

# Hall D DAQ Software Code Management

Elliott Wolin and David Lawrence

13-Dec-2011

GlueX Doc #1892 covering 12GEV line 1532030ca

Design of the Hall D Online code management system involves the following tasks:

1. Choose a code management system.
2. Choose a location for the repository.
3. Determine access restrictions on the repository.
4. Decide on a directory structure within the repository.
5. Choose a build system.
6. Develop a release management/installation/deployment strategy.

Note that the relation to the offline code management system needs to be taken into account for a few of the tasks above.

## **Discussion**

Requirements for the online code management system are a bit different from those for the offline system, but in general it is preferable to use the same or similar packages when possible. Some differences in the online include:

1. Well-defined task, will not expand much as experiment matures.
2. Restricted set of computer architectures and operating systems.
3. Code only runs on the Counting House Computing Cluster (HDCC).
4. Small base of dedicated code developers.
5. Code must run 24x7 and be highly reliable (downtime very costly).

These differences may lead to different choices in the online code management system.

## **Hall D Strategy**

SVN has proven to be a good replacement for its predecessor CVS by relaxing many limitations and adding critical features. The JLab backbone network and CUE file servers have proven to be

extremely reliable so there is no good reason to maintain a separate repository on the file servers in the HDCC. Thus we will simply use a new “online” subdirectory of the existing offline repository located on the JLab central computing system.

A core group of online developers will be given read/write access to the online section of the repository. Development will be performed in personal accounts and commits will be authenticated. In this way we can track who introduced changes or new features. Code will need to be checked back into the repository prior to deployment (see below).

The repository directory structure under the main online directory is not critical. Initial plans for the structure are shown below. We expect this will be modified during the writing phase:

- online
  - SCONS
  - scripts (miscellaneous)
  - package1
    - include
    - libsrc
    - execsrc
    - scripts
    - test
    - examples
    - doc
  - package2
  - etc.

Python-based SCONS is a popular modern build management system that has worked well for the JLab DAQ group, CLAS12 and many others around the world. We will use SCONS (note...the Hall D offline uses gmake). The SCONS scripts will be written such that common tasks will be done in generic scripts that will be invoked by SCONS scripts in the individual package directories, thus minimizing the work needed when creating a new package.

Code installation/deployment will be performed in a special *hdsys* account (group: *hdsys*) created for this purpose, and this account will have exclusive write access to the deployment directories. This will not be a normal login account, instead installers will use ssh to log into this account, with login access controlled by system managers via entries in the *hdsys* ssh “authorized\_keys” file. Access to this account will be restricted to a small group.

In this way files in the deployment directories will always have the correct ownership and protections, and confusion due to incorrect umask setting are avoided (this is common when multiple accounts write to the same deployment directory). Deployment for use by operators will be done in a carefully controlled manner after extensive testing.

Operators will use a special *hdops* account (group: *hdops*) while on shift that will only have read/execute access to the deployment directories. Thus by design code development and

installation/deployment cannot be done from the operator account. Note that there will be far more operators than authorized code developers, and more developers than installers.

### **Manpower Estimates**

The 12GeV schedule allocates 4 man-weeks for planning and 4 man-weeks for implementing the code management system. This seems adequate assuming writing of the many SCONS scripts is accounted for in the individual sections on code writing instead of here. Note that scripts related to overall release management are included here.