Navigating CORD Documentation

This page describes the project's approach to documentation. The community is invited to help us keep documentation up-to-date.

Keep in mind that CORD integrates components from several related but independent projects, some of which have their own wiki pages (and other documentation). CORD documents often include links to information about these other components, but how their documentation is organized and managed may differ from the approach described below.

Wiki

This wiki is assumed to be the starting place for most people that want to either learn more about CORD, or to start contributing to CORD. It contains relatively static and high-level information, with a focus on community-related activity (as opposed to software specifics). This includes, for example, the white papers listed on the Documentation page, and community-oriented informational pages (e.g., A-CORD, M-CORD). Any documentation that is tied to a given release of the software is assumed to be stored in Gerrit and assembled in a GitBook (see below).

Some general and community-focused information that changes on a release-by-release basis is maintained in the wiki. The Roadmap and Service Inventory are two examples. In such cases, information about specific releases is explicitly archived within the page (e.g., see the bottom of the Roadmap page). We do not use the wiki's "Spaces" mechanism to checkpoint per-release pages.

GitBook / Guide

Software-specific documentation, especially anything that potentially changes on a release-by-release basis, is maintained as .md files in the `/docs` directory of each project in Gerrit. The "root" of the GitBook is in https://github.com/opencord/cord/tree/master/docs (checked out by the repo tool as `docs`). All content committed back to Gerrit will also be viewable on a public server at http://guide.opencord.org.

You can find more information in the GitBook documentation, but the general approach for contributing content is to add `.md` files to the appropriate `docs` directories checked out from Gerrit, and then view/test it locally using the GitBook toolchain.

To run the toolchain you'll need NodeJs and virtualenv installed on your machine, and then execute:

```
$ npm install gitbook-cli -g
```

To build a GitBook and serve the content locally (viewable at `http://localhost:4040`) in a checked out repo, run

```
$ make clean
$ make
```

By convention, we build a single composite GitBook that consists of a set of Guides. The composite Guide is currently organized as follows:

- Installation Guide – Information about building and installing CORD on a target POD.
- Operations Guide – Information about operations and maintenance on a running POD, including diagnostic support.
- Modeling Guide – Information about using XOS to model and on-board services into CORD.
- Development Guide – Information about common developer workflows and any developer tools.
- Testing Guide – Information about CORD's testing infrastructure.

Note that many of `.md` files include in the Guide live in one of the `<project>/docs` directories. That directory may contain other `.md` files relevant to the corresponding <project>, where the subset included in the Guide is specified in `docs/SUMMARY.md`. That file must be edited to include new content the composite Guide.

Google Docs

Design documents that are the subject of active collaboration are usually stored as Google Docs, organized according to release (e.g., see satisfying-cactus). As design documents mature (and releases are cut), it is expected that these Google Docs move to either the Wiki or GitBook.

Release and sprint planning is recorded in a similar way, collected in a different Google Docs folder, also organized on a release-by-release basis.

REST APIs

Documentation for CORD's REST APIs is auto-generated using Swagger and is available on any CORD installation under the `apidocs/` endpoint, as well as in the Guide.