

# ATLAS Releases: Build, Packaging and Deployment - short overview

Oana Boeriu  
University of Sheffield

on behalf of the ATLAS SIT/Code Distribution Groups

with input from:

Attila Krasznahorkay

Emil Obreshkov

Graeme Stewart

# Introduction

- The ATLAS software is composed of multiple projects that are each built up out of many packages.
- After using CMT for many years, at the beginning of 2016 the build of the software releases has been switched over to CMake.
- Around 2000 packages provide:
  - shared libraries to be used by other packages
  - module libraries used by the framework to load software components
  - executables performing specialized tasks
  - scripts, config files - to be installed with the package build target
- A package declares what other packages it needs for its own build.
- A project - a collection of packages built according to the dependency declaration.

# ATLAS Project - Highest Complexity Build Overview



- LCG - represents the externals provided by the LCG release
- dqm-/tdaq-common - online software built with their own configuration
- GAUDI - built using its own CMake configuration
- still building this type of release every night

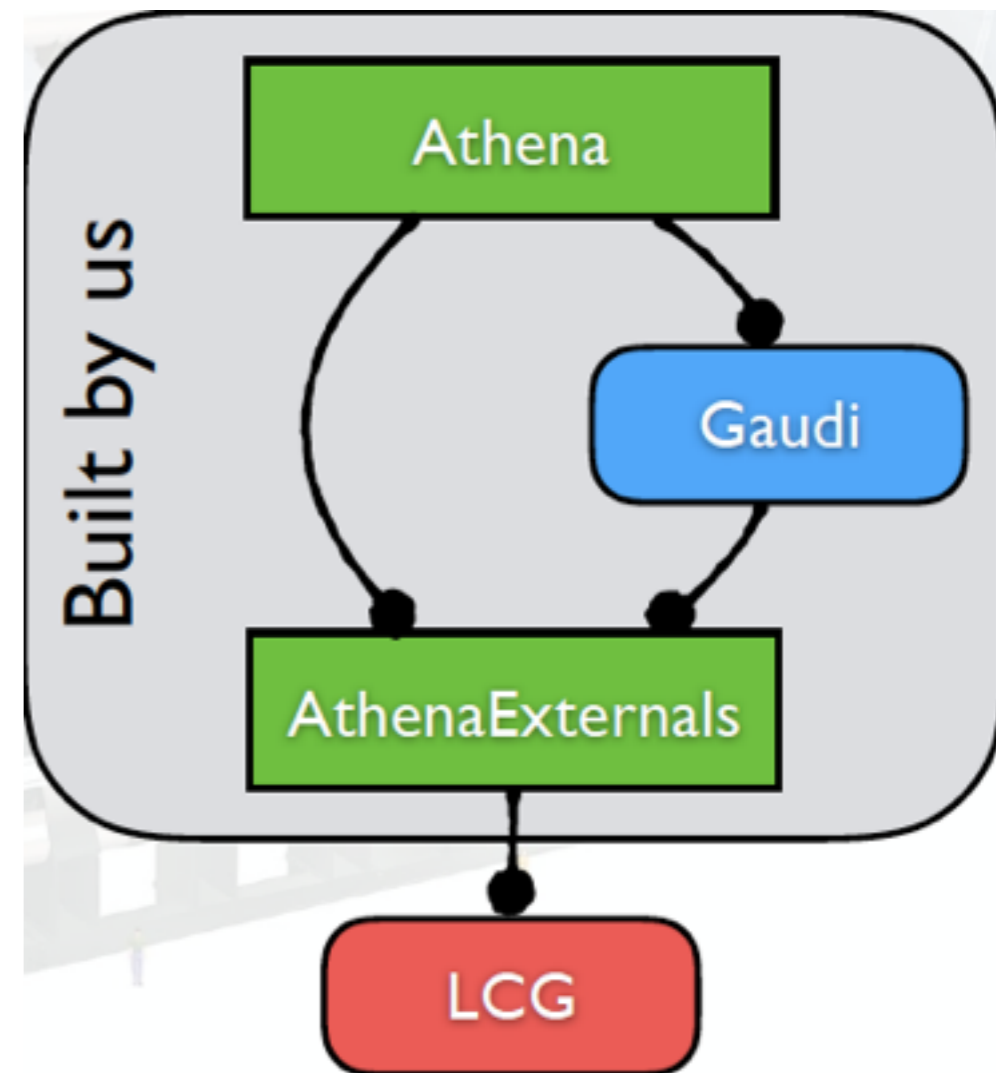
# The Athena Project

- The Athena project structure was simplified, but we still need three projects.
- Dependencies: AthenaExternals and Gaudi
- Versioning - we build each night a particular state of athena git repository together with versions of LCG, AtlasExternals & Gaudi
- The version of LCG is hardcoded in AthenaExternals:

```
31 # Set up which LCG version to use:
32 set( LCG_VERSION_POSTFIX "" )
33 set( LCG_VERSION_NUMBER 87 )
34 find_package( LCG ${LCG_VERSION_NUMBER} REQUIRED EXACT )
```

- Specific version of AthenaExternals and Gaudi needed - taken from the externals.txt file

```
8 AthenaExternalsVersion = 1.0.4
9 GaudiVersion = v28r1.005
```



- **Steps of the build:**
- master CMakeLists.txt file describing a project
- **AtlasSetup** sets up:
  - the environment e.g. TDAQ\_RELEASE\_BASE
  - the gcc compiler
  - sets up the version of CMake
- Find the base project(s) that it depends on
