

DevMtg 2018-08-01

Developers Meeting on Weds, August 1, 2018

Today's Meeting Times



- DSpace Developers Meeting / Backlog Hour: 20:00 UTC in [#duraspace IRC](#) or [#dev-mtg Slack channel](#) (these two channels sync all conversations)
 - Please note that all meetings are [publicly logged](#)

Agenda

Quick Reminders

Friendly reminders of upcoming meetings, discussions etc

- [DSpace 7 Working Group \(2016-2023\)](#): Next meeting is tomorrow, Thurs, August 2 at 14:00 UTC
- [DSpace Entities Working Group \(2017-18\)](#): *This Working Group will be replaced by [DSpace 7 Entities Working Group \(2018-19\)](#)*
- [DSpace Developer Show and Tell Meetings](#): Next Show & Tell will be August 28 at 15:00 UTC. The topic will be DSpace + Docker

Discussion Topics

If you have a topic you'd like to have added to the agenda, please just add it.

1. (Ongoing Topic) [DSpace 7](#) Status Updates for this week.
 - a. [DSpace 7 Working Group \(2016-2023\)](#) is where the work is taking place
 - b. DSpace 7 Dev Status spreadsheet: https://docs.google.com/spreadsheets/d/18brPF7cZy_UKyj97Ta44UJg5Z8OwJGi7PLoPJVz-g3g/edit#gid=0
2. (Ongoing Topic) DSpace 6.x Status Updates for this week
 - a. **Master ports from 6.3.** A number of PRs merged into 6.3 release have not yet been ported to `master` branch. These PRs are closed but still have the "port to master" label.
 - i. https://docs.google.com/spreadsheets/d/1X-Zk56gz-wg6p7JaiuBzzUquqOvwwx_-o_ZDDvGPSQU/edit?usp=sharing
 - ii. As we're working through this list, we should remove the 'port to master' label from the dspace-6.x PR and if possible, edit the JIRA comment to remove the "63_PORT_TO_MASTER" text.
 - b. 6.4 will surely happen at some point, but no definitive plan or schedule at this time. Please continue to help move forward / merge PRs into the dspace-6.x branch, and we can continue to monitor when a 6.4 release makes sense.
3. NOTE: IRC logging bot has been blocked from Freenode 😞
 - a. Stopped logging in IRC on Friday, July 27. Logs are now filled with SASL-related error messages: <http://irclogs.duraspace.org/index.php?date=2018-07-27>
 - b. Tim has not found a way around this as of yet. Indications elsewhere are that [Freenode has begun blocking AWS-based IPs, and requiring SASL authentication](#). Our currently used logbot doesn't support SASL authentication.
 - c. Also reached out to <https://botbot.me/> to see if they could log our #duraspace IRC channel.
4. [DSpace and Docker](#)
 - a. Tutorial: <https://dspace-labs.github.io/DSpace-Docker-Images/>
 - b. PRs for DSpace 4, 5, 6, 7 - see [DSpace and Docker](#) link above
5. Discussion topics / half-baked ideas (*Anything more to touch on with these?*)
 - a. Bulk Operations Support Enhancements (from [Mark H. Wood](#))
 - i. Better support for bulk operations (in database layer), so that business logic doesn't need to know so much about the database layer. Specifically, perhaps a way to pass a callback into the database layer, to be applied iteratively to the results of a query.
 - ii. Then, the database layer can handle batching, transaction boundaries, and other things that it should know about, and the business logic won't have to deal with them.
 - iii. This is the result of thinking about a recent -tech posting from a site with half a million objects that needed checksum processing.
 - iv. (This is almost an extension of the tabled topic below regarding [DSpace Database Access](#), but a bit more specific in trying to simplify/improve upon how bulk operations are handled)
 - b. [Curation System Needs](#) (from [Terrence W Brady](#))
6. How to encourage / credit folks who do Code Reviews? ([Tim Donohue](#))
 - a. We have a lot of open PRs. As we know, the process for reviewing is very ad-hoc, sometimes encounters delays. If we can find ways to encourage/empower folks (even non-Committers if they know Java / Angular well) to do code reviews & be credited publicly...maybe we can speed up this process?
 - b. Other brainstorming welcome!
7. Tickets, Pull Requests or Email threads/discussions requiring more attention? (*Please feel free to add any you wish to discuss under this topic*)

