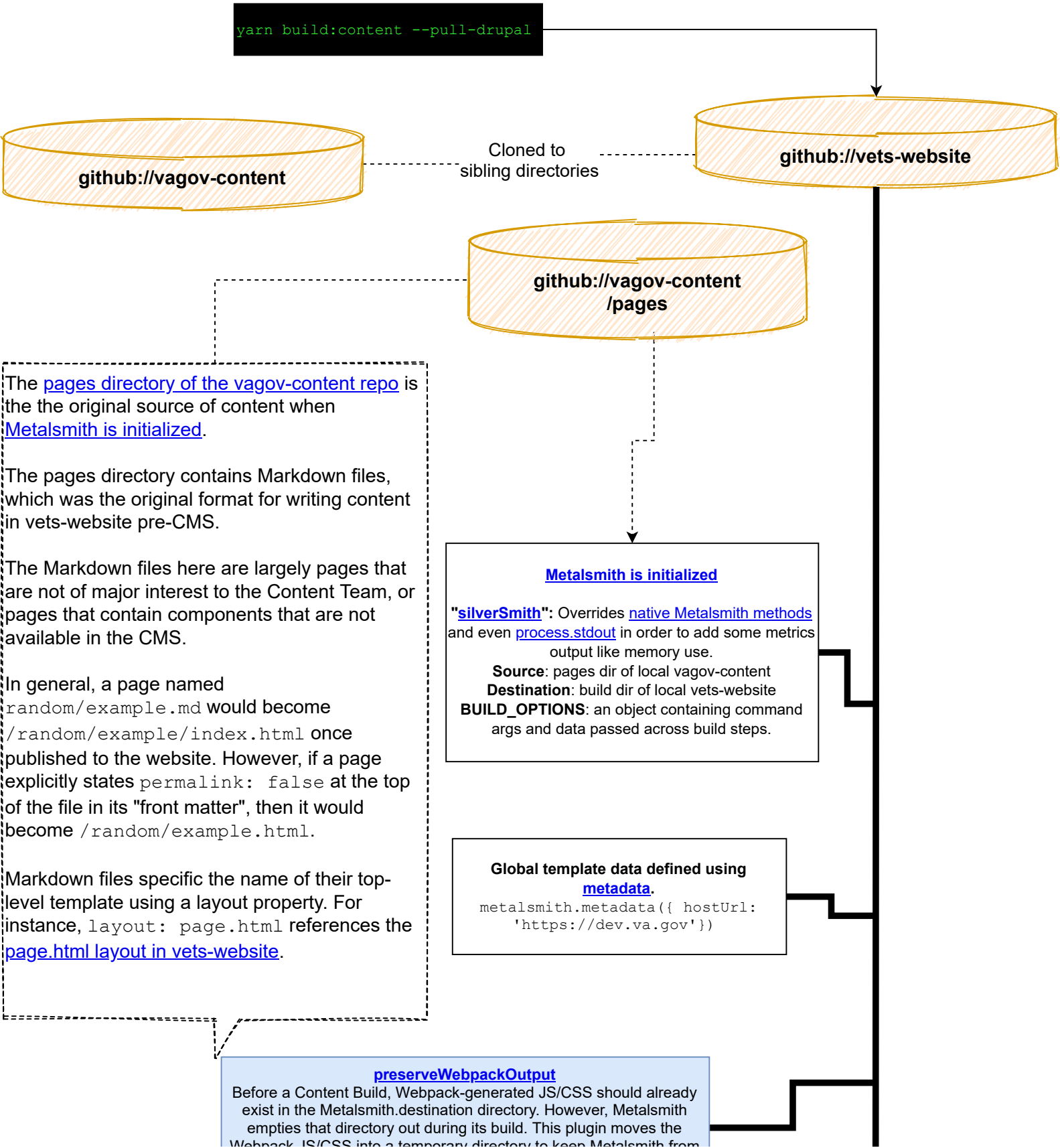


The content build

This document illustrates the process for what we know as the Content Build, which is executed when the command `yarn build:content --pull-drupal` is executed in a vets-website project. It is also executed as the "rerelease_job" during a Content Release. In this case, the commit SHA of the latest vets-website release is used to check out the source code at that time, then a Content Build is executed to rehydrate the static content of the website while leaving Webpack assets the same as was in the original release.



Webpack JS/CSS into a temporary directory to keep Metalsmith from deleting them. Then once Metalsmith starts, it moves it back to the output [destination](#) directory.

[createReactPages](#)

Reads the registry.json to generate HTML landing pages for React apps. These generated files are added into Metalsmith memory.

[getDrupalContent](#)

CMS

[loadDrupal](#)

Decides between GraphQL queries or CMS Export request

[getContentViaGraphQL](#)

Sends GraphQL queries or loads cached data out of a local file, pages.json.

[getAllPagesViaIndividualizedGraphQLQueries](#)

1. Sends a request for [counting](#) certain types of nodes. This is used for pagination.
2. [Gets a list of GraphQL queries](#). This is a function instead of a static list because some queries require CMS feature flags, which are loaded async. So, if it were a static list, the CMS feature flags would not be defined.
3. [Executes the queries in parallel](#).
4. [Consolidates the response JSON](#) of each query into a single data structure.
5. [Returns](#) the final data structure, which will be cached as a pages.json file. At this time, this data is referred to as **drupalData**.

[convertDrupalFilesToLocal](#)

Does a find-and-replace throughout drupalData, replacing CMS domains with AWS domains. It also creates [placeholders](#) in Metalsmith memory for files that need to be downloaded from the CMS in a future build step.

