ATLAS software packaging

Grigory Rybkin
(For the ATLAS Collaboration)
Laboratoire de l’Accélérateur Linéaire, Université Paris-Sud, CNRS/IN2P3, Orsay, France
E-mail: Grigori.Rybkine@cern.ch

Abstract. Software packaging is indispensable part of build and prerequisite for deployment processes. Full ATLAS software stack consists of TDAQ, HLT, and Offline software. These software groups depend on some 80 external software packages. We present tools, package PackDist, developed and used to package all this software except for TDAQ project. PackDist is based on and driven by CMT, ATLAS software configuration and build tool, and consists of shell and Python scripts. The packaging unit used is CMT project. Each CMT project is packaged as several packages—platform dependent (one per platform available), source code excluding header files, other platform independent files, documentation, and debug information packages (the last two being built optionally). Packaging can be done recursively to package all the dependencies. The whole set of packages for one software release, distribution kit, also includes configuration packages and contains some 120 packages for one platform. Also packaged are physics analysis projects (currently 6) used by particular physics groups on top of the full release. The tools provide an installation test for the full distribution kit. Packaging is done in two formats for use with the Pacman and RPM package managers. The tools are functional on the platforms supported by ATLAS—GNU/Linux and Mac OS X. The packaged software is used for software deployment on all ATLAS computing resources from the detector and trigger computing farms, collaboration laboratories computing centres, grid sites, to physicist laptops, and CERN VMFS and covers the use cases of running all applications as well as of software development.

1. Introduction
ATLAS [1] is a general-purpose particle physics experiment at the Large Hadron Collider (LHC) at CERN, the European Laboratory for Particle Physics near Geneva. More than 2,500 physicists from over 200 universities and laboratories in 6 continents participate in the experiment. A large software suite has been developed to simulate and analyse data from the experiment. It consists of Trigger and Data Acquisition (TDAQ), High Level Trigger (HLT), and Offline software making use of some 80 external software packages, or of around 8 GB of files organised into about 2000 packages, including the external software. Production releases are used for large-scale simulation, and data analysis. Physics analysis releases are needed to particular physics groups in addition to the full production releases. Developers need access to the latest development releases and often to the latest night’s build. All this requires packaging and subsequent deployment of the ATLAS software on all available computing resources such as the trigger computing farm at CERN, universities and laboratories computing centres, Worldwide LHC Computing Grid sites, and individual laptops. The first generation of ATLAS software packaging and deployment tools [2] successfully met the challenge. Then the structure of the Offline and HLT software changed—it was split from a monolithic project into some 10 projects. The other requirement
was to optimise packaging time. The PackDist packaging tools [3] met the requirements. In this paper, we present PackDist in detail, in particular, its latest developments and features.

2. Tools

2.1. Configuration management

ATLAS uses the Configuration Management Tool (CMT) [4] to manage configuration and building of its software and set up the user environment. CMT formalizes software production, especially configuration management, around a package-oriented principle. CMT permits description of the configuration requirements through text files, called requirements and projects files, and several environment variables, and deduces from the description the effective set of configuration parameters needed to operate the packages, i.e., to build, use, or query them.

2.2. Packaging

The ATLAS software packaging tools, PackDist, is a CMT package including a suite of shell and Python scripts. The tools make use of the CMT commands and query mechanisms to visit the CMT projects and packages tree, and to retrieve the configuration/meta-data parameters, including dependencies, generating a set of packages in several packaging systems formats for distribution. The tools can be run via a driver script as well as CMT command

\$ cmt make run

They are highly configurable by means of configuration files with simple colon or equal sign separated parameter and value syntax, command line options, and the PACKOPTS environment variable. In particular, packaging can be done recursively to package all the dependencies. PackDist allows for the generated packages installation, e.g., for a test, by running the CMT command

