Note: This is a draft, and some sections are under development.

About me

Name: Nikhil Ramakrishnan Email: ramakrishnan.nikhil@gmail.com GitHub Profile: https://github.com/nikramakrishnan

Gitter: nikramakrishnan

Institute: Bennett University Degree: B.Tech. Computer Science & Engineering (Class of 2020) Location: Delhi NCR, India (GMT +5:30)

Location during Summer Break: Mumbai, India

Redevelop FreeType's documentation using markdown

Documentation is an essential part of FreeType which provides developers access to and information about the API.

Current State

Global Documentation

The global documentation is directly written in HTML, which is inconvenient and not very flexible.

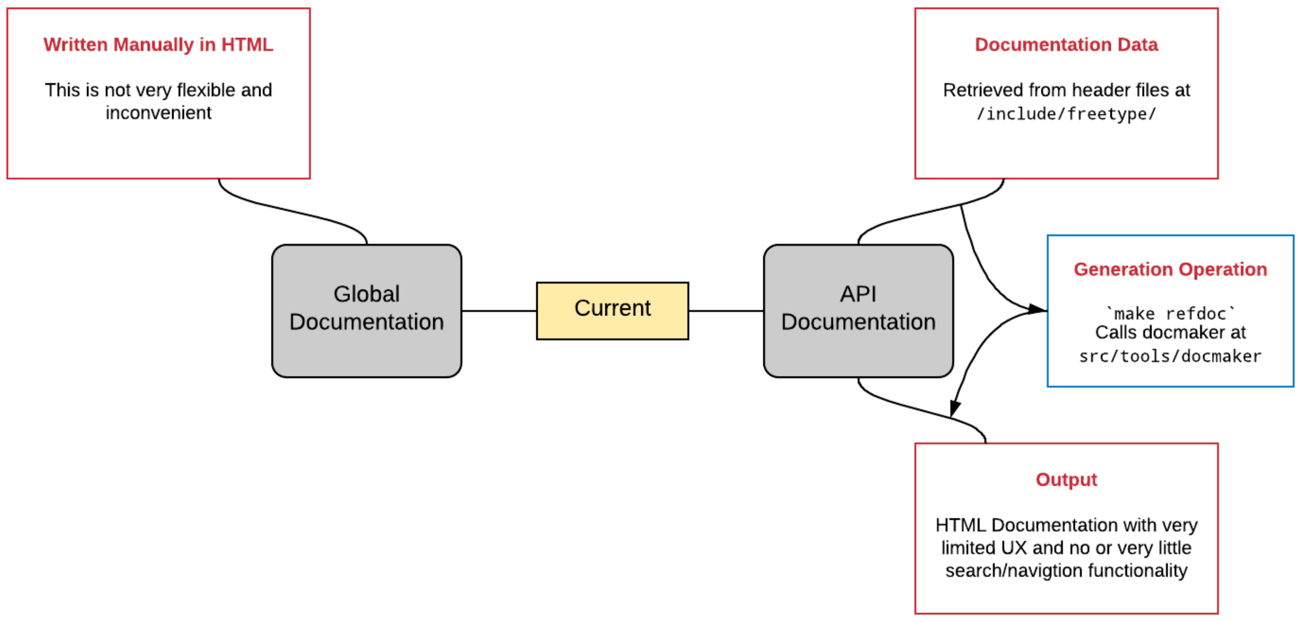
API Documentation

API Documentation uses a Python script called `docmaker` that converts a very limited, markdown-like syntax to HTML.

Aim of my project

The aim of my project is to redesign the documentation methods for FreeType, and make it easier to update and manage the FreeType website. This includes changes in the current method of retrieving and generating the API documentation, one-time conversion of the current pages to Markdown, and deploying the website from a repository by performing Markdown to HTML conversions. This is explained below in detail.