Tabled Topics

These topics are ones we've touched on in the past and likely need to revisit (with other interested parties). If a topic below is of interest to you, say something and we'll promote it to an agenda topic!

1. Management of database connections for DSpace going forward (7.0 and beyond). What behavior is ideal? Also see notes at [DSpace Database Access](#)
 - a. In DSpace 5, each "Context" established a new DB connection. Context then committed or aborted the connection after it was done (based on results of that request). Context could also be shared between methods if a single transaction needed to perform actions across multiple methods.
 - b. In DSpace 6, Hibernate manages the DB connection pool. Each **thread** grabs a Connection from the pool. This means two Context objects could use the same Connection (if they are in the same thread). In other words, code can no longer assume each `new Context()` is treated as a new database transaction.
 - i. Should we be making use of `SessionFactory.openSession()` for READ-ONLY Contexts (or any change of Context state) to ensure we are creating a new Connection (and not simply modifying the state of an existing one)? Currently we always use `SessionFactory.getCurrentSession()` in `HibernateDBConnection`, which doesn't guarantee a new connection: https://github.com/DSpace/DSpace/blob/dspace-6_x/dspace-api/src/main/java/org/dspace/core/HibernateDBConnection.java

Ticket Summaries

1. Help us test / code review! These are tickets needing code review/testing and flagged for a future release (ordered by release & priority)

key	summary	type	created	updated	assignee	reporter	priority	status	fixversions
Unable to locate Jira server for this macro. It may be due to Application Link configuration.									

2. Newly created tickets this week:

key	summary	type	created	assignee	reporter	priority	status
Unable to locate Jira server for this macro. It may be due to Application Link configuration.							

3. Old, unresolved tickets with activity this week:

key	summary	type	created	updated	assignee	reporter	priority	status
Unable to locate Jira server for this macro. It may be due to Application Link configuration.								

4. Tickets resolved this week:

key	summary	type	created	assignee	reporter	priority	status	resolution
Unable to locate Jira server for this macro. It may be due to Application Link configuration.								

5. Tickets requiring review. This is the JIRA Backlog of "Received" tickets:

key	summary	type	created	updated	assignee	reporter	priority
-----	---------	------	---------	---------	----------	----------	----------

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Meeting Notes

Meeting Transcript (IRC Bot is not working)

- IRC Transcript is available at ~~<http://irclogs.duraspace.org/index.php?date=2018-08-01>~~

Log from #dev-mtg Slack (All times are CDT)

Tim Donohue [2:43 PM]

@here: Reminder that our next DSpace DevMtg is at the top of the hour (~15mins). Agenda is at: <https://wiki.duraspace.org/display/DSPACE/DevMtg+2018-08-01>

Tim Donohue [3:00 PM]

@here: It's DevMtg time (agenda is above). Let's start off with the usual quick roll-call

Terry Brady [3:00 PM]

hello

Mark Wood [3:00 PM]

Hi

Tim Donohue [3:01 PM]

well, we got a few of us. I'm imagining others may pop in here in a bit.

Regarding usual updates... I don't have much to say this week about DSpace 7. Just a reminder that the next DSpace 7 meeting is tomorrow at 14UTC in Zoom.

On the DSpace 6.x front...the usual reminder that *we really need some help porting all the 6.3 PRs to `master`*. I've been trying to find time myself to get back to that, and if anyone is willing to chip in here, we could use it

Currently, there are bugs on `master` that were fixed/released in 6.3

I think that's it on the 6.x front though... we just need to ensure we get those PRs created/merged prior to (even thinking about) any sort of 6.4 release.

