Getting Started with OpenIndiana Docs

Contents

The process for contributing to OpenIndiana Docs is simple and follows the same best practices used in the development of the OpenIndiana distribution.

> ⓘ **NOTE:**
> First and foremost, we understand using development tools and techniques to write end user documentation is not for everyone. For this reason, we're happy to accept your contribution in any format you wish to provide.
> After receiving your submission, we will review the document for conversion to markdown and subsequent incorporation into OpenIndiana Docs. Contributions may be submitted in plain text, .doc, .docx, .odt, html, xml, latex, pdf, GitHub Gist, etc.
> In summary, if you believe your contribution would be helpful to the greater OpenIndiana community, we'll be willing to review it. For further information, please contact us via one of the methods provided below.
> To make a suggestion or report a problem with a document, please make your request by submitting a Github issue.
> The docs team can be reached via email: ***docs at openindiana.org***.
> You may also inquire via IRC:
> - #openindiana (libera.chat)
> - #oi-dev (libera.chat)

1 Prerequisites

If you would like to make a contribution using the development tools, you must have a GitHub account. If you do not already have a GitHub account, sign up for a Github account.

1.1 Install and configure Git

Operating System	Command
Arch	`pacman -S git`
Centos/RHEL	`yum install git`
Debian/Mint/Ubuntu	`apt-get install git`
Fedora	`dnf install git`
OpenIndiana	`pkg install git`

> ⓘ **NOTE:**
> After GIT is installed, be sure to configure your name and email address.
> For further details about configuring GIT, see: https://help.github.com/articles/set-up-git/
> Also, if you need a good book for quickly getting up to speed with GIT, see here: https://git-scm.com/book/en/v2
> Fear not though, only basic git commands are required for working with OI-DOCS.

1.2 Install python-pip

If your OS doesn't provide pre-packaged mkdocs, you'll need to install python-pip. This is not necessary on OpenIndiana.

Operating System	Command
Arch	`pacman -S python-pip`
Centos/RHEL	`yum install python-pip`
Debian/Mint/Ubuntu	`apt-get install python-pip`
Fedora	N/A - (Included with Fedora 25)
OpenIndiana	Not needed - (Use MkDocs IPS package)

1.3 Install rubygems

If your OS doesn't provide pre-packaged mdl, you'll need to install Ruby Gems. This is not necessary on OpenIndiana.

Operating System	Command
Arch	`pacman -S ruby`
Centos/RHEL	`yum install rubygems`
Debian/Mint/Ubuntu	`apt-get install rubgems-integration`
Fedora	`dnf install rubygems`
OpenIndiana	Not needed - (Use mdl IPS package)

1.4 Install mkdocs

For OpenIndiana Hipster, MKDocs and all of it's dependencies have been packaged and are available in the OI Hipster repository. So, if you're already running Hipster, installing MKDocs is as simple as: `pkg install mkdocs`. However, the MKDocs Material theme may need to be installed additionally using `pip`.

- Most other operating systems: `pip install mkdocs-material`

After MKDocs has been installed, be sure to verify your installation with `mkdocs --version`

> ⓘ **NOTE:**
> If you experience difficulties installing mkdocs, try using the python 3 version of `pip`. For example:
>
> - For Linux Mint 18, the python 3 version of `pip` would be `pip3`.

1.5 Install Markdown Lint (mdl)

On OpenIndiana you can install mdl using standard package management tools:

- `pkg install developer/documentation-tool/mdl`

For most operating systems:

- `gem install mdl -v 0.4.0`

1.6 Install Markdown plugin for VIM or Emacs (optional)

- https://github.com/plasticboy/vim-markdown/
- https://www.emacswiki.org/emacs/MarkdownMode

1.7 Install Pandoc & XeTex (optional)

For converting the docs to PDF, OI-Docs requires a Pandoc version of 2.8 (released 2019) or above and XeTex/XeLaTex. The latest versions of Pandoc are available (in .deb & .tar.gz) from Pandoc GitHub releases. Pandoc is not yet avalible packaged on OpenIndiana.

Operating System	Command
Arch	`pacman -S pandoc`
Centos/RHEL	`yum install pandoc texlive-xetex`
Debian/Mint/Ubuntu	`apt-get install pandoc texlive-xetex`
Fedora	`dnf install pandoc texlive-xetex`
OpenIndiana	Not packaged yet

> ⓘ **NOTE:**
> Not all OS distro repositories contain XeTex. In this case it could be installed as part of a manual TeX Live installation.
> As of Summer 2021, the version of Pandoc available from OS distro repositories may be too old (was the case for Centos/RHEL and Debian < 11). In this case you can try and use the Pandoc GitHub release versions instead.