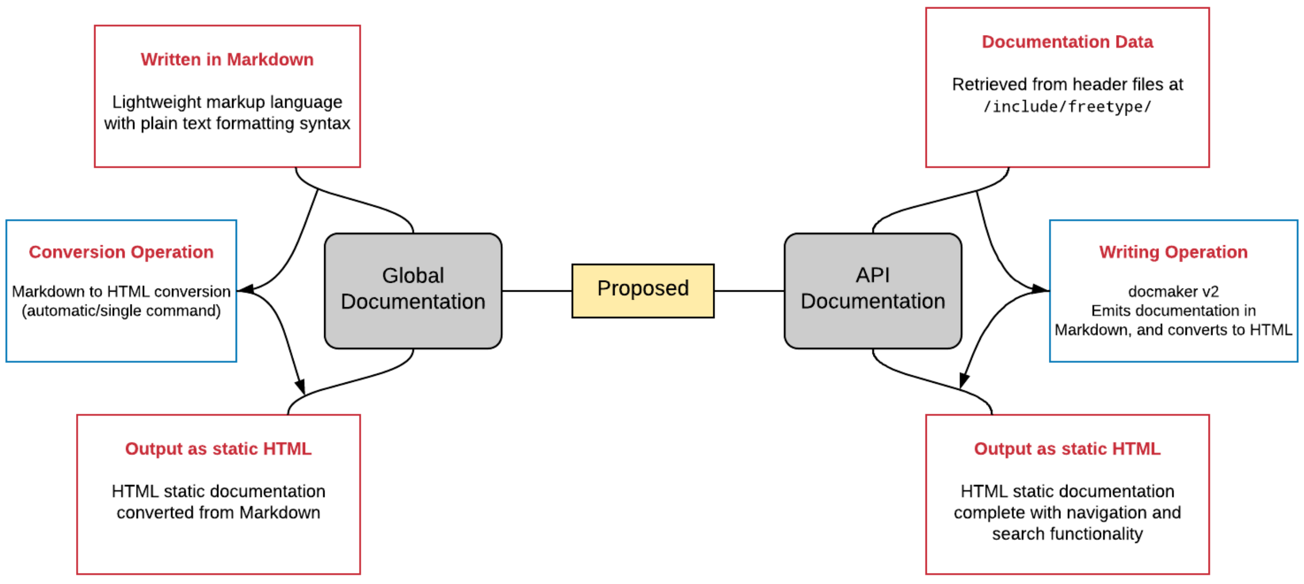
Current Functioning


Written Manually in HTML
This is not very flexible and inconvenient
Global Documentation
Current
API Documentation
Documentation Data
Retrieved from header files at /include/freetype/
Generation Operation
`make refdoc` Calls docmaker at src/tools/docmaker
Output
HTML Documentation with very limited UX and no or very little search/navigtion functionality

This diagram shows how the current website and documentation works. There are a few issues with this:

1. **Manual Webpage Writing**: Pages need to be written in HTML manually. This makes creating and modifying content might discourage contributors from writing guides/tutorials.
2. **Navigation and Search** has to be implemented manually for each page currently.
3. **API Documentation** uses a limited Markdown-like syntax which is directly converted to HTML by `docmaker`.
4. The documentation has a **very limited** UX and does not have search functionality.
5. `docmaker` is **not documented** and might cause difficulties when contributors are trying to fix bugs or add features.

Post-project Functioning


Written in Markdown
Lightweight markup language with plain text formatting syntax
Conversion Operation
Markdown to HTML conversion (automatic/single command)
Global Documentation
Output as static HTML
HTML static documentation converted from Markdown
Proposed
API Documentation
Documentation Data
Retrieved from header files at /include/freetype/
Writing Operation
docmaker v2 Emits documentation in Markdown, and converts to HTML
Output as static HTML
HTML static documentation complete with navigation and search functionality

This diagram shows how I plan to implement the website and the new documentation. These are the milestones:

