

Hall D Online Scripting

Elliott Wolin and David Lawrence

14-Dec-2011

Numerous scripts will be required to implement Hall D Online systems, some running in the background or in CRON jobs, others running interactively with a graphical component. They will be used for a wide variety of tasks including:

- allowing operator to choose run options
- making automatic entries into the elog
- reading out hardware parameters for archive purposes
- sending page or email under fault conditions
- checking processes running on a node against a list of required processes
- performing tasks related to run control transitions

Since many scripts will require an interactive graphical display, choice of scripting language is strongly coupled with the availability of compatible graphical packages.

Web programming is somewhat of a specialty, and scripting languages appropriate for CRON jobs and interactive scripting may not perform as well as languages designed specifically for web programming.

Scripting Languages

Python has emerged as the leading general purpose scripting language in modern HENP and in the world, even surpassing the previous leader, Perl. Ruby has many vocal adherents, but is far less popular than Python or Perl. Python has a very large user base and module library, and is well integrated with many popular graphical toolkits (e.g. GTK using Glade). Modules implementing database-independent access to relational databases are also available.

PHP remains quite popular for server-side web programming, although many web programming modules have been developed for Python. PHP also has a good interface to relational databases. Javascript is the leading non-Microsoft client-side programming package. Note that both PHP and Javascript are implemented as special instructions embedded in html pages.

Hall D Strategy

We will use Python as our preferred scripting language, but will allow Perl for legacy purposes. If the script is very short, a few lines at most, Unix shell languages are acceptable, but Python should be used for all non-trivial scripts.

We plan to use GTK + Glade for graphics, although we remain flexible as graphical programming interfaces evolve rapidly and packages can gain or lose popularity quite quickly.

We plan to use PHP for server-side web programming and Javascript for the client side. We note that Python is developing rapidly on the server side, though.

Repositories and Directories

We will use a separate online directory residing in the standard Hall D main SVN repository¹ to store all online files, and will check files out of SVN to the file server in the Hall-D Computing Cluster (HDCC). The Hall D main SVN repository resides on the CUE² file system, not on the HDCC file server, but recent experience shows that the CUE file servers are quite reliable and there is no need for a separate repository in the Counting House. If problems do arise we can always move the online system files to a dedicated SVN repository on the HDCC file server .

No special requirements exist for the online directory structure, and we expect this will develop naturally during the writing phase. Initial plans are to treat scripts like executable programs and require an “install” step to copy them to release directories. Script source code will reside in subdirectories containing related code, with one additional directory to hold miscellaneous scripts. Note that the SVN system is flexible enough to allow revisions to the directory structure as needed.

Access

Access to online files will be restricted according to which accounts can modify and install files, and which need only have read-access to the files.

¹ <https://halldsvn.jlab.org/repos>

² JLab Central Computing Environment

Online scripts, like all critical online files, will have restricted access to modification of those files deployed in the live system. Two special accounts will be created on the HDCC that will enforce this restriction (more details can be found in the Hall D Software Code Management planning document). Briefly:

hdops: The *hdops* account will be used by shift takers (operators) to run the experiment. The *hdops* account will have read and execute access to the directories containing the deployed online scripts, but will not be able to modify them. In addition, the login file for the *hdops* account will be write restricted such that it cannot be modified by the *hdops* account. This is to prevent additional PATH elements from being added that would point to directories *hdops* could modify files in, bypassing the intended access restriction.

hdsys: The *hdsys* account will be accessible only to select individuals for the purpose of deploying scripts and other online system files. A developer who wishes to deploy a modified script will use the *hdsys* account to do an “svn update” in the appropriate directory followed by an install step.

Current plans are for the install step to simply create links in a standard deployment directory back to files in the *hdsys* svn directories. In this way the *hdops* account PATH can be short and simple, and not have to include many different directories. Note that svn can be used to check that the deployed scripts are in sync with the repository.

Code developers will be allowed to check online code out of the main online repository and check it back in from their own personal accounts. The developer group may include individuals that do not have access to the *hdsys* account. It will be left to those who *do* have access to *hdsys* to verify changes before deploying them.

In this way we will be able to track who makes changes to the code, if and when it gets installed, and ensure that shift operators, who just need to read and execute access to the installation areas, cannot anonymously modify anything.

Manpower Estimates

The 12GeV schedule lists 6 man-weeks for planning software scripts. This document completes most of the planning stage, the exception being the details of the directory structure, which, as described above, will develop naturally as the scripts are written and deployed. The schedule also has 10 man-weeks for writing scripts. We believe the planning stage should be about half of what is allocated, with the excess likely needed in the writing stage.

In addition the schedule includes 11 man-weeks to check-out DAQ software and 11 man-weeks for check-out of the archiving system. Online scripts are covered under both categories, along with many other items, but this seems adequate.