2 Fork the OpenIndiana Docs repository

- Open your web browser to the OpenIndiana Docs GitHub Repository.
- Click the **Fork** button found in the upper right hand corner of the page.
- Forking creates a server side clone of the upstream repository.
- This clone is your own personal copy of the OpenIndiana Docs repository.

3 Create a local clone of your fork

```
git clone https://github.com/YOUR-USER-NAME/oi-docs.git
```

The local clone is where you will perform your work. Think of your local clone as your working copy of the repository. The local clone is also where you will commit your changes. Periodically you will push these local changes to your fork residing on Github.

4 Add the upstream repository

```
cd oi-docs
git remote add upstream https://github.com/OpenIndiana/oi-docs.git
```

Verify things with `git remote -v`

You should now see:

```
origin  https://github.com/YOUR-USER-NAME/oi-docs.git (fetch)
origin  https://github.com/YOUR-USER-NAME/oi-docs.git (push)
upstream        https://github.com/OpenIndiana/oi-docs.git (fetch)
upstream        https://github.com/OpenIndiana/oi-docs.git (push)
```

5 Sync and merge changes from the upstream repository

Periodically you will want to *rebase* your local copy by bringing in changes from the upstream repository. In plain English, this means the upstream repository is locally registered so you can periodically *pull down* changes from the upstream master repository and merge them into your local clone. This way your local clone remains in synchronization with the master upstream repository. It is always a good idea to perform a pull from the upstream master repository prior to making changes to your local clone (working copy).

```
git pull upstream master
```

6 OpenIndiana Docs site structure (master branch)

All site development occurs within the master branch.

```
oi-docs/
├── docs/
├── link_validator.py
├── makepdf.sh
├── markdownlint-rules.rb
├── mkdocs.yml
├── pandoc/
├── pdf/
├── README.md
└── site/
```

Resource	Description
oi-docs/	site root folder
docs/	documentation root folder
link_validator.py	URL validation script
makepdf.sh	PDF Generation script
markdownlint-rules.rb	Markdown Lint configuration
mkdocs.yml	Site and menu configuration
pandoc/	Pandoc configuration files folder (for makepdf.sh)
pdf/	Generated PDFs folder
README.md	Git readme
site/	Live preview folder (no edits)

> △ **CAUTION:**
> - Please do **NOT** perform any work within the `site/` or `pdf/` folder.
>   - `site/` is a temporary folder created by MkDocs when the site is run locally in preview mode.
>   - `pdf/` is a temporary folder for storing locally generated PDFs.
> - Also, please do **NOT** perform any work within the gh-pages branch.
>   - The gh-branch is destroyed and rebuilt each time the site is deployed to GitHub pages.

```
docs/
├── books/
├── community-hcl/
├── contrib/
├── dev/
├── favicon.ico
├── handbook/
├── index.md
├── misc/
├── Openindiana.png
├── release-notes/
└── retired/
```

Resource	Description
books/	Legacy OpenSolaris Books
contrib/	Contributor guidance docs
community-hcl/	Community HCL docs
dev/	Development oriented docs
favicon.ico	Site favicon icon
handbook/	OpenIndiana handbook docs
index.md	Site front page
misc/	Miscellaneous docs
Openindiana.png	OpenIndiana project graphic
release-notes/	Release Notes docs
retired	Deprecated docs, etc.

7 Make some changes

Open your favorite text editor and begin authoring content.

For example: `vim somefile.md` or `emacs somefile.md`

Some text editors (Atom, VIM, etc.) natively include Markdown syntax highlighting (or offer it as a plugin).

> ⓘ **NOTE:**
> Major changes should be performed within a separate branch, appropriately named to reflect the changes being made.

For a list of subject to write about:

- See the site TODO list.
- Have a look at the site Github issues list.
- See the Topics page for a list of suggestions and TODO's.
- Write a new tutorial, or complete a small section of the handbook, etc.
- Consult with other doc team contributors for even more ideas.

8 Adding new pages to the site menu

Site configuration settings and menus are located in `mkdocs.yml`. To add a page to the menu, simply add a line to this file. See the code snippet below for examples.

```
- Docs:
    - 'OpenIndiana Code of Conduct': misc/conduct.md
    - 'OpenIndiana FAQ': misc/faq.md
    - 'OpenIndiana Handbook': handbook/handbook.md
    - OpenSolaris Books:
          - 'Book Index': books/index.md
          - 'Basic Administration': books/basic.md
          - 'Advanced Administration': books/advanced.md
          - 'Solaris Express Administration': books/express.md
          - 'Trusted Extensions Administration': books/trusted.md
          - 'Development Titles': books/develop.md
```

9 Visualize your changes using live preview

To assist with the content authoring process, it may be helpful to visualize your changes using a live preview.