[loadCachedDrupalFiles](#)

Loads images, PDFs and other static files out of local disk, if available. If found, the file won't be fetched from the CMS later.

[pipeDrupalPagesIntoMetalsmith](#)

Each node contained in drupalData.data.nodeQuery is moved into Metalsmith memory as an individual page.

Page data is manipulated

[compilePage](#) stitches together page instances with sidebar data.

For [benefit pages](#), it uses the page's "administration.entity.name" field to determine the sidebar.

For [VAMC pages](#), it uses the page's "office.entity.entityLabel" to determine the sidebar.

[Special operations](#) are performed against nodes based on their entityBundle.

Generated listing pages

Some types of pages are generated, such a [health care region list pages](#).

Homepage generated

The data for the homepage is [generated](#) by stitching together several types of menus and the [YML file from vagov-content](#) representing the homepage banner.

github://vagov-content
/fragments/home/banner.yml

Original Drupal data is made available to future build steps by binding it to [BUILD_OPTIONS.drupalData](#)

vagov-content Markdown and CMS pages are now in Metalsmith memory together.

```
// The "files" variable passed to and from each Metalsmith plugin
{
  "school-administrators.md": {
    title: "Resources for schools",
    layout: "page-breadcrumbs.html",
    // example vagov-content page
  },
  "health-care/index.html": {
    title: "VA health care",
    layout: "page.drupal.liquid",
    // example CMS page
  },
  // thousands and thousands more.
}
```

useDrupalPrefix

1. Moves all CMS pages to "/drupal/" if the [environment](#) is configured to do so. This dates back to years ago and hasn't been used since.
2. If there is a Markdown page in vagov-content whose location conflicts with a page from the CMS, the Markdown page will be [removed](#).

createOutreachAssetsData

Creates data to be used in the /outreach-and-events/outreach-materials/ page.

createResourcesAndSupportWebsiteSection

1. Deletes CMS pages out of Metalsmith memory if "fieldStandalonePage" is false.
2. Generates the category and tag listings pages.
3. Generates the JSON file used to power the search tool on /resources/search/?query=health. [More information](#).

[createEnvironmentFilter](#)

Deletes some files out of Metalsmith memory based on the environment (AKA "buildtype") which indicates dev/staging/prod.

Markdown files that contain "vagovprod: false" in front matter will be deleted from prod builds.

CMS pages of type "campaign_landing_page" will be deleted from prod builds (this product is in beta.)

[filenames](#)

A third-party plugin. Unclear what this does or if it is still needed.

[checkCollections](#)

Validates the "collection" property written in Markdown front matter. Most likely does nothing at this point.

[collections](#)

A third-party plugin that was used to group Markdown pages. It was useful on Vets.gov for sidebars but almost certainly tech debt today.

[leftNavRailResets](#)

Adds a "previous" property onto Markdown files, which was used when VA.gov sidebars were on GitHub instead of the CMS. This is certainly tech debt today.

[dateInFilename](#)

A third-party plugin. Unclear what this is but it dates back to Vets.gov and is almost certainly tech debt.

github://vets-website
/src/site/assets

[assets](#)

Adds into Metalsmith the vets-website assets directory, which contains images and other files that applications may require.

github://vagov-
content/assets

[assets](#)

Adds into Metalsmith the vagov-content assets directory, which contains images and other files that Markdown files may expect.

[inPlace](#)

A third-party [plugin](#) that does something with the Liquid engine.

Unclear what this does but it contains an ominous comment above it.

[markdown](#)

A third-party [plugin](#). Converts the Markdown content of Markdown files in Metalsmith memory into HTML.

[permalinks](#)

Moves HTML files into the their own directory to have the trailing slashes. For instance, /school-administrators.html is moved to /school-administrators/index.html.

[createHeaderFooter](#)

Manipulates the CMS data for the website's main menu to match the expected format of the MegaMenu React component.

More detail, https://github.com/department-of-veterans-affairs/va.gov-team/blob/master/platform/cms/accelerated_publishing/content-build/menus-widgets-apis.md#the-megamenu

[navigation](#)

A third-party [plugin](#). Almost certainly tech debt at this point. Was used for Vets.gov sidebars somehow.

[layouts](#)

A third-party [plugin](#) that we configure to use the Tiny Liquid templating engine. This is what adds meaning to the "layout" property attached to each HTML file in Metalsmith memory.

This plugin is what uses the data from each page in Metalsmith memory to produce actual HTML page content.

For instance, if this was the "files" data in Metalsmith memory -

```
{
  "some/example/index.html": {
    layout: "page/drupal.liquid",
    title: "Example",
    contents: ''
  },
  // ... thousands more
}
```

Then it would be configured to use page.drupal.liquid as its top-level Liquid template. Its "contents" property would contain the HTML result of the templating.

[rewriteVaDomains](#)

Does a find-and-replace throughout all HTML files in Metalsmith memory to link to the va.gov environment of the current build-type.

For instance, if the current build is a staging build, then instances of "https://www.va.gov/health-care/" would be replaced with "https://staging.va.gov/health-care."

See this "[domainReplacements](#)" property.

[rewriteDrupalPages](#)

If enabled in the current environment, it cycles through pages that originated from Drupal, and [replaces](#) links in each page to use the "/drupal/" prefix if the link is to another Drupal page.

This was useful for Drupal's rollout, but is definitely not used today.

[createDrupalDebugPage](#)

Creates a page used to list out all of the pages that originated from the CMS, or an error if the CMS responded with error. This page is at /drupal/debug/. For instance, <https://staging.va.gov/drupal/debug/>.

This was useful for Drupal's rollout, but is definitely not useful today.

Not sure of the relationship between these plugins

