

Documentation workshop whiteboard

[Indico agenda](#)

Contents:

[Advance input](#)

[Workshop live notes](#)

[Existing doc tools and services in ATLAS - Maria Smizanska](#)

[Jekyll - Adam Barton, Alexey Boldyrev](#)

[Doxygen - Adam Barton, Alexey Boldyrev](#)

[IT documentation and training tools - Maria Dimou](#)

[Modern documentation tools and approaches - Giordon Stark, Mario Lassnig](#)

[Other experiments - Ed Moyse](#)

[Documentation links of interest](#)

[Workshop report in S&C Week Dec 15 2018](#)

[Post-workshop notes and actions](#)

[Nov 15 workshop announcement](#)

The documentation toolset (both narrative and code documentation) has been growing rapidly in the open source world. This workshop is to help ATLAS identify promising directions and tools to pursue, within the context of preserving adequate stability, continuity and preservation of investment in the existing documentation.

This whiteboard is for gathering live notes during the workshop, guiding the discussion periods, and for gathering advance input. Mario Lassnig and Giordon Stark will be curating the advance input and combining it with their own

Advance input

Comment/suggestions are on in the document. **Please feel free to add your comments to the list below.** We're interested in documentation tools, approaches you think we should (or should not) be using based on your experience and knowledge of the documentation toolset and ecosystem. Or, send your input to Mario Lassnig (mario.lassnig at cern.ch) and/or Giordon Stark (gstark at cern.ch) so they can add it here.

- I think if we have a documentation improvement days for few days and ask people to update twiki information in their relevant area that people will be willing to improve the documentation. Years ago we did that and I saw some good work done then. So i suggest we select few days and do it again. (Emil Obreshkov)
 - (Giordon) Agreed, similar effort has been done elsewhere:
<https://www.eventbrite.com/e/bids-docathon-kickoff-tickets-32302896834?aff=mcivte>
- Some input from me, on LXR. I admit bias because I implemented it in ATLAS, as in BaBar before ;-) Not bias that LXR should live forever but that we should continue to have the functionality it supports. BNL has continued to maintain it because there has never been anything found by me or Alex that covers what it does well, namely browser based navigability (including hyperlinked header files) and searchability of the code. Graeme and Walter wanted to officially get rid of it in the SIT workshop a while back; US ATLAS subsequently asked BNL to continue to support it because at least a substantial number of US developers want it supported. If you guys were able to clarify

whether we're able to retire LXR, replacing it with something equivalent/better, or in the absence of a replacement that we should continue to support LXR, it would be helpful. (Torre Wenaus)

