

# The Deployment Mechanisms for the ATLAS Software

ARNAULT, C. (CNRS), DE SALVO, A. (INFN), GEORGE, S. (RHUL),

**RYBKINE, G. (RHUL), YOUSSEF, S. (BU)**

Department of Physics, Royal Holloway, University of London, TW20 0EX, UK

## Abstract

One of the most important problems in software management of a very large and complex project such as ATLAS is how to deploy the software on sites ranging from computing centers to laptops and including the computing elements of a computing grid organization. A set of tools based on both CMT - the ATLAS basic configuration management tool - and Pacman has been developed. The resulting mechanism now supports the systematic production of distribution kits for every ATLAS release, the automatic installation of kits and the running of test suites to validate the installed kits. This mechanism is meant to be fully compliant with the Grid requirements and has been tested in several grid contexts. Several remaining issues will be discussed.

## 1. Introduction

ATLAS [1], [2] is a general-purpose particle physics experiment that will study topics including the origin of mass, the processes that allowed an excess of matter over antimatter in the universe, evidence for Supersymmetry and other new physics. The experiment is being constructed by some 2000 scientists in over 100 institutes in 6 continents. The experiment will be located at the 27 km circumference Large Hadron Collider at CERN, the European Laboratory for Particle Physics near Geneva.

A large software suite has [3] been developed to simulate and analyse data from the experiment. It consists of around 4 G Bytes of files organized into about 1000 packages, including over 50 external packages which are not developed within the project. Production releases are made approximately every 3-4 months, intermediate developer releases every two weeks and snapshot builds every night.

Production releases will be used for large-scale simulation, data mining and data analysis. Some analysis is also done with intermediate releases. Developers need access to the latest releases and sometimes the latest night's build. "Data Challenge" [4], "Tier-0", "Rome Production" exercises have been carried out to gain experience with the Grid working model, test software tools, and harness available computing resources of ATLAS collaborators around the world for the production of large quantities of simulated data. All this required and requires deployment of the ATLAS software at various computer sites ranging from computing centers in the usual sense down to individual laptops and including the computing

elements of a computing grid organization. This paper describes the approach taken to packaging and installation and how the installation kits are produced.

## 2. Tools in use

### 2.1 CMT

ATLAS uses the Configuration Management Tool (CMT) [5] to manage configuration and building of its software and set up the user environment. CMT comprises several shell-based utilities. It is an attempt to formalize software production, especially configuration management, around a package-oriented principle. The packages represent a set of software components that are to be used for producing a system. CMT permits description of the configuration requirements and deduces from the description the effective set of configuration parameters needed to operate the packages (for building them or using them, including querying).

### 2.2 Pacman

Pacman [6] is a package manager. It can be used to define how the software they wish to distribute should be fetched, installed, configured and updated. The packed software can be any mix of tarballs, RPM or other formats. The description (or meta-data) goes in a separate "Pacman file" that references the packed software itself. A directory of Pacman files, usually accessible via the web, is known as a "Pacman cache". The cache contains instructions on how to fetch and install software. The software itself can be anywhere else on the web.

### 3. Full Chain

#### 3.1 Creating the distribution kit

Since ATLAS already uses CMT, and the software is already organised into a hierarchy of interdependent CMT packages, the packaging unit chosen is a CMT software package. A set of scripts was developed to construct the distribution kit of every package. This comprises a “gzip” compressed “tar” archive [7] with the package contents selected for the kit and a Pacman file for the package. The scripts make heavy use of CMT to select the files for the kit and deduce the package meta-data for the Pacman file that describes, in particular, the dependencies of the package. The kit is created by the central librarian as the part of the release procedure. The full procedures and associated scripts can be found in the ATLAS Deployment package [8].

#### 3.2 Installing the kit

The procedure is straightforward. First, install Pacman, if not already installed, then choose a directory for the installation, choose from a list on a web page which release to install, and install it with a “pacman -get ” command. It is also possible to install a particular package from a release with a similar command. Then, all the packages on which the chosen package depends will be installed too automatically. Currently, the complete installation consists of one compiler package, over 60 external libraries packages, one CMT package and over 1000 ATLAS specific and the software framework packages taking approximately 4 G Bytes of disk space.

#### 3.3 Validating the kit

A test suite was developed to validate the kit [9]. The tests are first run against the release. Then the kit is installed at the primary installation site (CERN) and the tests are run against the installation to check whether the results are the same. This is part of the release procedure. The same test suite is then used to validate kit installations at remote sites.

#### 3.4 Using the kit

To use the installation for running jobs or developing ATLAS software, some environment variables have to be set (e.g. PATH, LD\_LIBRARY\_PATH). This is done with CMT, in a way that will be familiar to those who have used the ATLAS software directly from the primary site installation.

### 4. Conclusions

A set of tools based on both CMT and Pacman and procedures has been developed. The resulting mechanism has become an integral part of the ATLAS software release procedure and now supports the systematic production of distribution kits for various binary conditions for every release, the partial or complete automatic installation of kits on any site and the running of tests to validate the installed kits. This mechanism is meant to be fully compliant with the Grid requirements. In particular, the deployment may be done by a non-root user, the compiler libraries used to build the distributed software are also installed if necessary. It has been tested in the contexts of several grids and, in particular, of LCG.

Most recent enhancements include the possibility of incremental updates of the installation, and multiple installations co-existing and sharing the configuration tools and common software.

Among further possible enhancements are the adapting of the procedures for the sub-projects into which the ATLAS software will be divided and a full source distribution kit, so that the software can build from scratch.

### 5. References

- [1] ATLAS experiment <http://cern.ch/atlas>
- [2] GridPP ATLAS introduction <http://www.gridpp.ac.uk/atlas/ATLASIntroduction.html>
- [3] ATLAS offline software <http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/>
- [4] G. Poulard for the ATLAS Collaboration, ATLAS Data Challenges, Paper 541, CHEP03; R. Sturrock et al. ATLAS Collaboration, A Step Towards A Computing Grid For The LHC Experiments: ATLAS Data Challenge 1, CERN-PH-EP-2004-028, CERN, Geneva, 30 Apr 2004.
- [5] CMT <http://www.cmtsite.org/>
- [6] Pacman <http://physics.bu.edu/~youssef/pacman>
- [7] Tar <http://www.gnu.org/software/tar/tar.html>
- [8] ATLAS software deployment tools and procedures <http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/Deployment/>
- [9] ATLAS kit validation suite <http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/Tools/KitValidation>