On to the #3 topic on this agenda. Our IRC logs are broken as of last week. The IRC log bot we were using is unable to connect to freenode (unclear if this is permanent or temporary, but it's been going on for nearly a week now)

So, at least for the time being, we are unlogged. We'll have to copy our notes (manually) into the agenda doc. As noted in the agenda...I've seen reports of similar issues elsewhere (googled around). I also reached out to <https://botbot.me/> to see if they will log us (haven't heard back yet)

So far, no solutions. But, if anyone has ideas, feel free to pass them my way. I'm hoping maybe botbot.me can simply help us out, so that we don't have to even manage these logs ourselves anymore

I think that's all I have to say there, unless there's questions/comments

Ok, on to #4 on the agenda! DSpace + Docker

<https://wiki.duraspace.org/display/DSPACE/DSpace+and+Docker>

@terrywbrady: is there anything here you'd like to specifically say/note?

Terry Brady [3:11 PM]

I have Dockerfiles migrated to DSpace/DSpace and DSpace/DSpace-angular. It would be good to get these reviewed and merged.

The tutorial should help even a new Docker user to get up and running.

Once those PR's are merged, we have an opportunity to automate image builds anytime that a branch is updated.

That also requires DSpace/DSpace to grant some permissions to DockerHub.

Tim Donohue [3:12 PM]

Are we wanting to test this on a single branch (merge one specific PR & try out), or go "all in" (try to merge all PRs)?

Terry Brady [3:14 PM]

The person who volunteers to test, can pick the branch where they want to begin. I recommend 6x since it is "production". The variations between branches are slight. I noted that in the PR's

@Pablo Prieto did some testing on this last week and helped me work through issues with the angular image.

Pablo Prieto [3:14 PM]

joined #dev-mtg by invitation from Terry Brady.

Tim Donohue [3:15 PM]

A basic question (looking at the 6.x PR: <https://github.com/DSPACE/DSPACE/pull/2134/files>) ...does all this docker stuff need to be in the root folder? Is there any purpose to creating a `docker` subfolder to group it as related?

Terry Brady [3:15 PM]

I have some other notes in DSpace and Docker that will be more relevant to discuss after the merge.

Pablo Prieto [3:15 PM]

Hi all

Terry Brady [3:17 PM]

The build will run from the root directory, so it seemed like a logical place to put it. We could also test the process from a subfolder.

When a user runs the build on their desktop, they will likely be in the root folder.

Pablo Prieto [3:17 PM]

I think it needs to be in the root for docker build . to work

Else I guess it would have to be docker build ../ <-- dunno if this works. I always build inside root.

Terry Brady [3:18 PM]

Do check out this link <https://wiki.duraspace.org/display/DSPACE/Sample+Output+Accessing+Built+Images> ... it will demonstrate how easy it is to switch DSpace versions.

Tim Donohue [3:19 PM]

Most of the files seem OK in the root. The one that specifically "bugs me" is the `docker.local.cfg`. I guess the name implies what it's for, but I kinda wish it wasn't floating around in the root directory

I don't mind the `Dockerfile` and `.dockerignore` all that much

Terry Brady [3:20 PM]

That file would be easy to move.

We mount it as local.cfg within the build.

Tim Donohue [3:21 PM]

Can we simply generate it dynamically in the `Dockerfile`? (it's so small) (edited)

Mark Wood [3:21 PM]

So it could live in dspace/etc or some place like that.

Terry Brady [3:21 PM]

That works for local.cfg. build.properties is much bigger when building 4x or 5x.

@tdonohue could you add these as review comments on the PR (to track the discussion). I can then make some revisions to match.

Tim Donohue [3:23 PM]

yuck, `build.properties.docker`: <https://github.com/DSPACE/DSPACE/pull/2136/files>

Sorry...these are likely nitpicky

But, I'm worried about confusion about extra configs laying around (for those who don't use/need docker)

Terry Brady [3:24 PM]

I hate that build.properties file. It seems as if I do not include all the variable definitions the build becomes unhappy.

