

Software Management for the NO ν A Experiment

G.S Davies¹, J.P Davies², C Group³, B Rebel⁴, K Sachdev⁵,
J Zirnstein⁵

¹ Indiana University, Department of Physics, Swain Hall West, Room 117, 727 E. Third St.,
Bloomington IN 47405-7105, USA

² University of Sussex, Department of Physics and Astronomy, Sussex House, Falmer,
Brighton, BN1 9RH, UK

³ University of Virginia, Department of Physics, 382 McCormick Road, PO Box 400714,
Charlottesville VA 22904-4714, USA

⁴ Fermilab, PO Box 500, Batavia IL 60510-5011, USA

⁵ University of Minnesota, School of Physics & Astronomy, Tate Lab Room 148, 116 Church
Street S.E., Minneapolis MN 55455, USA

E-mail: gsdavies@iu.edu, j.p.davies@sussex.ac.edu, rcg6p@virginia.edu,
brebel@fnal.gov, ksachdev@physics.umn.edu, zirnstein@physics.umn.edu

Abstract. The NO ν A software (NO ν ASoft) is written in C++, and built on the Fermilab Computing Division's art framework that uses ROOT analysis software. NO ν ASoft makes use of more than 50 external software packages, is developed by more than 50 developers and is used by more than 100 physicists from over 30 universities and laboratories in 3 continents. The software builds are handled by Fermilab's custom version of Software Release Tools (SRT), a UNIX based software management system for large, collaborative projects that is used by several experiments at Fermilab. The system provides software version control with SVN configured in a client-server mode and is based on the code originally developed by the BaBar collaboration. In this paper, we present efforts towards distributing the NO ν A software via the CernVM File System distributed file system. We will also describe our recent work to use a CMake build system and Jenkins, the open source continuous integration system, for NO ν ASoft.

1. NO ν A and NO ν ASoft

NO ν A [1] is a long-baseline neutrino experiment that consists of two detectors: Far Detector (FD) close to International Falls, Minnesota and Near Detector (ND) at Fermi National Accelerator Laboratory (FNAL), Batavia, Illinois as shown in Figure 1. The FD is a 14 kton detector and the ND is 300 ton as depicted in Figure 2. The experiment uses the Neutrinos at the Main Injector (NuMI) [2] neutrino beam from FNAL that is predominantly ν_μ in composition. NO ν A is located 14 mrad off-axis from the NuMI beam to maximise oscillation probability. The ND is used to measure the beam spectrum and composition before oscillation and will enable the study of $\nu_\mu \rightarrow \nu_e$ oscillation (ν_e appearance) and also $\nu_\mu \rightarrow \nu_\mu$ oscillation (ν_μ disappearance). The detectors are composed of PVC modules that consist of tubular cells filled with liquid scintillator. Alternate modules are orthogonal in orientation to enable 3-D reconstruction. A loop of optical fibre runs through every cell, each read out by an Avalanche Photo-Diode (APD).

The international nature of the experiment poses a challenge for providing computing access to all collaborators. The NO ν A Offline Analysis Software (NO ν ASoft) is written in C++ and built on FNAL Computing Division's art framework [3] that uses CERN's ROOT [4] analysis



software. NO ν ASoft makes use of more than 50 external software packages, is developed by more than 50 developers and is used by more than 200 scientists from 38 institutions and 7 countries across 3 continents. You are referred to Figure 3 to observe the code development activity of collaborators.



Figure 1. Figure indicates the positions of the NO ν A detectors within the United States.

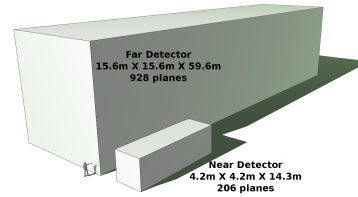


Figure 2. Figure shows the relative sizes of the NO ν A detectors. NDOS was a prototype built to test detector systems.

