

Arches Community Webinar: Documentation

November 20, 2020

Presented by: Adam Cox

So I wanted to start with a quick update about-- or just an overview of Arches Documentation, and where it exists, and where the code is, and how you can contribute. So let me share the screen.

So this, many of you are probably familiar, is the main page for the Arches Documentation. We host it on Read the Docs. And I want to especially draw attention to the URL here, which says, "stable." And that's the default if you just go to this website. If you don't to that raw URL, it'll default to the stable link.

So this is the documentation that is for the current release of Arches. The alternative link in the top there, you can view-- you can switch versions down here. And if you switch to latest, that's going to be showing the unreleased features, so to speak, documentation.

So it's important to just keep this URL-- this end part of the URL in mind if you are passing around links to the documentation. Because as you can see down here, we also maintain the documentation for all past versions of Arches. So 4.4.2, this will show up in the URL. And you'll be able to see the documentation as it applies to that version of Arches.

So by default, it's most important to pass around the stable links to the documentation because that's what we're most concerned with keeping current and relevant. At this moment, however, we've been doing a lot of work. So I wanted-- I'm making some updates to the structure a little bit. So I wanted to show what that's going to look like.

So I'm going to go to the latest documentation and-- get a little more room-- and just quickly go over the sections that as we've divided them up, it's a little bit of an update from how Arches Documentation has been in the past. So we have an introductory section. We've changed what used to be-- it used to contain a little bit more. But now, the installing section is more concise about installing and also contains information about packages and projects.

We've pulled some of what was in here into a new configuring section. So this has and will have more, hopefully, basic, one-time or maybe two-time configuration changes that you need to make so initial setup, settings, how to make sure that the collector is ready to go, et cetera.

The administering section is more for database admins who may be designing the database, managing ontologies. We see these as changes that will need to continue to happen throughout the maintenance of Arches installation from the administrator side.

Using Arches is more about actual creating resources, deleting them, how to use the search panels. A lot of this-- and even a lot of this is content that is mirrored in the actual help documentation within the app itself on the right-hand top-right corner. You can get to a lot of this content. We have a couple of sections about the Arches Collector, the mobile app.

And then, finally-- well, not quite finally, I guess-- we have, which we've had for a long time, a developing section. This has a lot of information, which will help developers customize, extend, import, export information to Arches.

So looking at it from a user standpoint, this is kind of developer, implementer, server-side. This is browser-use, really. Or some is browser. Some is not. In the configuring section, administer and using, that's for people using the app through the browser. And then developing, of course, is server back-end.

And then, finally, we've put up a few more-- moved a good bit of content into this running Arches and production section. Some of this needs to be updated. Well, that's on the list to do. But we hope this new segmentation of the documentation makes sense-- feedback always welcome.

And from here, to get to the Arches repo or the repository on GitHub that creates this documentation, a real easy way is to click through this. This will click through to the repository and even-- what that clicks to is specifically the file that is being rendered in that page.

So once we're in the repository, let me just give a quick overview of what this looks like. In the code-- well, let's just go to the code section. We've had all of the information in the wiki up to this point, and I'm just now moving it into the readme, so it's more prevalent, more obvious in the front. But here, in the main page of the Arches Documentation repo, we have instructions on how to do a number of different things related to the documentation.

So the first I want to talk about is how to report an issue. This is a standard way for-- or we use the standard GitHub ticketing system for this. And as I mentioned here, if you see a problem with a documentation, really, what we would prefer is the first thing you do is look through the existing issues.

And I have a link here, but you can get to it right here. So this is our issue-tracking system for tickets or problems with the repo. And you can see, there's plenty of them here. There's plenty of things that need to be done with the documentation. They range from small typos to whole new sections that we see useful in adding.

So if you do see a problem, check this out first. And if it's something-- and if you find the issue that you wanted to report, and there's no documentation about how to delete resources-- oh, OK, someone's already made an issue about this. It's always good to add a comment here that just says, I found this to be a problem too. Or even simpler, you can add a reaction. It's nice to have traces of people and their reactions and use of this content.

The next thing that I wanted to go over is the branch release and versioning system. So this, we've set things up in a way that plays nice with Read the Docs, which is the-- as you can see, the hosting platform, of course. I'm going to go back to the-- so we've set things up to play nice with that, as well as be a reasonable way to manage documentation or just additions in the future.

So what this means is we have a separate branch in this repository for each version of the documentation. And so if, for example, a change needed to happen, some important change needed to happen that people who are running 4.4.1 needed to be aware of, we could put a change directly into that version of the documentation very easily. And the connections here are important to understand, which is that the stable branch that I mentioned before, the default Arches Documentation content-- so this runs-- this is the content that is in the highest numbered branch in the repository. And then, the latest branch is the master branch. Or the latest documentation is built from the master branch.

So what are these branches? You can see up here at the top of the code base for the repo, if you hit this branch dropdown, these are all the different branches. And this is how you can switch between them to see the content. So what this means, one of the consequences of having different versions and different branches is that we don't use quite a standard in GitHub or Git Workflow. In that, we don't merge branches together very often, if ever.