- From the root of the `oi-docs/` directory:
- Type: `mkdocs serve` and press enter.
- Open your web browser to 127.0.0.1:8000.

Each time you save your changes, the site page is automatically reloaded within your web browser.

To shut down the live preview web server, use `CTRL` + `C`.

> ⓘ **NOTE:**
> If you wish to preview your changes on a remotely networked system or on a networked mobile device such as a tablet, the site can also be served on your LAN IP address. To do so, issue the following command: `mkdocs serve --dev-addr=0.0.0.0:8000`

10 Running Markdown Lint (locally)

Markdown Lint is used to check your changes for Markdown syntax errors. Prior to submitting a pull request (PR), please consider running Markdown Lint locally on your computer.

From the root of the `oi-docs/` directory, execute the following command:

```
mdl -s markdownlint-rules.rb .
```

Markdown Lint will automatically traverse the entire folder structure looking for problems. Alternately you may also run `mdl` on a specific file. Simply replace the period (.) with the path to the file.

> ⓘ **NOTE:**
> Before you can run `mdl`, it may be necessary to modify your `$PATH` variable: see "Install Markdown Lint" above.

11 Generating PDFs (locally)

Pandoc (via XeLaTex) is used to create PDF versions of pages. The `makepdf.sh` bash script automatically moves through a subset of the docs directories creating a PDF for every Markdown file. Prior to submitting a pull request (PR), please consider running the Pandoc PDF generation script locally on your computer to ensure no Pandoc incompatibilities have been introduced.

To convert multiple pages to PDF, execute the following command from the root of the `oi-docs/` directory:

```
./makepdf.sh
```

This will produce PDF versions (placed in `oi-docs/pdf/`) of most of the doc pages. Not all pages are suitible for conversion yet, some contain incompatible HTML which needs to be fixed before they can be converted.

To convert a single file, the `-f` option can be used. Likewise, the `-d` option can be used for converting files all files in a given directory.

By default the PDFs are styled/formatted to look similar to the HTML docs produced by MkDocs. This can be changed using the `-s` option. Currently the only other option is 'opensolaris' which produces PDFs in the style of OpenSolaris Solbook documentation.

The formatting and restyling required for the PDFs is configured with files in the `pandoc/` folder of the root directory. Each style uses a pair of files, a `.yaml` file to specifiy Pandoc settings & LaTex/XeLaTex headers and a `.lua` Lua script to apply formatting to the Pandoc native representation (AST). For more information on Pandoc, see the official Pandoc Manual.

> △ **CAUTION:**
> Inline HTML is not supported by Pandoc for PDF output and is not rendered at all in PDFs produced. Ensure that you only use Markdown in your edits. The only exception is for the mkdocs `div` elements used to create these 'Note' breakouts - support for these has been manually programmed into the Pandoc lua filters.

12 Enabling spell checking in vim

You can invoke spell checking in your current session by inputting the command:

```
:set spell spelllang=en_us
```

If you would like a more permanent solution, enable spell checking in your `.vimrc` by adding the line:

```
set spelllang=en_us
```

Misspelled words will now be highlighted (color varies dependent on your .Xresources file)

12.1 Keyboard shortcuts for spell checking in vim

- `]s` to find the previously misspelled word.
- `[s` to find the next misspelled word.
- With the cursor at the beginning of a word, use `z=` to bring up a list of suggested replacements.
- `zg` will add a word to the local dictionary exception file.
- `zw` is used to mark a word as incorrect.

13 Commit and push your changes.

```
git status
git add some-file-you-just-edited
git commit -m 'your commit message'
git push
```

When you make a commit, you are committing those changes locally to your clone. When you perform a push, your are pushing your commits from your local clone to your fork residing on Github.

14 Send a pull request.

- Open your web browser to ***your forked copy*** of the OpenIndiana Docs repository.

For example: `https://github.com/your-user-name/oi-docs`

- Click the button for *New pull request*.
- Add some notes about your change.
- Submit your PR (pull request).

Pull requests are used to request a *pull in* of changes from your fork to the master repository.

> ⓘ **NOTE:**
> After a pull request has been submitted, and for the duration of time your pull request remains open and uncommitted to the OI-DOCS master repository, any additional commits you make to your own fork of the oi-docs repository will automatically be included in your open pull request.

15 What happens next?

At this point a member of the OpenIndiana Project docs team will review your changes. If no corrections are required, your changes will be accepted and merged into the upstream repository.

Upon commit, publishing occurs automatically using Travis-CI.