- As someone who is just starting out with ATLAS software, I have found LXR to be invaluable. Its language-aware search and ability to click through the source code and see definitions and usage are very useful. Please do not retire it unless it is replaced with something that is equivalent or better. (OpenGrok?) (Justin Chiu, UVic)
- Are you going to debate on the existing (Twiki, web cds, git whatever) and provide maybe better solutions if necessary. (Fairouz Malek)
- Wow CDS we will not touch :-) But around the edges of the huge investment we have in twikis, which of course we must preserve, yes we're interested to discuss better solutions for documentation going forward. (Torre Wenaus)
- My question was basically because I wasn't sure in which part of documentation you would like to focus on this workshop. I have personally found documentation in ATLAS to be a bit outdated in terms of tools and I was glad to see that there is an effort going on to improve this. I am not sure if this documentation was only for software tools or for everything in ATLAS. For instance as far as I know Twikis have their own syntax which have been around for a very long time and I remember having a conversation with someone in IT some time ago mentioning that twikis might get an update. For software tools one could use things like sphinx (python), rdocs (rby) or doxygen for almost anything to make automatic documentation compilable for instance within Gitlab but since the rules require to use Twikis this means that this documentation is not supported since there isn't, as far as I know, automatic deployment to twikis. (Juanpe Araque)
- Our focus in the workshop is on software and computing but the management has shown interest in what we're doing and in what conclusions we come to about the documentation tools in general (in particular twiki). Sphinx is getting a lot of attention, also the gitlab analog of github Pages which uses Jekyll, and we use doxygen already, but these are isolated islands around the perimeter of a rather aged twiki core, I hope we can emerge from the workshop with a slate of modern tools we endorse and integrate well. (Torre Wenaus)
- In general there is a lot of documentation, but some structure/organization is missing in some cases and information spread through many places (for most projects in ADC).
 - Twiki: except for the links I have bookmarked, I don't know how to quickly find anything (maybe it's just my problem).
 - A lot of documentation is moving away from Twiki (e.g. Rucio read the docs, Harvester is using GitHub wiki, or new pages hosted in GitHub or others), so you really need to know what is where
 - A lot of information can only be found in presentations
 - Different documentation for different users: developers, ops, shifters, users - who maintains each?
 - Generally there is a mix of outdated and up to date documentation
 - Are we using the best tools? (Fernando Barreiro)
- All XRootD documentation is maintained on MS Word (a holdover from the early days). Documentation is generally written before the features being documented are released (though sometimes we mess up). It is then converted to html and pdf formats and posted on the XRootD website. Since changes in documentation may be hard to track, each document's publish date is specified and the last section of the document lists all of the changes made since the last time it was published. An area of improvement would be to allow other software contributors to add to the documentation which is, now, not possible. On the other hand, doing so may create a disjointed hard to read document simply because it contains too many writing styles. So, that tension has stymied any progress (btw - it's easier to get the writing style correct the first time than through editing existing text). The biggest area of improvement is to produce a "guide" document that is easier to read than all of the documents we have. This has been requested many times. However, each time we produce a test version of this, no one actually reads it. So, that has stymied are progress in doing a serious job of it. Some additional packages that are not part of core-Xrootd but related to ATLAS have document either in Twiki (such as FAX) or Github (rucio related plugin to Xcache, etc.)
- We have students coming for our institute for a short period of time, and want to use computing resource. I naively think that they could read our doc in Twiki. They couldn't and they didn't tell us, until we/they ran into trouble (Wei Yang)
- my observations in other projects is that, at the end of the day, the best way to guarantee having [almost] all pieces of documentation organized and mostly up to date is to have dedicated people whose one and only one job is that.

Usually are also the people in charge of outreach. They contact with all developers, and gather almost all the inputs from them. But they are in charge of making everything uniform, easily accessible (maybe keeping a global index), experts on tools and techniques for docs migration, etc. Just an idea.

- TWikis! We use them every day, but my wild guess is that more than 50% of them contain outdated documentation. One of the main issues is how to keep documentation up-to-date, be it TWikis or something else.
 - (i) We have an ATLASDOC tracker in JIRA. People should enter a ticket to identify areas that need improvement/freshening up. Requests would then be dispatched to the appropriate responsible.
 - (ii) We could have a button on the TWiki pages to report outdated documentation. If a user lands on an outdated page they would push the button and this is recorded. Statistics information from the buttons can be used by the documentation team to identify pages in much need of improvement and enter a JIRA request
 - (iii) do not use emails to report missing/outdated documentation. Use JIRA so the dashboard can be used to track the workload (Davide Costanzo)
- I would like to comment about organisation of our documentation and how/if we can improve education. The reality is that about $\frac{1}{3}$ of ATLAS authors are students, and every year we have 200+ new students joining the collaboration and staying for 3-4 years. The tutorials are great, but it would be good to make sure that the documentation beyond the tutorial is also well organised, to cover many of the tools that we use. A suggestion is to have a curated collection of links to documentation tutorials for the open sources tools we use could be useful (git, python, C++, statistics, ...). In the end universities are responsible for their students' education, but we should think how to best use an ATLAS-wide system. (Davide Costanzo)
 - In addition to links to tutorials on open source tools, I think just one page in the tutorial with instructions on how to use LXR, search JIRA, PAThelp, etc. would be helpful.
 - And how about a C++ tutorial following the software tutorial? Ideally one for beginners and one for more advanced users. With the plethora of online courses out there, maybe this is a moot point. (Justin, UVic)
- Some analysis or development workflows could be captured in Jupyter notebooks, so to embed the documentation in the notebook. (Davide Costanzo)

Workshop live notes

Existing doc tools and services in ATLAS - Maria Smizanska

- ATLAS has 8800 twiki pages
- Twiki search has had work done to improve it but still gets complaints, more improvements needed
- CERN not in favor of buying a Google license to use Google search
- Indico and CDS use a search setup that is separate from twiki (and seems to work better than twiki search)
- Proposal to have JIRA tickets to manage computing doc. A ticket for each area.
- Documentation stand-down in early 2018?
- Is there a list/structure showing archived pages? Everything not touched since 2012 has been archived.
- 50 most recent changes in ATLAS twiki: <https://twiki.cern.ch/twiki/bin/view/Atlas/WebChanges>