What we do is we branch off from the master when a new release of Arches is made. And we create a new branch with the version number. So, soon, we'll make a 5.1 branch. And then we maintain that one alongside the master branch. However, as you can imagine, new documentation should be added to the master branch for unreleased features before it's put into the stable documentation because that's not in the stable code base for Arches.

So this gives us a couple of different options for workflow. So when it comes to somebody actually making a contribution to Arches-- and for the sake of brevity, I'm not going to get too detailed into demonstrating all of this. But I will give a brief demonstration and explanation of what it would look like to make one pretty simple edit to the documentation.

So the first thing to do to contribute to this repo and many, any, basically any other GitHub repo, is to [INAUDIBLE] into your own GitHub account. And that allows you to make edits to your version of the repository. And then to make a pull request, which basically requests that your code be contributed back to this main repository.

So I can do this right now, and you'll see how quickly that happens. I've got it in one account, but I'll do it into this account. You'll see how quick it is to create a fork of a repository.

And so this is an exact mirror of the main repository that's under the official Arches project GitHub organization. So in this, we see all the same content, but it's important to recognize that this is not what Read the Docs builds from. This is separate. So Read the Docs only reads content from this repository.

So if we want a major change, we can do it in this one. And then, like I said, put a pull request back to the other-- to the main repository to ask for those changes to be incorporated. So I am going to-- there is one issue that I saw earlier that is real easy. And again, we can search for existing issues in here very simply.

So there's a really simple detail here. CSV import uses resource IDs with a space. But we actually want it without a space. So this is the typo that we need to fix. And so what I'm going to

do is fix this in the master branch, which means that whenever the master branch-- which means that the fix will be immediately applied to the latest branch. And soon, when we split off for 5.1, it will be incorporated into that new 5.1 branch.

So what we'll see here, in the CSV import description-- so this is in the import-export section. It's nice that Rob made a link so that we can see a little bit more of the context. And, in fact, the easiest way to do this, because we see here-- and it's this-- this is the issue, this little header right here. So we can view it here because he happened to put that link in.

And now, if we go into edit on GitHub, we'll see what specific file is rendered there and needs to be edited. So again, looking at the URL, we notice that, of course, it's the 5.0.0 branch, which is the stable branch. We actually want to edit the master branch. But this does tell us that it's the import-export file, importexport.txt.

So I'm going to go over to my repository that I just-- my fork of it. I'm going to the docs folder, which is where all the raw content is stored. And I'll find the import-export. For very simple fixes like this, we're able to just make this change directly in GitHub. This isn't really a recommended method for anything more than maybe typing out a new paragraph or making a small typo change because it's not going to render or preview any of this updated syntax. But it does allow a super fast way to make very small changes.

So I can just say, Edit this file. I have to make sure I'm on the right branch, in fact. So I'm going to cancel first, and I want to do this on the master branch. So I'm on the master branch. We're good. I'll say, Edit. I'll scroll down to find the table, which had that column header, this one here. So this is the only change I need to make, very simple.

And then I'll say, fix typo. And we really prefer if you can add the number of the ticket that you're addressing into the commit message. And if you do this, it creates a hyperlink between the issue. So we'll say number 128. So fix typo number 128. And what's important is that I can do this directly to the master branch or to a new branch. So I want to make a new branch. I'll say, master fix 128. I'll say, propose change.

So this opens a pull request, and I don't want to be making it to my own repo, necessarily. It's quicker if I-- oh, I see. I'll make it to my own repo. No, I'm going to-- I'm going to back up a second here. I'm going to actually commit this change directly to the master branch on my repo. And I realize there's another change to make. Oh, there's a lot of changes to make.

And I'll say, fix typo 128. And then commit this directly to the master branch, the typo in the typo message. So I've committed this to my master branch, and what I would do now, finally, is go to create a pull request. And I'll say, create pull request. And I'll do this, compared to the Arches docs base repository from mine.

So this is the comparison that says take my new master branch and merge the changes from it into the official Arches documentation master branch. And what it will do is show the poorly spelled commit message and also a link to the actual commit, right there. And I missed the number sign too. Sorry, I'm feeling a little behind on time.

So at any rate, I will create this pull request and say-- this is another opportunity to put a link to the ticket in here. And I'll do it here. We prefer if you can fill this information out. But well, it's very important to fill this first part out, and then these boxes have to do with the branching.

And in this case, this one is not even relevant because it's about building the docs without error, which we made such a tiny change, we're not concerned with. And then, this has to do with something that we won't really cover today, but it's in the readme content of the official Arches Documentation-- what it means to cherry-pick and all of that.

So we will create this pull request. I'll just take this opportunity. So I've created this pull request. And once you do it, it actually moves you straight over to back to the official Arches repo. So this says-- this gives a summary. It says, Mr. Adam Cox, my username wants to merge one commit. And that commit changed that one little typo into Arches project master from Mr. Adam Cox master. So it's basically saying from my repo branch, pull it into the official repo branch. And this is where you have to leave-- where you can leave it for your purposes or for anybody really contributing unless you have contributor's status for this repo.

That's a really quick way to make some small changes. I kind of had a feeling that people who are much more familiar with Git, will understand the documentation that we have written here about how to commit changes to an existing branch or especially to make changes that apply to both of the unreleased and the exist-- and the stable branches. So that's the most important thing is read this up. But I think anyone familiar with Git, that'll make sense.