Configuration Management for Observing Software

Configuration Management Plan

Upgrading Software on the Operational Computers

For mature software that is not under active development and has only occasional releases due to minor bug fixes, the developers at FNAL compile the code at FNAL and install the software into a UPS database on a FNAL computer. The observers can then install the software to the local UPS database via a simple install script. Software packages that follow this upgrade strategy include astrom, astroda, dervish, and murmur. For the TCC, which runs under OpenVMS on dec Alphas, the developer takes care of the installation for the observing team, but coordinates his upgrades with our Bright Time/ Dark Run cycles. Other software packages not installable with UPS/UPD such as IOP/SOP and the MCP/TPM are exported from the FNAL CVS database by observers and installed into the local UPS database. No matter how the software is installed, the following describes the handoff procedure:

- Starting at each testing opportunity (see the Shakeup/Shakedown policy statement), the developer checks all code changes into the CVS repository, possibly merges the last CVS branch with the mainline and tags a new mainline version. Such tag names should be of the form v*X*_*Y*_0 where *X* and *Y* are two arbitrary numbers that get changed sequentially through new mainline versions. The last number (*0* in this case) is the branch release number for the product. This number gets incremented for every branch tag.

- A CVS branch should also be created at this point with a tag v*X*_*Y*.

- The Lead Observers is then notified of the changes checked in and is provided with the tag name in accordance with the Shakeup/Shakedown and Software Documentation Policies.

- In time for the shakeup/down tests (or during the dark run for a critical/high bug fix), the Observers will checkout the source code with the tag name specified by the developer and compile and install it. (For reference, we use the installation procedures as seen on our www page, http://sdsshost.apo.nmsu.edu/sdssProcedures/swInstall.html.) In some cases, the developer will install the software, but this should be arranged with the Lead Observer beforehand. Once installed, the new software version will be declared to the local UPS database as a test version.

- The observers test the software module. The release notes and shake request help the observers determine what to test, how to test it, and how to judge success.

- If bugs are found in the software module, the programmer will be notified of the problem though the problem report database (and the night log). Depending on the testing load during shakeup/down, the developer may have the opportunity to correct an obvious bug immediately during the shake period. The decision to fix and test code in this manner will be negotiated between the observers performing the shake testing and the developer responsible for the code. Under no circumstances should the debugging process be allowed to affect the ability of the observers to complete the pre-established shake plan.

- During testing, bug fixes to the software module are made on the branch while new development continues on the mainline.

- Once the bugs are shaken out of the test version, the latest branch version of the software will be declared current to the UPS database and undeclared test.

- The process repeats at the next shake opportunity.

Configuration Management Tools

Source Code Control System

All software developed at Fermilab and most of the code developed at other participating Institutions is stored and tracked in a source code control system. The system in use for Sloan is CVS (Concurrent Version System). Each software package is stored as a separate module in a central source code repository located on a computer at Fermilab. The CVS system allows us to create a stable release of a software package by creating a branch in the repository for that software module. Branches are typically tagged with a version number and only bug fixes for this version of the software module are made on this branch. After a few test and bug fix cycles on this branch a stable version of the software module should develop. Meanwhile enhancements and ongoing development of the software module can continue in the mainline of the repository. The branch and the mainline are periodically merged by the developer so that the bug fixes make it into the enhanced versions of the software module.

Product Database

CVS does a good job of tracking changes to flat ASCII files such as source code, but is awkward to use for tracking compiled binary sources or for switching between different versions of software packages. For this task we use a Fermilab developed database and tracking tool called UPS (Unix Products Services). In the UPS system each software package/module is called a product. All products in the UPS are cataloged in a simple ASCII database and are stored in a special Unix file partition (/p). Each product has a sub-directory in the /p partition, where several complete versions of each product can be stored. This database makes it straightforward to switch between different versions of a product and to track the software dependencies for products.

Problem Reporting Database

All Problem Reports (PRs) for both software and hardware are tracked using a Web-based Problem Report System called the SDSS Problem-Reporting Database (GNATS). PRs in the database can be classified as critical, serious or non-critical. Critical-high software bugs are defined as a problem that prevents telescope operations, adversely affects our ability to take good quality data efficiently, or makes reducing the data impossible. All critical-high bugs are fixed immediately and the fixes are tested right away. Serious bugs are problems or bugs in important telescope commands or operational tools, but for which there is a known work-around. The exact way the GNATS database is used for observing software is discussed in the Policy for Observing Systems Software Development.