Jekyll - Adam Barton, Alexey Boldyrev

- There is no intent that the Jekyll based software documentation replace the twiki
- About the twiki vs Jekyll pros/cons - Maria Dimou / IT
 - In our documentation investigation <https://twiki.cern.ch/CDAGroup/CDAPublicDoc> and [a comment in today's notes](#)

Doxygen - Adam Barton, Alexey Boldyrev

- Question about how many people use doxygen, and if not that many then what could be done to improve this?
- Scott mentioned that he never looks at the webpages, but the incode documentation is very useful
- Requests to have doxygen added more prominently (e.g. in tutorials)

IT documentation and training tools - Maria Dimou

- Check out the many links in the talk, e.g. [existing videos on git and doc tools](#) (mostly youtube)
- Trend is in the direction of Markdown
- host in gitlab if CERN-only, github if the audience is wider
- Sphinx for developers due to good code rendering
- GitBook is commercial and restrictive. No more than 5 co-editors
- MkDocs looks interesting, can convert GitBooks to MkDocs
- See the [notes](#) on their doc tool evaluations.
- Decided to try Jekyll.
 - Decided against <http://gohugo.io/about/> primarily because Jekyll was known (and themes not as attractive)
- recommendation: no sharepoint
- how about subtitles on the videos? - Good suggestion, will plan on that
- https://twiki.cern.ch/CDAGroup/CDAPublicDocTemplate#Notes_from_2017_12_11
- Our Jekyll-based new website in a screenshot due to permission problems
<https://twiki.cern.ch/twiki/pub/CDAGroup/CDAPublicDocTemplate/it-cda-dev-jekyll-site.png>
- Sign up to <e-learning-interest-group@cern.ch> to learn about new developments

Modern documentation tools and approaches - Giordon Stark, Mario Lassnig

- have a docathon documentation focus week?
 - Generally favorable. Can also try to drive doc activity with JIRA and milestones that are actively tracked.
 - Was done 6 years ago, seemed to work. Pizza Wednesday Nights maybe?
- Comment that suggestions on how to search (e.g. JIRA, lxr, gitlab) should be added to tutorials
- Comment about separation of concerns - developer-specific code should stay with the code in GitLab while physics-related stuff should be in twiki
- Breathe and doxygen - does it work well and quickly? Is it painful to add a second step?
 - No, I ran it on Athena doxygen and it took seconds to run
- How hard is it to get Breathe API functionality included? It doesn't seem to be there in xAODAnaHelpers example
 - Yes, you can go to <https://xaodanahelpers.readthedocs.io/en/master/API.html> and see the example of it working, but it requires specific extra rST functions, but it's not difficult at all
- Ask IT for gitlab pages support for open repositories
- Put up a JIRA ticket for allowing javascript such as diagramming packages in twikis
- LXR - absolutely invaluable because it gives a way to do fast syntax and click through and see the functions and how they are used. And if you just search in GitLab, even GitHub, you can do a little -- but still not as smart as LXR
 - Giordon: I brought it up because it seemed like LXR was getting harder to maintain
 - No, not at all, Very easy to maintain.
- What suggestions do you have for users? Do you want things in one document or something? What was your suggestion
 - Giordon: I think it's very useful for people maintaining documentation is that the code and documentation are all in one place
 - Do you want it more like workbook?

- Giordon: I don't know. That's up to you guys. I am just presenting some case studies of what people have been doing so far, or find better than existing solutions. Such as what Jet/MET has done.
- David Miller: One of the reasons that we tried to move to the paradigm you described in Jet/ETmiss is that we found that the people doing some of the lower level code could provide the technical documentation in GitLab near the code. But more conceptual documentation, if you will, why things need to be done a certain way, why recommendations for using physics objects in ATLAS are done this way or that way, results from those studies, etc... That doesn't belong in GitLab and that's something like a lot of the documentation in ATLAS that's provided by people who aren't software experts.
- David Miller: Twiki is used a lot in that regard for that reason, because that is really easy, can be directly edited. There aren't usually ten people trying to edit a twiki page at a time. The number of current updates is not a big deal. And you need something that graduate students doing their authorship qualification tasks all around ATLAS can do without learning something else.
- Giordon: So what you mentioned is what I like about SUSY tools on slide 16, SUSYTools has a separated their concerns. Their read me which is tied directly to GitLab is about very specific code related things such as using the tool or how to actually develop or contribute to it but the physics case that tends to follow the software release is in twiki where it is less tied and that's a nice thing to have.
- It's hard enough to document what we have. Forcing people to read documentation on making documentation is probably too much work. Something that is simple and easy-to-use, not a lot of check-ins or constantly creating merge requests -- reduce barrier to entry.
- Mario: Rucio documentation improved significantly because we made it public for the whole world to see. Don't want to look bad. If internal, less motivation to clean things up.
- Easier, within ATLAS to send an e-mail or have a look at a presentation. This doesn't translate well outside the community. This is where you find things in documentation to improve to not rely on email or internal slides.
- Diagrams are hard to add on twiki. Can we get javascript turned on? This is why some have moved on to using Google docs because it's easy to create/update diagrams and share.

