Maven Release Artifact Use & POM Optimizations

For the past few releases of DSpace, Maven has been the build system, and its implementation has grown more mature. I would like to suggest what I consider some important adjustments toward further maturity for the project.

The main change is to start using the "Release" artifacts when cutting an official DSpace release.

This year I have been working on a solution for MIT's use of the build system that would ensure a consistent and reproducible build without any unknown shifts in dependencies unless explicitly included by our team (Hint: DSpace doesn't currently have this) and reduce redundancies in DSpace's multi-level POM setup and allow more global control from the Parent POM.

For this, we needed to be certain that we were building with the same components every time. Building on Monday should be exactly the same as building on Friday, save for any changes we make locally in between. The current setup that DSpace source is released with does not provide this certainty and introduces uncertainty as to what code is being used. This is due to the fact that the POMs are shipped with SNAPSHOT repositories enabled, and the Release repository actually _disabled_. What this means is that if SNAPSHOT artifacts are redeployed off of the maintenance branch, the default build system could pick up modified code without the developer ever realizing something had changed. This is dangerous and doesn't allow for the consistency required by a production environment. What matters here is not whether it is the practice of the community to re-release snapshot builds into the repository, but the fact that there is uncertainty. This is why the distinction between these two classes of artifacts exists. The assumption that snapshots are stable is misleading and the mechanism leaves open the possibilty of shifting artifact code.

Currently, the release artifacts Are built and the repository populated with them. They are simply not referenced from the build…

http://maven.dspace.org/release/org/dspace/

It is also important to note that this is not a mirror of the list in the snapshot side of the repository.

http://maven.dspace.org/snapshot/org/dspace/

What we did was reverse this configuration. SNAPSHOT-repository: disabled. Release-repository: enabled. It is my opinion that the DSpace releases should be configured as such, starting with the DSpace 1.7 release. This way, institutions modifying and maintaining their repositories can be sure to have absolute certainty that they are in control of the changes introduced into the software when rebuilding to incorporate their own changes.

I understand that the shortcut of using snapshots instead of switching to releases bought us some flexibility early on, but I think it's time to stop short-cutting and gain some certainty in releases of the code. For instance, always building off of the snapshots might allow for builds to always pick up the latest fixes in a maintenance branch, but one picks up the bugs, too, and with our more frequent calendar-based releases, fixes are released officially on a more regular basis.

For the attemp to reduce redundant entries in the various levels of POMs, the main change was to eliminiate the redundant repository definitions at every level of the POM hierarchy, allowing Maven repositories that apply to the entire build to be modified in a single file, and applied via the parental hierarchy of POMs. I would recommend this clean-up task for the next release as well, as it will make swapping between snapshot repositories & release ones much easier, when developers want to work with the "edge" DSpace or the latest cut of a maintenance branch.

I know this setup has served its purpose, but IMHO, I think adjusting our practices to align with Maven's well thought-out conventions would serve to stabilize the platform.