

# SIT Meeting Highlights & Atlas Offline SW Release Building Strategy

Frederick Luehring

Indiana University

December 9, 2005

Friday Plenary

Atlas Software Workshop at LAL

# SIT Session Highlights

- Documentation
  - There was lots of news (see previous talk).
- Code Management
  - Lots of work on migrating from CVS to Subversion (SVN)
    - If we decide to switch to SVN, this change affect everyone using the Atlas software. Decision to use SVN must be considered carefully.
  - Latest version of CMT is now in production.
  - Likely time for switching is March but some more testing remains before there is a formal request to the SPMB.
  - Tag collector crashes have been investigated. Additional protections and monitoring have been put in so that when the tag collector crashes it will be restarted automatically.
    - Restart mechanism was tested and worked fine during the SIT meeting!

# SIT Highlights (continued)

- Release & Kit Building

- Overall status for Release & Kit Building is very good.
- Nightlies usually OK but there are occasional problems.
  - Three types of problems - no single main failure source.
- Release schedule is 11.0.3 on Dec. 13: no delay is possible.
- Project builds working but still under final testing.
  - Modifications to tag collector may delay dropping the current monolithic builds until March but there is a possible work-around.
  - There will be a time when both project and monolithic builds are done.
- Grigori Rybkine is now responsible for kit building tools.
  - Many changes are already made.
  - Support for multi-platforms and documentation is coming soon.
- We discussed building the production releases and kits using the nightly release building tools.
- Working with HLT/TDAQ/Commissioning on providing offline software at Point1 and coordination of shared packages.

# SIT Highlights (continued)

- Testing & Validation

- All three testing systems (ATN, KV, and RTT) now using the common control tags and file.
- The rate of successful tests is increasing but there still are tests that fail every night in ATN and RTT.
  - Developer please fix or remove your tests!
- Testing group is working to provide timely feedback about the problems with the release/kit builds.
- RTT is just now gaining the possibility of using additional dedicated nodes in Lancaster.
  - This should reduce the turn around time for longer tests.
  - RTT now has the possibility of defining jobs in different categories such as “long jobs” and “short jobs”.
  - Active exploring use of the grid to provide more resources for running tests.

# Release Building Strategy

- I am worried about the way that we have been closing, building, and validating the 11.0.X releases.
  - The process seems not as organized as it should be and if it continues could be very problematic when Atlas is running.
- What the rest of this talk is and is not:
  - This talk is not about the build tools.
    - I believe that we have the tools that we need to manage building releases with the geographically dispersed Atlas developer community.
    - Some work remains on Project Builds, Testing, and Subversion
  - This talk is about the way we manage building a release.
  - This talk is not trying to assign fault.
  - This talk is an attempt to start discussion that will lead to improvements.
    - This discuss must involve all interested communities.

# Observations on Recent Builds

- When asked how close they are to being ready, developers give very optimistic time estimates.
- Developers lose track of the release schedule request tags after the build start.
- Developers who know the release schedule put in many tags in at the very last minute.
  - No matter how delayed the release is, many tags come in at the last minute.
- The full tag approval process rarely rejects any tags.
- There is no full-time, dedicated release coordinator.

# Observations (continued)

- As soon as the release is closed, there is immediate pressure to open the next release.
- The testing packages often are not ready once the release is built.
- The release patching process is not well publicized.
- Bugs are not systematically reported in one place.
- Critical bugs are found after the release is out.
- Some developers use Y.0.X as their main development path instead of Y.X.0.

# Consequences

- The release schedule slips - sometimes badly.
- There is little discipline about deadlines.
- People lose track of the schedule.
- Planning is difficult.
- The release builds converge very slowly.
- There is little or no time to validate the release.
- The kit must be changed after it has been announced.
- The lack of structure makes mistakes more likely.

# Fixes - Build Management

- That during release building we post a daily status messages to atlas-sw@cern.ch.
- That we all agree to stop putting new functionality into production releases (Y.0.0, Y.0.1, ...)
  - It is reasonable that a high-level (SPMB?) exceptions can be made to allow specific new functionality in Y.0.X releases.
- That we again have a dedicated release coordinator.
  - As well as coordinators for each project.
- That starting two days before a Y.0.X build written requests are required ahead of time for new tags.
  - Ideally there should only be bug fix tags during this period.
  - We should classify the submitted tags into risk categories e.g.: job option changes, changes affecting a few packages and changes affecting lots of packages.

# Fixes - Build Validation

- That starting from two days before the release someone should validate the release each day.
- That the validation of the build and kit be a two stage process with quick tests followed by longer tests.
- That we require that tags submitted be tested by a coordinator before they are accepted.
- That we not open the next release in a branch until the current kit is validated as working.
  - Of course there will still be cases where hidden problems are discovered after the kit is built.
  - Not clear how this plays out with project builds.

# Fixes - Release/Kit Patching

- That we should minimize the amount of patching packages in the release in place.
  - There is a trade-off between the delay in creating a new version and the confusion caused by patching in place.
    - Patching should only take place before the release is announced.
    - Ideally there should be no patches in place after the release is announced and patch should cause a new release version.
- That the kit have its patch revision number changed every time that kit is changed in anyway once the kit has been announced to the community.
  - This is already agreed to.
- That all changes after the first full build of a release be announced by an email to atlas-sw@cern.ch.

# Conclusions

- We need to strike the delicate balance between having too little control of the build process and having too little flexibility to react to the inevitable problems that develop during the building process.
  - I do realize that there will always be this tension.
- Some of the possible fixes suggested may not be practical or may be unacceptable to the developers.
- However I do ask that we open a dialog on how make the releases converge to code usable for production.
  - We need to think about how to organize the new project-based approach to building the releases.
  - It seems unlikely that the current loosely structured approach will work when we have the fixed deadlines imposed by the LHC schedule for data taking.