Other experiments - Ed Moyse

- we should be building documentation for multiple releases in doxygen. And make the entry page nicer.
- LHCb's starter doc is extremely nice. The best. Lowers the threshold for new people.
 - Graeme later commented that the actual tutorials themselves aren't that great.
- Some others are doing better than us, quite a lot are doing worse than us in their documentation
- we should cross-reference between doxygen and other info sources (git, lxr, ...)
- can we have ~20 JIRA tickets covering the documentation domains, to collect issues, associate every twiki page in the hierarchy with one of these domains and hence JIRAs, and have a link on every twiki page to its JIRA ticket to add a comment -- and ask people when they see a page needing update to click and add a comment to the JIRA. Thus crowd source the QA and notification of outdated/wrong material
 - before the docathon, have an 'add issues' marathon to collect issues
 - specifically address the issues gathered in the docathon

Documentation links of interest

Please feel free to add your comments to the list below.

- Doxygen
 - ATLAS <https://atlas-sw-doxygen.web.cern.ch/atlas-sw-doxygen/index.php>
 - CMS <http://cmsdoxygen.web.cern.ch/cmsdoxygen/>
 - Breathe
 - <https://breathe.readthedocs.io/en/latest/> -- convert Doxygen into Sphinx-ready-input
- Twiki

- ATLAS S&C <https://twiki.cern.ch/twiki/bin/view/AtlasComputing/AtlasComputing>
 - CMS offline workbook <https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBook>
- [GitHub Pages](#), [Jekyll](#) and [Bootstrap](#)
 - HSF (github) <http://hepsoftwarefoundation.org/>
 - ATLAS software documentation (gitlab) <https://atlassoftwaredocs.web.cern.ch/>
 - CMS software (github) <http://cms-sw.github.io/>
 - <http://slateci.io/> github based OSG project
- GitLab Pages
 - <https://about.gitlab.com/features/pages/> -- lots of support for various documentation
- Pandoc -- (<https://pandoc.org/>) Convert documentation from one format to another
- MkDocs <http://www.mkdocs.org/> -- static site generator from Markdown sources
 - New OSG documentation site, migration of wiki to github/Markdown/MkDocs
<http://osg-docs.s3-website-us-east-1.amazonaws.com/>

[Workshop report in S&C Week Dec 15 2018](#)

Post-workshop notes and actions

Nov 15 workshop announcement

Hi all,

We would like to make you aware of the documentation workshop coming up during the next S&C Week, Monday morning December 11. The documentation toolset (both narrative and code doc) has been growing rapidly in the open source world. We want the workshop to help identify promising directions and tools to pursue, within the context of preserving adequate stability, continuity and preservation of investment in the existing documentation.

A tentative outline of planned talks (~2 hours talks, ~1 hour discussion) is

- 1) doc overview and status, current tools, doc project view on future tools and future evolution
- 2) talk from IT on doc tools, video doc/training
- 3) Modern doc tools and approaches: experience, ideas, possible directions
- 4) Survey of doc tools and practices in other experiments
- 5) Software education

We would like your input to the workshop, not only through participation but by giving input in advance towards talk 3. That talk will be prepared by Mario Lassnig (mario.lassnig at [cern.ch](mailto:mario.lassnig@cern.ch)) and a TBC colleague, based both on their experience and your input. You can give input either by emailing Mario or by commenting in the googledoc linked from the indico agenda. Please contribute your comments and join us at the workshop.

The indico agenda will take shape at <https://indico.cern.ch/event/681173/>.

Torre & Davide