LiquiDoc Developer Manual

Table of Contents

Introducing LiquiDoc and LiquiDoc CMF

Welcome to the LiquiDoc Developer Manual!

Here you will learn to extend and customize a sophisticated, open-source toolchain LiquiDoc as a developer.

This Developer's Manual will get you started modifying and hopefully contributing back to LiquiDoc and the LiquiDoc Content Management Framework.

In this context, a *developer* is someone who hacks the core tooling of a docs project. If your job is to *implement*, *configure*, or *design* a content project *using* LiquiDoc or LDCMF, this is probably not the guide for you. Those tasks are associated with the **administrator** roll.

Who is this Manual for?

In the context of this guide, a *developer* is someone who works to modify the codebase or extend the capacities of LiquiDoc and LiquiDoc CMF. Some of the material in Manual overlaps with the {docpro_guide_link} and/or the {admin_guide_link}.

As a developer contributing to LiquiDoc and LDCMF, you need to understand the basic contribution workflow; plus, you should probably document your changes in the proper place and format. Therefore, a great deal of the material in the {docpro_guide_name} is reproduced here, sometimes with modifications tailored to the developer experience.

Likewise, if you are hacking or adding to LiquiDoc, you likely need a good understanding of LiquiDoc and LDCMF architecture and configuration, so we've included a lot of {admin_guide_name} content in this Manual, as well.

This document contains lots of jargon. See the Jargon Guide if you get lost.

Appendix A: Jargon Guide

This is the full list of specialized terms used in this product documentation. They are also generated as JSON at `/data/terms.json` so we can highlight them in the text when we get to it. This is just to show the power of storing data in flat files reusable throughout product docs.

artifact

A digital package (file or archive) representing a discrete component of a product. Here we use *artifact* to describe a descrete instance of output, such as a single HTML or PDF file, or a Jekyll website or Deck.js slide presentation.

AsciiDoc

Dynamic, lightweight markup language for developing rich, complex documentation projects in flat files. (Resource)

Asciidoctor

Suite of open source tools used to process AsciiDoc markup into various rendered output formats. (Resource)

build

The (usually automated) series of actions necessary to compile and package software or documentation.

Liquid

Open source templating markup language maintained by Shopify (Resource)

YAML

a slightly dynamic, semi-structured data format for nested data (Resource)

Appendix B: NOTICE of Packaged Dependencies

The following open source packages are fully or partially included with LiquiDoc.

Package	**Jekyll Documentation Theme**
License	MIT
Author	Tom Johnson
Website	https://github.com/tomjoht/documentation-theme-jekyll

Package	**M+ OUTLINE FONTS (M+ TESTFLIGHT 058)**
License	unlimited
Author	M+ Fonts Project
Website	http://mplus-fonts.osdn.jp/about-en.html

Package	**Noto Fonts**
License	SIL OFL
Author	Google i18n
Website	https://www.google.com/get/noto/

Package	**Font Awesome**
License	SIL OFL 1.1
Author	Fonticons, Inc
Website	https://fontawesome.com/

Package	**"Coding Style Guide"**
License	MIT
Author	Dan Allen, Paul Rayner, and the Asciidoctor Project
Website	https://github.com/asciidoctor/jekyll-asciidoc/blob/master/coding-style-guide.adoc