Mark Wood [3:25 PM]

Yes, it does.

Terry Brady [3:25 PM]

Mark's suggestion of keeping these in /etc sounds good

Mark Wood [3:25 PM]

Maybe gather them all, with the usual names, in etc/docker/

Terry Brady [3:26 PM]

If Dockerfile and .dockerignore can stay in root, I think that suggestion sounds good

Tim Donohue [3:26 PM]

yea, I guess they could go under `dspace/etc/docker`. They'd likely end up under [dspace]/etc/docker in production sites...but, we could add a filter to Ant if we want to keep them out of production install directories

Terry Brady [3:27 PM]
Good idea. Does stuff in etc get installed?

Tim Donohue [3:28 PM]
yes, `[src]/dspace/etc/` gets installed to `[dspace]/etc/`

Terry Brady [3:28 PM]
I was glad that I investigated the automated build stuff... it forced me to improve & simplify the Dockerfile.

Mark Wood [3:28 PM]
Aha, yes, we keep some utility SQL scripts there.

Tim Donohue [3:28 PM]
The other option is to stick them under something like `[src]/src/main/resources/docker/` (i.e. we have a `src/main` directory under the root: https://github.com/DSpace/DSpace/tree/dspace-4_x/src/main)

Terry Brady [3:29 PM]
where does the test driver local.cfg live?

Mark Wood [3:29 PM]
Maybe we need a designated directory for "junk we only use while building."

Tim Donohue [3:30 PM]
@mwood: that's the `[src]/src/main` directory, I think

Mark Wood [3:30 PM]
./dspace-api/src/test/data/dspaceFolder/config/local.cfg

Terry Brady [3:30 PM]
thanks.
I hope you guys get a chance to step through the tutorial. Once you start playing with this stuff you will see that it is quite elegant for testing.

Mark Wood [3:31 PM]
@tdonohue I think you're right. Assembly descriptors and license wrangling configuration.
So: src/main/docker ?

Pablo Prieto [3:32 PM]
I agree @terrywbrady, I use docker DSpace containers to do testing

Tim Donohue [3:32 PM]
@mwood: exactly. That's why I think I'm leaning towards this stuff going under `[src]/src/main` somewhere... either just at `[src]/src/main/docker` or `[src]/src/main/resources/docker/`

Pablo Prieto [3:32 PM]
I am able to build dspace-angular with the Dockerfile inside a docker subfolder using `docker build -f docker /Dockerfile` .

Tim Donohue [3:32 PM]
src/main/docker is likely good enough...it follows the pattern set here: https://github.com/DSpace/DSpace/tree/dspace-6_x/src/main

Pablo Prieto [3:33 PM]
I guess it can be any folder

Tim Donohue [3:33 PM]
@terrywbrady: I'll add comments to the PRs to have those files moved. Then, I'm OK with it. I just don't like DSpace Docker configs in the root folder

Terry Brady [3:35 PM]
Thanks @tdonohue. @Pablo Prieto, I hope you will also add reviews to the PR's especially the ones you have tested.
We are always seeking a second approval on our PR's.

Mark Wood [3:35 PM]
I recall a question about letting some other site rummage in Github?

Pablo Prieto [3:35 PM]
@terrybrady Sure

Terry Brady [3:35 PM]
Thanks @Pablo Prieto
Yes @mwood. That is how we can automated builds

Mark Wood [3:36 PM]
So that's waiting on someone to say "okay, do that"?

Terry Brady [3:37 PM]
I have a secondary github account terrybradyC9 that I used for testing builds. That account has limited access to repos.
@mwood yes. I would like some approval on that. It cannot be activated until the Dockerfiles are merged. I posted to Code4Lib hoping to find another open source project that has granted this access.
I saw some forks of Fedora that had automated builds, but I did not see it on the main Fedora images.

