

Software Sharing at Fermilab - Experiences from Run II, KITS and Fermitools

R. Pordes, L. Loebel-Carpenter, E. Schermerhorn

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois

Abstract

Fermilab provides many different software packages for specific Experiments or Projects. We promote a strategy to distribute and reuse software once the investment in its development has been made. We describe experiences with three approaches and our ideas for continuing this effort into the future. In the CDF /D0/ CD Joint Offline Project, software is jointly developed or obtained and then supported by the Computing Division and the Collaborations. We describe one such project as an example - the development and support of the experiments data base management systems. Key elements in the promotion of reuse and sharing of software are providing and maintaining central services which simplify and standardize how software is distributed and managed. We describe our central code repository, mechanisms to support distribution from a central restricted FTP site, our email support system, and the service provided by a searchable catalog of software and documentation. Through the Fermitools Program, we have been providing software to the more general community. We work towards a semblance of open source within the operating DOE guidelines.

Keywords: software, tools

1 Introduction

Fermilab users and employees gain from having consistent, site wide mechanisms and standards for the distribution and support of software packages. We also benefit from developing and publicizing software that can be applied to more than one experiment or application in support of the Laboratory's program. The Computing Division provides tools and central services towards the development, support and distribution of software. Through specific examples, this paper describes some of these mechanisms, some of the benefits and problems associated with such an approach, and mentions some goals for the future. We focus on three examples: The CDF/D0/CD Joint Support Databases Project; Software Package maintenance and support; Publication of Fermilab developed software to the community.

2 CDF/D0/CD Joint Support Databases Project

CDF and D0 Run II requirements for calibration and run information, together with the number of data files to be tracked - for raw, reconstructed and analyzed events - led to the decision by both experiments to rely on Oracle for their administrative and file catalog databases. Each experiment has separate online and offline Oracle repositories. Each experiment has many non-experts participating in the design and implementation of the databases and the applications to use them. The largest - D0 offline in support of the SAM data access project - is anticipated to reach a half Terabyte in size by the end of data taking. Each experiment requires databases with similar functions but different in detail. Through one of fifteen Joint Projects [1], a central team is working with the

collaborations on the design, implementation and support of these databases. The Joint Project and central team facilitates the use of common tools and methodologies; provides a forum for the interchange of problems and solutions; and promotes the sharing of software and designs. The common or shared aspects of the Support Databases project include:

- Use of Oracle Designer 2000 for the design and maintenance of the tables and indexes. Cross-fertilization of expertise is achieved by the core team providing the support for Oracle designer, defining standards for the naming, sizing and documentation requirements for the database design. We plan to have the experiments review each others database designs, which should result in more robust, complete designs for each.
- Consolidation on a single supported platform for most of the needed databases. This reduces the overall support load and increases the pool of expertise.
- A coordinated liaison and lobby group for interaction with and licensing with the company.
- Coordinated database monitoring and backup strategies and implementations. A small number of database administrators act as support for both experiments. Backup information is maintained in a central repository. Common monitoring and alarm tools are used. Centralized recording of errors is achieved. Mechanisms for alerts when errors occur allows deployment of the central data services to dispatch the problems to the support staff.
- It is proposed to have a common off-site database strategy and infrastructure. We are currently evaluating Oracle on Linux. Collaborating institutions will have local sub-sets of the main catalogs and administrative information - calibration, run data etc - which must be updated from the main repository at Fermilab and verified as being consistent with the central database of information.

3 Software Product Services and Support

The Fermilab Computing Division provides packaging, distribution and/or management of general application software for our employees and users. We provide this uniformly for acquired, purchased or locally developed packages. Any Fermilab developer or package provider can request to place software under the main KITS area which is accessible via FTP or the standard Fermilab UPD (unix product distribute) mechanisms [2]. Procedures are in place to provide for licensed or otherwise restricted access software storage and distribution. Support is given for multiple instances of a software package to be stored, allowing for binary distribution, multiple Versions, Flavors (Flavor defines the operating system) and Qualifiers (Qualifiers specify other configuration information such as what compiler options were used to build the binaries). For example, D0 requires software libraries compiled with `-thread`, whereas CDF does not; multiple instances of CORBA and ROOT packages are provided in order to satisfy their different requests.