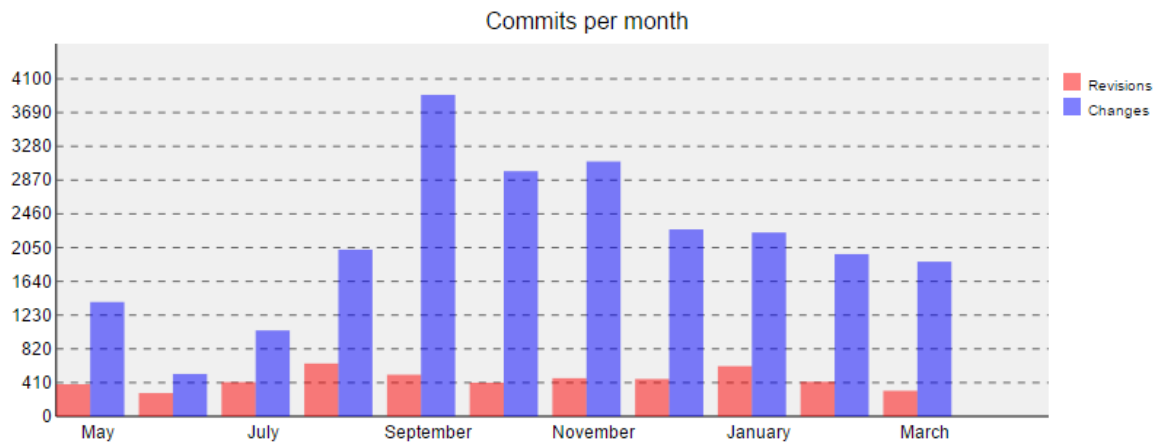


Figure 3. Figure shows the frequency of svn commits to NO ν A Soft over the course of a year. NO ν A Soft developers average approximately 2000 svn commit changes per month and around 400 revision changes.

1.1. Software Release Tools (SRT)

NO ν A Soft builds are handled by FNAL's custom version of Software Release Tools (SRT) [5], a unix based software management system for large, collaborative projects that is used by several experiments at Fermilab. The system provides software version control with SVN configured in a client-server mode and is based on the code originally developed by the BaBar collaboration.

1.2. UPS and UPD

The software for experiments relies upon external tools. These packages and tools are referred to as external products, or simply products. For Fermilab-based experiments, access to external products is provided by a Fermilab-developed product-management package called Unix Product Support [6]. UPS supports multiple versions of a product and builds per version. A product is built multiple times for use with different versions of an operating system, C++ compiler or optimisations. UPS provides a way to select a particular build. The full identifier of a UPS product includes:

- product name
- version
- flavour
- full set of qualifiers

UPD (UNIX Product Distribution) is a companion product to UPS, and provides the functionality for uploading/downloading products between local systems and product distribution servers. NO ν ASoft uses UPS for some internal products as well because of the ease in which they can be distributed via UPD, for example: 'upd install <product> <version>'. This enables us to easily distribute any software to any (supported) system, to run our code on and connect our collaborators.

2. CernVM File System (CVMFS)

CernVM File System (CVMFS) [7] is a network file system based on HTTP and optimized to deliver experiment software in a fast, scalable, and reliable way. Files and file metadata are aggressively cached and downloaded on demand. CVMFS client provides a virtual file system that loads data only on access. All releases of NO ν ASoft are hosted as a CVMFS repository on a web server. Data transfer via HTTP removes any requirement on installing or building NO ν ASoft locally at non-Fermilab locations. Key components of CVMFS are:

- Filesystem with a single source of data (repository Stratum 0) maintained by a dedicated release manager machine
- CVMFS server tool kit publishes the current state of the repository on the release manager machine
- Repositories are replicated to a handful of web servers (Stratum 1) and data cached by standard web proxies such as Squid
 - Squid is a caching proxy for the Web supporting HTTP, HTTPS, FTP etc
 - Bandwidth requirement is reduced and response times improved by caching and reusing frequently-requested webpages

Currently, NO ν ASoft is maintained and published on an Open Science Grid (OSG) CVMFS server - nova.opensciencegrid.org - that hosts both stratum 0/1 servers. We have access via OSG to multiple clients and specific NO ν A dedicated offsite clients that include Harvard University, Southern Methodist University, Institute of Physics of the Academy of Sciences (Prague, Czech Republic) and Ohio Supercomputing.

3. Jenkins

Jenkins [8] is an award-winning application that monitors executions of repeated jobs, such as building a software project or jobs run by cron. Jenkins focuses on the following two jobs:

- Building/testing software projects continuously
- Monitoring executions of externally-run jobs

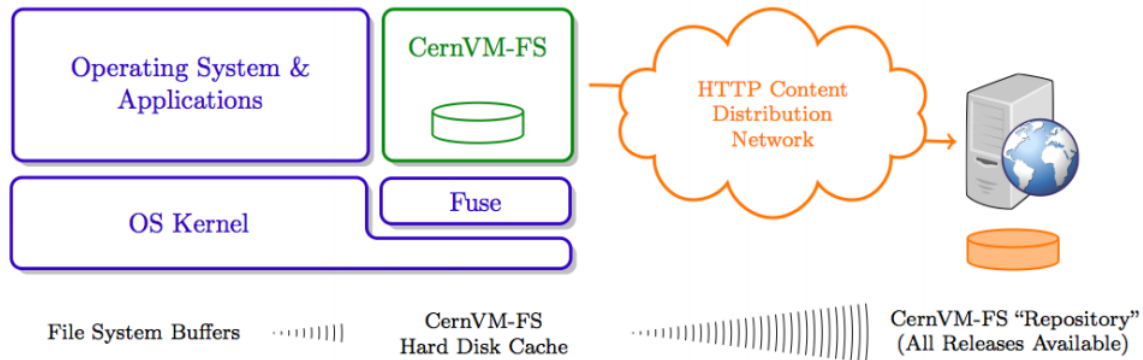


Figure 4. Pictorial representation of CVMFS and its relation to operating system and HTTP transfer to repository.