Tim Donohue [3:39 PM]
I'd just need to see what access DockerHub needs to our GitHub. I'm presuming though that it's not much different than say Travis CI or Snyk (both of which we had to give limited access to GitHub in order to support continuous integration/checks). So, It's likely OK

Terry Brady [3:39 PM]
@tdonohue would one of your Fedora colleagues have any insight to offer? (edited)
It asks for a lot of permissions and then promises to do limited stuff.
I think it adds some hooks/triggers in your GitHub account that call out to the Docker builds.

Mark Wood [3:40 PM]
Heh, a lot of projects need to re-granulate their access design after seeing what people actually want to do....

Tim Donohue [3:41 PM]
Fedora has a separate Docker repo: <https://github.com/fcrepo4-labs/fcrepo4-docker> (in their labs area). They don't seem to have Docker built automatically from the main repo at <https://github.com/fcrepo4/fcrepo4>
That's what I'm aware of on the Fedora side. I can ask around if there are more specific questions...but they don't seem to have this "automated" from what I see

Mark Wood [3:42 PM]
It might be useful to know why.

Terry Brady [3:42 PM]
It would be good to know if they didn't automate because of concerns about permissions.

Tim Donohue [3:43 PM]
I can ask and see if there was a specific decision, or if it's simply that it's experimental (hence in their "labs" repo)

Terry Brady [3:45 PM]
Docker runs DockerHub and DockerCloud and I am not sure who the customer is for DockerCloud. I have seemed to find many open source projects on DockerHub, so I focused on its automated build capabilities. At a follow up meeting, I can share more details. (Some of those details are already on the DSpace and Docker page)
Thanks for the time on this (and for your upcoming PR reviews):grinning:

Tim Donohue [3:46 PM]
Sure thing. One other question that just came to mind here... if we are auto-generating Docker images on all these branches...does that mean we generate a new image *per commit* (i.e. Docker images don't really track release versions, but individual commits)?

Terry Brady [3:47 PM]
It imagine that we would place this on the 4 major branches, and it would re-run on each merge.

Tim Donohue [3:47 PM]
So, we should have a big red warning that Docker images are not guaranteed stable. You're essentially running code that isn't yet released (i.e. it's merged code, but not necessarily gone through the whole release/testing process)

Terry Brady [3:48 PM]
DockerHub only allows one simultaneous build (until you pay a subscription), so things would likely queue up.

Pablo Prieto [3:48 PM]
I believe Docker is not meant for production use.

Tim Donohue [3:48 PM]

Docker is for production use, actually :slightly_smiling_face:

Mark Wood [3:48 PM]

Which is more useful? latest bleeding-edge code, or releases?

Terry Brady [3:48 PM]

Our recommendation will be to use it for testing.

Tim Donohue [3:48 PM]

Or, it can be...AWS supports Docker for production, etc

Pablo Prieto [3:49 PM]

I've read some warnings on Docker for production, specially on red-hat
They might be outdated, though.

Mark Wood [3:49 PM]

I've heard rumors that our "corporate" IT folk prefer Dockerized applications.

Tim Donohue [3:49 PM]

The reason I asked this...is that Fedora's strategy seems to be *release based*. They have a new Docker *per*
released version here: <https://github.com/fcrepo4-labs/fcrepo4-docker>

Terry Brady [3:49 PM]

Once all of this is done, I will be pushing for us to deploy AWS instances of each of the supported branches.

Tim Donohue [3:50 PM]

We are talking about a different strategy from Fedora then...an automated, *commit-based* one

Terry Brady [3:50 PM]

We should also build an image for each release. (edited)
We might also decide that we rebuild branches on a schedule rather than on each commit.
In that case, our automation might need to reside in another spot.

Tim Donohue [3:52 PM]

right, I guess I'm wondering what exactly we are going for...are we building a developer tool that needs to be
commit-based? Are we building something that is for testing/demos that needs to be more release-based? Are we
doing both?

