Contributor Roles

Contents

1 Prerequisite skills 2

2 Content Author 2

3 Content Reviewer 2

4 The Content Review Process 2

There are multiple ways community members can contribute to the OpenIndiana Documentation revitalization effort. These contributions fall into several different roles. Each role requires a different skill level.

1 Prerequisite skills

Basic familiarity with:

- Text editors
- The Markdown text markup syntax
- GIT distributed version control
- The MkDocs documentation framework

Pages are written in Markdown text markup. The skill level required to work with these pages is minimal. If you can use a text editor, you can write docs with Markdown.

2 Content Author

The easiest way to help the docs revitalization effort is by authoring content. The content author would (as the title suggests) write content to be hosted on the OpenIndiana Docs website. Content can include just about anything, news posts, tutorials, Howto's, or larger efforts such as the handbook or updating the OSOL books.

> ⓘ **NOTE:**
> The nice thing about helping the project as a content author is the level of commitment is entirely up to you the individual. In other words, there is no requirement to get involved in anything major. In fact, it would be better to start small and only later consider taking on something larger.

Helpful skills include a willingness to write and work as a team with content reviewers.

3 Content Reviewer

The content reviewer wears multiple hats and effectively performs the roles of:

- Editorial reviewer
- Configuration manager
- Release manager
- Doc team leader/trainer.

This person (or persons) would review pull request and provide editorial feedback to the content authors. Helpful skills include diplomacy, good organizational abilities, as well as a knack for performing editorial reviews.

4 The Content Review Process

Upon receipt of a pull request, the content reviewer performs the following steps:

- Determines whether the changes are suitable as they are or require additional modifications (providing feedback to the originator as necessary).
- If changes are extensive or involve inline html, the content reviewer may check out your pull request (PR) to perform QA testing.
- QA testing is performed by running MkDocs in development mode.
- Upon acceptance, merges the pull request into the mainline branch.
- Upon merge, the site is automatically published via GitHub Actions.