Beginner Guide tutorial