The large number of utilities and packages in frequent and widespread use at Fermilab, make it difficult for the available small central support groups to maintain sufficient expertise, current knowledge and to ensure that the most recent versions are available on site. We have therefore instituted defined support using "community support mail aliases" [3]. For each software package a mail alias `xxx-users@fnal.gov` is provided to which the most active and knowledgeable users and supporters, as well as the central package providers are subscribed. All mail to the list is archived and made available on the web. The main CD document search engine [4] searches these archives and the central HELPDESK support services know to browse them for answers to "frequently asked questions". Standards and conventions for the naming, packaging and documentation of software are developed and publicized. The resulting consistency encourages automation of the presentation of information, increases the productivity of users who need to access and interact with many diverse packages, and increases the pay-back from the development of infrastructure tools for the

maintenance and support of the software products.

We have extended the ups/upd support structure to include a "product census" [5]. UPC uses the ups/upd mechanisms for listing the versions of software available on any node which supports ups. A Perl script delivers web readable reports. Alternatively, tabular text reports are generated, which can be used by system administrators to easily check the versions of software installed on a local node against those available on the central distribution site. This tool is becoming more valuable as we seek to maintain consistency across many diverse clusters and nodes, and coordinate the versions of software across the many computing systems being installed for Run II.

Of course, infrastructure is an ongoing maintenance load in its own right. As operating systems, compilers, and experiments change and adopt new technologies, resources are required to maintain the product support framework. Clearly the goal is to use fewer resources to maintain an infrastructure than we would expend without it. However, the bookkeeping is hard to do, especially in a situation where no infrastructure is available and many people in many different areas spend a small fraction of their time on support activities.

A further benefit of standards and conventions and tools to maintain software packages is that we have a good record of the configuration used to build a final application executable. This knowledge is invaluable to the debugging of complex applications and systems.

As elsewhere, NT [7] and Linux PCs [8] for desktop and application servers are becoming ubiquitous. For scientific applications we support product installation, distribution and setup under cygwin - the traditional ups, upd and upc products are available. For Linux, many products and device drivers are available as RPMs. We are looking at ways to integrate our product support mechanisms across RPM and ups/upd. The KITS repository, documentation and support mechanisms are clearly independent of the platform and packaging of the software package; but the configuration management and distribution infrastructure must be extended and revamped. The performance and cost effectiveness of such desktop systems should allow us to restrict support of some general software utilities - e.g. word processing - to workstations, rather than having to support the software on all the platforms on site.

4 Publishing Fermilab Software

Fermilab developers - engineers, physicists and computer professionals - who develop or modify software can take advantage of the Fermitools program [6] to publish their software to the wider community when it becomes used by our own Fermilab users. The software is made available on an anonymous FTP site and is available for download when a client has accepted the posted Terms and Conditions.

Product Providers conform to a minimum set of standards - README file, etc. - and are given space on the Fermitools web pages. Support email aliases and associated Hypermail archives are established for the packages and help with support and bureaucracy is given. We are able to give our developers the opportunity to publish and advertise their software under existing operating guidelines without having to address the needs on a case by case basis.

The Fermitools program also addresses a less publicised aspect of software support - "end of life" and de-support. The Fermitools operating guidelines provide mechanisms to allow us to transition this support from our local groups to those still using the software.

5 Acknowledgments

We would like to thank all the people who are contributing to this work from the experiments and the Computing Division, especially those in the Fermi Users Environment group.

References

- 1 <http://runiicompating/>
- 2 <http://www.fnal.gov/docs/products/ups/ReferenceManual/html/>
- 3 <http://www-oss.fnal.gov/uas/Community-Support.html>
- 4 <http://cddocs.fnal.gov/cfdocs/productsDB/docs.html>
- 5 <http://www.fnal.gov/upc/>
- 6 <http://www.fnal.gov/fermitools/>
- 7 Software Release Tools on NT, G. Watts et al. CHEP 98, Chicago
- 8 Software Management at Fermilab, R. Harris et al. Chep 98. Porting and supporting the Fermilab CDF Run II Software under Linux