

GSoD 2022 Project Proposal

ClangREPL - Improving Analysis Tool Documentation for the HSF

Mentors: **David Lange, Vassil Vasilev**

Abstract

A User documentation is designed to assist end users to use a product or service. Good user documentation is very important because it provides an avenue for users to learn how to use a software, its features, tips, tricks and also resolve common problems encountered when using the software. It also reduces support cost and is part of the corporate identity of the product: a good user documentation is a sign of healthiness of the product, the developer team.

Without a good user documentation, a user may not know how to do the above things effectively and efficiently. User Documentations can play a pivotal role in ensuring a product's success because great communication is and will always be at the heart of any business or product and a great documentation just takes that communication and puts it in a manageable framework that everyone can access for success.

As we know Particle physics relies heavily on open-source codes that compute key quantities for both theory and experiment. Often such codes have been developed for many years, with early documentation rendered incorrect or incomplete by later developments. Key user-facing information can be very difficult to find (e.g. in automatically generated code interface documentation). It can be difficult for new physics-oriented users and contributors to get involved: the docs that exist are more for the developers and often assume too much technical prowess.

So here I will be focused on interactive analysis based on C++ and my intent will be to build the sort of documentation that enables users engagement while being easy to update as our codes continue to evolve.

HEP researchers have developed several unique software technologies in the area of data analysis. Over the last decade they developed an interactive, interpretative C++ interpreter (aka REPL) as part of the ROOT analysis project where they invested a significant effort to replace CINT, the C++ interpreter used until ROOT5, with a newly implemented REPL based on LLVM-Cling. Cling is a core component of ROOT and has been in production since 2014. Cling is also a standalone tool. It is recognized for enabling interactivity, dynamic interoperability and rapid prototyping capabilities for C++ developers. Now after advancing the implementation and generalizing the usage we need to improve the overall documentation experience in the area of interactive C++.

Scope

In this project I will audit the existing documentation for the Clang-REPL (interactive C++) and identify the gaps in the information or presentation from the point-of-view of new, physics-oriented users and incorporate feedback from developers in the project and the community.

The scope of the interactive analysis component continues with:

- Demonstrate interactive use-cases of Clang REPL in the LLVM documentation by developing basic documentation and tutorials apt for Clang REPL
- Develop advanced Clang REPL tutorials and integrate Clang REPL into Xeus and contribute to a community blog post on them
- Review and enhance the developers documentation, examples and tutorials

Why is your proposed user documentation an improvement over the current one?

The proposed user documentation will be structured to improve and ensure efficiency, consistency, and peace of mind for any end user. It will contain written guides, tutorials & blog posts.

Analysis

For the documentation we can use **Sphinx** or **Read the Docs**. I researched a lot about these tools to get a better understanding of how it works.

Sphinx

Sphinx is a robust and mature solution for software documentation. It includes a number of features that writers expect, such as:

- Single source publishing, content reuse through includes, conditional includes based on content type and tags.
- Multiple mature HTML themes that provide a great user experience on mobile and desktop
- Referencing across pages, documents, and projects
- Index and Glossary support
- Internationalization support

It uses reStructuredText as its markup language, and many of its strengths come from the power and straightforwardness of reStructuredText and its ability to translate documentations.

Read the Docs

Read the Docs simplifies software documentation by automating the building, versioning, and hosting of your docs for you.

- **Never out of sync**

Whenever you push code to your favorite version control system, whether that is Git, Mercurial, Bazaar, or Subversion, Read the Docs will automatically build your docs so your code and documentation are always up-to-date.

- **Multiple versions**

Read the Docs can host and build multiple versions of your docs so having a 1.0 version of your docs and a 2.0 version of your docs is as easy as having a separate branch or tag in your version control system.

- **Free and open source**

Read the Docs is free and open source and hosts documentation for nearly 100,000 large and small open source projects in almost every human and computer language.

Verdict

Sphinx is an incredibly powerful tool. Read the Docs builds on top to provide hosting for Sphinx documentation that keeps your docs up to date across versions. Together, they are a wonderful set of tools that developers and technical writers can use to create user documentation that would be best for end-users.

Sphinx includes support for **translating** documentation into **multiple languages**. It supports version control which will be used to manage the documentation. Unlike the current wiki where anyone can edit and not add the right information, this version control system would ensure that all changes are reviewed first before it is merged into the main repository. Version control will also make the documentation increase open source contribution to the project because people can create issues, open pull requests, etc. Sphinx and read the Docs are used by a list of other great communities such as; ASP.NET, Kernel, Julia, Jupyter, PHPMyAdmin, Write the Docs, etc and it can be a great tool to use for the new user documentation.

Success Metrics

The Interactive Analysis Development success metrics include:

1. Publishing at least 2 blog posts on <https://blog.lvm.org>

2. Publishing at least 3 LLVM help document in <https://clang.llvm.org/docs/>
3. Publish at least two tutorials about more advanced Clang REPL features at <https://compiler-research.org/tutorials/>
4. Reviewing and improving developer's documentation
5. Restructure the documentation.
6. Update the documentation to fit the modern versions of Clang REPL.
7. Rewrite the user documentation to make it easier to understand.
8. We can make it community driven so users can report any issues of the documentation and also be able to make changes that will be reviewed and merged by the Clang REPL team.