\$ cmt make installation

3. Packaging strategy

The packaging of ATLAS specific software is done at the granularity of CMT projects for faster build and installation. Each CMT project is packaged as several packages—platform specific (one per platform available, the files identified by location and CMTCONFIG value), source code excluding header files (identified by extensions and location), other platform independent files, documentation (the files identified by extensions and location, built optionally), debug information packages (the files identified by extensions and location, built optionally). The necessary external software that is not natively managed by CMT is used by the ATLAS software via dedicated interface CMT packages which also provide the meta-data used by PackDist to package the external software. The most important meta-data parameter is normally the location of the external software build and is specified via the \texttt{<package>_export_paths} macro. PackDist generates an external software package with these contents. The parameter may also be an external software package and be specified via the \texttt{<package>_requires} macro. Finally, PackDist packages several configuration setup packages. The dependencies among the project platform specific packages are set to be the dependencies of the CMT projects. The project platform specific packages are also made dependent on those external software packages for which the dedicated interface CMT packages are included in the CMT project. In addition, they are also made dependent on configuration setup packages and CMT.

4. Packaging formats and platforms

The meta-data retrieved with CMT can be used to generate packages in various packaging formats. The default is the Pacman package manager [5] format. The Pacman package comprises
a “Pacman file”, a text file that references the packaged software—typically, a “gzip” compressed “tar” archive [6]—and describes how the software should be fetched, installed, configured, updated, in particular, the dependencies of the package, and the referenced packaged software itself. The other supported and heavily used by ATLAS format is the RPM format [7]. The RPM package payload is built from the contents of the same compressed archive while the metadata, such as dependencies, name, version, release, pre- and post-install scripts, etc., are the same by design. This guarantees that installations of packages of either format are identical.

The PackDist tools are functional on the platforms supported by ATLAS—GNU/Linux and Mac OS X.

5. Distribution kit
The whole set of packages for one software release is called the distribution kit. It comprises ATLAS specific software, some 80 external software packages and several configuration packages and totals some 120 packages for one platform. Also packaged are physics analysis projects—currently six—used by particular physics groups on top of the full release.

6. Deployment
The packaged software is used for software deployment on all ATLAS computing resources: the detector and trigger computing farms, collaboration laboratories computing centres, grid sites, physicist laptops, CERN Virtual Machine File System (CVMFS). Everywhere, except for the trigger computing farms, the installation is done with the Pacman package manager resolving the dependencies among the packages. On the trigger computing farms where the HLT and Offline projects have a combined installation with the TDAQ project, this is achieved with the help of the apt-get for RPM tool [8], patched in order to allow for non-privileged user operation.

7. Use cases
The deployed software covers the use cases of running all applications, and of software development.

8. Conclusions
The packaging tools, PackDist, have been developed, maintained, and enhanced to package the whole ATLAS software stack (except for TDAQ project) and all the necessary external software for subsequent deployment on all the computing resources available for the experiment. The tools are an integral and important part of the release procedure. The most recent enhancements include the packaging of separate files with debug symbols into debug information packages. Among possible further developments is the support for more packaging systems formats, e.g., Debian system format.

References
[1] ATLAS experiment http://cern.ch/atlas/
[2] Arnault C, De Salvo A, George S and Rybkine G 2004 The deployment mechanisms for the ATLAS software Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP 2004 (Interlaken, Switzerland)
[3] Obreshkov E et al. 2008 Organization and management of ATLAS offline software releases Nucl. Instrum. Meth. A 584 244–251, Albrand S et al. 2010 Organization, management, and documentation of ATLAS offline software releases J. Phys.: Conf. Series 219 042012
[4] Arnault C 2001 Experiencing CMT in software production of large and complex projects Proc. Int. Conf. on Computing in High Energy and Nuclear Physics CHEP 2001 (Beijing, China) http://www.cmsite.org/
[5] Pacman http://physics.bu.edu/~youssef/pacman/
[6] Tar http://www.gnu.org/software/tar/tar.html
[7] RPM http://www.rpm.org/
[8] apt-get for RPM http://apt-rpm.org/