1. Investigating how to convert the complete documentation and website to markdown using an automated approach
2. Extending `docmaker` to emit markdown for the API Documentation
3. Making sure that cross-references resolve in the documentation
4. Set up a Makefile (or equivalent) that statically generates the complete HTML code of the FreeType API documentation.
5. Implementing a method to update the FreeType website directly from the repository.
6. adding automatic generation of navigation bars and search to the webpages

Benefits to Community

This project will upgrade the current FreeType documentation to Markdown, a highly portable and lightweight markup language. This has a lot of merits: - No need to manually write HTML for website and documentation - Cutting-edge documentation standards - Markdown documentation is used by many open-source projects, and there is active development towards representing markdown documentation in beautiful HTML static websites (pandoc, MkDocs). - This will make it easier for contributors to write guides and tutorials on the website. - This project can be extended to other open-source projects: to extract documentation, write to markdown, generate a website all with a single command.

Timeline

Week -2 to 0 (April 23 to May 14) - **Community Bonding** - Interact with the mentor and the FreeType developer community.

Week 1 & 2 (May 14 to May 27) - **Inspect docmaker and create roadmap** - Inspect the current `docmaker`, and evaluate its usability and extendibility. - Finalize the evaluations and document a roadmap for docs generation from source to emit markdown, and conversion of the same to a static site. - Begin implementation of modified `docmaker`

Week 3 & 4 (May 28 to June 10) - **API goes Markdown!** - Implement markdown API documentation generation from header files. - Making sure that cross-references resolve in the documentation - Test and evaluate converters to static websites and compare outputs of different HTML converters.

Week 5 & 6 (June 11 to June 24) - **Markdown to HTML** - Select and implement an HTML converter for API Documentation. - Make sure everything works as expected, and implement navigation and search functionality.

Week 7 (June 25 to July 1) - **Static Conversion of Website** - Evaluate and test different options for converting current website and global documentation (except API documentation) to markdown (one-time), such that it can be easily integrated into the website. - Write code for the same and perform few conversions.

Week 8 & 9 (July 2 to July 15) - **MakeFiles and Website Updating -** Set up a method to update website by generating HTML from Markdown uploaded to a repository. - Set up a `makeFile` to generate complete HTML pages for the API Documentation, to be included in the distribution tarball. - Community feedback and review for the module.

Week 10-13 (July 16 to August 6) - **Final work, bug-fixes, suggestions -** This period would be dedicated to polish things, fix bugs, taking feedback from community and implementing suggestions. - Finishing up everything, and deploying the documentation to website. - My classes for the next semester would begin on July 23rd. This could decrease my contribution during the remaining period. - This would officially conclude my work for FreeType Documentation in GSoC. I would continue to contribute to the project, as and when possible.

Experience

I have experience in C/C++, Python, PHP. I have worked on many projects till date.

Some of my notable projects are: 1. **Visitor Management System**: A complete management system for visitors on a campus, with JSON API endpoints. GitHub Link: https://github.com/nikramakrishnan/visitor-management Documentation Link: https://github.com/nikramakrishnan/visitor-management/wiki 2. **System Cleaner**, a handy tool that organizes folders, and cleans up temporary files. GitHub Link: https://github.com/nikramakrishnan/system-cleaner 3. **Octave**, Machine learning based musical instrument detection. 4. I also have been solving **competitive programming** problems on CodeChef as a hobby. GitHub Link: https://github.com/nikramakrishnan/codechef-solutions

Skill Set

Programming Languages: C/C++, Python, Java, PHP, MySQL, Node.js **Front-end**: HTML, CSS, JavaScript

List of Documentation Converter Candidates (not exhaustive)

This list is compiled from the GSoC ideas list and my own searches.

- Pandoc (https://pandoc.org/)
- Jekyll (https://jekyllrb.com/)
- MkDocs (http://www.mkdocs.org/)
- Couscous (http://couscous.io/)
- Hugo (http://gohugo.io/)