Timeline

As described already, the interactive analysis technical work will nominally continue through October but as I mentioned earlier from **June to Mid September I will be available 30-40 hours every week and according to our discussion 5-6 hours a day for the documentation can be enough but still if I do not achieve my goals then extra efforts will be put from my side.** And I will have my exams in the last week of September so I will need of two weeks to focus on exams and during this time I will be available for 4-5 hours per day(20-25 hours per week) but most of our main work will be ported before this time and I will have to focus on only writing the blog post. So to write the blog post that time will be sufficient according to my understanding.

And, now for the month of October again I will be available for 30-35 hours. So if anything left during my exams then I will easily compensate during this month because I will have one week buffer. And, I have one more plus as you can see from my Past experience I have worked remotely for more than 1 years at different organizations so I can manage my time easily because I am a fast learner :)

Dates	Activities & Goals	Time Commitment
June	<ul style="list-style-type: none"> - Orientation & Documentation Audit - Connect with mentor/mentors - Finding suitable hours to communicate with mentor/mentors as our timezones are different - 1 week for meeting the team - 1 week for getting into relevant technologies - 1 week for environment setup - 1 week for documentation audit <p>Note: The above weeks will not complete time so</p>	30-40 hours every week

	remaining time will be dedicated to auditing the documentation	
July	<ul style="list-style-type: none"> - Demonstrating interactive C++ use-cases in the LLVM documentation by developing basic documentation and tutorial apt for Clang REPL - The buffer time will be dedicated to setting up a meeting with Developers to know their review and persepective 	30-40 hours every week
August	<ul style="list-style-type: none"> - Develop Advanced Clang-Repl tutorials - Integrate Clang REPL into the XEUS - Have a communication with the Ceus-Cling team and possibly porting the kernel to LLVM - If time remains then it will be dedicated to complete some tasks of September month as I will have my Mid Semester exams during the last week of September 	30-40 hours every week
September	<ul style="list-style-type: none"> - Write a blog post on a working notebook demonstrating a tutorial and it should organised in a proper way - Review and enhance the developer documentation - Add examples & Tutorials 	<ul style="list-style-type: none"> - 25 - 30 hours every week for 1st two weeks - 20 - 25 hours for last two weeks
October	<ul style="list-style-type: none"> - Buffer time will be used to complete the task left in the September month - Integrate & Validate the written documentation - Audit and Outline further work 	- 30 - 40 hours every week

During June

- Become more familiar with the user documentation and current version of Clang REPL.
- Learn new techniques and technical writing skills which will be helpful while implementing the project.
- Set up a communication channel and time (due to time difference)
- Refine my goals and set expectations on both sides

- Learn more about the community and product
- Discussing the proposed documentation structure with the mentors and other members of the organization.
- Finalize the proposed features and other modifications that need to be made to the existing documentation.

Documentorial Contributions to Open Source Organizations.

- <https://github.com/JuliaData/DataFrames.jl/pull/2700>
- <https://github.com/JuliaData/DataFrames.jl/pull/2703>
- <https://github.com/JuliaData/DataFrames.jl/pull/2723>
- <https://github.com/JuliaData/DataFrames.jl/pull/2755>
- <https://github.com/JuliaData/DataFrames.jl/pull/2717>
- <https://github.com/SciML/NeuralPDE.jl/pull/210>
- <https://github.com/SciML/Surrogates.jl/pull/218>
- <https://github.com/SciML/Surrogates.jl/pull/196>
- <https://github.com/SciML/Surrogates.jl/pull/148>
- <https://github.com/SciML/NeuralPDE.jl/pull/86>
- <https://github.com/SciML/Surrogates.jl/pull/158>
- <https://github.com/SciML/Surrogates.jl/pull/179>
- <https://github.com/SciML/Surrogates.jl/pull/180>
- <https://github.com/SciML/Surrogates.jl/pull/183>
- <https://github.com/SciML/Surrogates.jl/pull/196>

These are some of my PRs which I have made in the Julia Community.

Why this project?

I have always had a conviction that writing code, solving problems and building software reaches its full potential when you also have the ability to enlighten others about it through writing. Personally, I have always been fascinated by the efforts undertaken by the CERN HSF community. It will be an honour to work with the community.

Why am I the right person for this project?

I believe myself to be the right person for this project because

- I have past experience in improving the documentation of organizations and I can use any version control system, so carrying out commands on GitHub/GitLab will not be a problem. Moreover, what drives me is working on projects that create value to people.
- I believe that If you want someone to do something in the most efficient way possible, you document it. By documenting your processes, you ensure efficiency, consistency, and peace of mind for anyone involved.
- I know the needs of the any software documentation. This will enable me to write the documentation in such a way that every other user across the globe understands at first glance.

Personal Information

- **Name:** Rohit Singh Rathaur
- **Github:** <https://github.com/RohitRathore1>
- **Blog:** <https://www.linkedin.com/in/rohit-singh-rathaur/recent-activity/posts/>
- **Resume:** https://rohitrathore.netlify.app/rohit_resume.pdf
- **Location:** Ranchi, Jharkhand, India
- **Time Zone:** GMT +5:30
- **Typical working hours:** 4PM - 11 PM (GMT +5:30)

References

- [ReadTheDocs](#)
- [Sphinx](#)