We build NO ν ASoftnightly for distribution to developers and use Jenkins as an early warning system for repository commits that will break the development code base and also for aiding with building different versions on different operating systems.

S	W	Name	Last Success	Last Failure	Last Duration
		Externals_install_check	22 min - #15	48 min - #12	20 sec
		nova_build	4 mo 16 days - #7	N/A	3 hr 0 min
		nova_build_output	4 mo 16 days - #8	N/A	6 min 12 sec
		nova_nightly_build	8 hr 11 min - #167	12 days - #155	1 hr 55 min
		nova_SRT_nightly_build	10 hr - #58	11 days - #47	2 hr 44 min
		nova_SRT_nightly_build_output	7 hr 36 min - #14	N/A	2 min 26 sec
		nova_SRT_nightly_release_build	21 days - #5	19 days - #5	2 hr 10 min
		nova_SRT_nightly_release_build_output	21 days - #11	21 days - #10	1 min 42 sec
		nova_SRT_nightly_release_build	8 hr 58 min - #72	11 days - #61	25 min
		nova_SRT_nightly_release_build_output	8 hr 32 min - #50	N/A	2 min 2 sec
		nova_SRT_nightly_release_build	1 day 5 hr - #13	1 mo 3 days - #1	24 min
		nova_SRT_nightly_release_build_output	7 days 22 hr - #8	8 days 3 hr - #1	16 min
		nova_SRT_nightly_release_build	1 day 4 hr - #21	1 day 4 hr - #20	1 min 27 sec
		nova_SRT_nightly_release_build_output	20 days - #21	25 days - #18	15 min
		nova_SRT_nightly_release_build	20 days - #30	N/A	32 sec
		nova_SRT_nightly_release_build_output	21 days - #25	24 days - #21	25 min
		nova_SRT_nightly_release_build	21 days - #31	N/A	39 sec
		nova_SRT_nightly_release_build_output	1 mo 23 days - #5	N/A	4 min 51 sec

Figure 5. A snapshot of NO ν ASoftbuilds using the Jenkins application monitoring. It displays the current status of a particular build/flavour with duration, last success, last failure reported as well as a weather report showing the aggregated status of recent builds.

4. CMake

CMake [9] provides an alternative build system to the SRT system currently used by NOASoft. Test releases, while a convenient resource, are known to produce inconsistent local build related problems that are difficult to debug. CMake does away with test releases and requires each user to have a local copy of the entire repository. The CMake build system has been adapted by Fermilabs Computing Division to interface with the UPS package management system. It is completely implemented for NOASoft and is in use alongside of SRT. Since CMake is the industry standard, it has the additional advantage of ample documentation and help on technical forums on the Web.

5. Summary

The NO ν A software, NO ν ASoft, provides a solid, stable platform for code development for more than 200 scientists from 38 institutions and 7 countries across 3 continents. It makes use of over 50 external packages and sophisticated open source tools that are available such as CMake and Jenkins, all built around Fermilab's art framework.

Acknowledgements

The author acknowledges support for this research was carried out by the Fermilab scientific and technical staff. Fermilab is Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy

References

- [1] Ayres D *et al.* (NO ν A Collaboration) 2007 The NO ν A Technical Design Report Tech. Rep. FERMILAB-DESIGN-2007-01 Fermilab
- [2] Anderson K *et al.* 1998 The NuMI Facility Technical Design Report Tech. Rep. FERMILAB-DESIGN-1998-01 Fermilab
- [3] Green C, Kowalkowski J, Paterno M, Fischler M, Garren L and Lu Q 2012 *J. Phys.: Conf. Series* **396** 022020
- [4] Brun R and Rademakers F 1997 *Nucl. Inst. and Meth. in Phys. Res. A* **389** 81–86
- [5] Amundson J 2000 731–732
- [6] Complete guide and reference manual for ups and upd <https://www.fnal.gov/docs/products/ups/ReferenceManual/html/overview.html> accessed: 2015-05-13
- [7] Dykstra D 2015 URL <https://cds.cern.ch/record/2002386>
- [8] Jenkins: An extensible open source continuous integration server <https://jenkins-ci.org/> accessed: 2015-05-13
- [9] Martin K and Hoffman B 2007 **24** 54–7