2009-04-14 - Committer Meeting (Ithaca)

Present: Aaron, Andrew, Bill, Chris, Eddie

Migrating to Maven

We resolved to move our builds to use m2 instead of ant. Initial scope is for the repository distribution. Services can come afterward.

- Advantages to community:
    - Easier to contribute (understand code, etc)
    - Play nice with other m2-savvy projects (Duraspace, Akubra, DSpace, Topaz, etc)
    - More up-to-date "built" docs ("site", javadocs, etc)
- Advantages to us:
    - Easier deployment of new releases
    - Third party libs
    - Maven is mature/popular; we can benefit from plugins (eclipse, tomcat, etc)
    - Dependency tracking
    - Testing and reporting integration
    - Licensing info
    - Reduce scm downloads (developer happiness)
    - Encourages modularization to code
- Strategy
    - 1a - Source conformant to m2 standards (files and packages) + tweak build.xml
    - 1b - Get poms for all dependencies
    - 2 - Ant-maven integration (monolithic) with main goal of installer.jar
    - 3 - m2-only (initial split-out)
        - Java Client API
        - Server Webapp
        - Installer
        - (Old GUI)
        - New GUI
    - 4 - Split out server modules as projects
- Timing
    - Want to ensure no working branches are out during 1a, so
        - We should check outstanding branches in general
        - Soon after 3.2 release. But give 3.2 a couple weeks to settle first.
        - Needs to happen in a branch within in a short period of time (budget 2 weeks)
        - Kai's branch/local copy of his current work (will it go in prior to 3.2 release?)
    - 2 - would like to have in place for 3.3...this work can happen without disrupting outstanding branches (as much)
    - 3 - Nice-to-have by 3.3. but will require similar "locking of trunk". We'll revisit timing of this after 2 is done.
- Outstanding Questions:
    - During 1a, how to we preserve svn history.
    - How to do higher-level tests (ConfigA/B)
    - How to do database config in the normal webapp way with Fedora
    - Step 2: Unsure of how to create dependent pieces (wars, etc) in m2

Maven Repository

- We'll host our own for thirdParty and having a copy of our released artifacts.
- We'll point to central, codehaus, java.net, etc directly in our poms.
- We'll have our CI builds use the local mirror of above, by modifying settings.xml
- The local mirror isn't intended for public use. If a problem, we can put behind authN.
- We want to deploy to central. To support that, we should:
    - Deploy org.fedoracommons stuff to a place they can point to (our repo, rsync, etc)
    - Change package names
    - Use m2.fedoracommons.org
- Side note: need ssl cert for m2.fedoracommons.org
- Committers need accounts to deploy poms (thirdparty, etc)

Development Community

- Chris: Need to continue making participating inviting
- Eddie: Community needs sense that they "own" Fedora; not just us
- Brad McLean joined us for a few minutes and talked about some issues DSpace devs are currently facing:
    - Lots of patches, but hard to find people/time to apply them
    - Rules for committers/voting/etc. Looking at Apache as a model for this.