Terry Brady [3:53 PM]

Fortunately, I think we now have the Dockerfile targeted for the right location (in the code itself).
I see it as a rolling pre-release candidate that is always available.

Mark Wood [3:53 PM]

Developers can do their own builds. Maybe published images should be more stable?

Pablo Prieto [3:55 PM]

You're right. @mwood That's my use case. But there are many other options.

Terry Brady [3:55 PM]

My hope is that we could engage a new audience to participate in feature testing...

Mark Wood [3:55 PM]

Ah, that does call for frequent updates.

Tim Donohue [3:56 PM]

So, it looks like at DockerHub we can tag things (and you've done that already): <https://hub.docker.com/r/dspace/dspace/tags/>
And there's a README to describe which tags are "stable": <https://hub.docker.com/r/dspace/dspace/>

Terry Brady [3:56 PM]

Since we generally merge at the DEV meeting, we could manually build relevant branch images weekly after the
DEV meeting.

Tim Donohue [3:58 PM]

@terrywbrady: I'm confused by that comment...are "branch images" separate from the automated process you've
laid out of per-commit images? Does that change the strategy in these PRs?
I'm not sure I understand the proposal here completely yet...But, it seems like we need a strategy for (1)
which images to generate (manual or automated). (2) how to tag them, (3) how to warn folks which may be
unstable and which are more "production ready"

Terry Brady [4:00 PM]

I am making up names for things as I go. By "branch image" I am referring to 4x, 5x, 6x, master. These are volatile since the branches change. A "release image" would refer to stable releases/tags in GitHub. PR's are another matter since the changes reside in an external repo.

Mark Wood [4:00 PM]

Do we want to provide "production" images at all?

Terry Brady [4:01 PM]

We would want to provide the production images so someone could test the state of a system as of any published release.

(we do not release that often)

Tim Donohue [4:02 PM]

So, it sounds like we have two types of images... release-based (stable), and commit/branch-based (potentially unstable).

Do the recent PRs (like <https://github.com/DSPACE/DSPACE/pull/2134>) only support the latter? Or do they support both somehow?

Terry Brady [4:03 PM]

The PR's are indifferent.

The only difference is what triggers the build (and how easy it is to checkout source from GitHub).

When you start a Docker build, it assumes your source code is in the current directory (thus the placement of `.dockerignore`)

Tim Donohue [4:05 PM]

realizes we are over time here a bit

Ok, so this all said, I'd like to see us "finalize" a full strategy on a single branch (maybe 6.x?)

Terry Brady [4:05 PM]

@mwood sorry that we did not get to your agenda item

That seems like a good choice to me.

Mark Wood [4:06 PM]

That's not a problem. I haven't had time to dig into it this week.

Tim Donohue [4:06 PM]

I feel that merging a 1/2 finished strategy into every branch is potentially problematic...I'd rather we figure it out on one branch & then port it everywhere else

Terry Brady [4:06 PM]

Should I tag the others as work in progress?

Tim Donohue [4:07 PM]

Yes, likely that's the best choice, especially for 4.x and 5.x. It seems like 6.x may be the first step.

Terry Brady [4:07 PM]

Will do

Tim Donohue [4:07 PM]

thanks @terrywbrady!

Terry Brady [4:08 PM]

Thanks for the time on this. I think this is going to be really useful going forward.

Tim Donohue [4:08 PM]

We'll close out the meeting for today. I'll copy these notes (manually) into the agenda (since our IRCbot is not working)

Talk to you all next week, if not sooner!

DSPACEslackBot (IRC) APP [4:09 PM]

atst has quit the IRC channel

Mark Wood [4:09 PM]

Thanks, all.

Pablo Prieto [4:09 PM]

Thanks!

DSPACEslackBot (IRC) APP [4:09 PM]

mhwood has quit the IRC channel

Terry Brady [4:09 PM]
Have a good week!