- Relocatable RPMs - environment variables guide the build configuration for CTest and CPack
- **AthenaExternals** - first project to build and install
  - the setup.sh is sourced to set it up for the other projects
- **GAUDI** is built next and installed (“GAUDI\_ATLAS mode”)
  - GAUDI\_ROOT points to the installed location of Gaudi
- **Athena** is built and installed last

# Scripts

- General scripts, used for building Athena as well as other flavors (AnalysisBase, AthDerivation, AthAnalysis etc.) can be used from Build/AtlasBuildScripts
  - checkout scripts for AtlasExternals and Gaudi
  - environment setup scripts
- Project specific scripts are put into the main directory of that project, e.g. Projects/Athena
  - **build\_externals.sh** - a script building all externals
  - **build.sh** - a script building the Athena project
- Private manual builds are possible as well.

# Standard Builds & Packaging

- The build is done basically using:

```
asetup none,gccXX --cmakesetup  
./athena/Projects/Athena/build_externals.sh  
./athena/Projects/Athena/build.sh
```

- The nightly RPMs are release candidates which can be used directly to distribute a release on the grid
- CPack used, having the option of producing opt/dbg rpms
- Once the rpms are locally built, they are copied to eos by an independent script.

# Distribution

- Python scripts used for the cvmfs installation (some 2780 lines of code - of which ca 900 lines belong to monitoring)
- Create date-time directories for every nightly release, e.g.:
  - `/cvmfs/atlas-nightlies.cern.ch/repo/sw/master/`
    - `drwxr-xr-x. 5 cvmfs cvmfs 7 Nov 27 04:59 2017-11-26T2257` → 

3 Nov 27 04:35	AthenaExternals
3 Nov 27 04:36	GAUDI
3 Nov 27 04:36	Athena
    - `/cvmfs/atlas-nightlies.cern.ch/repo/sw/21.0 / 21.1 / 21.2 / 21.3 / ...`
- LCG, external sw, tdaq-/dqm-common used in common by several nightlies:

```
atlas
Geant4
sw
tdaq-common
dqm-common
```

- Possible to **install a nightly anywhere else** (using a standalone script) as **user**.
- **Private builds** can be installed as well on cvmfs to be used by all users.



# Future Improvements

- Having the option to save the binaries of AthenaExternals e.g. and not start a build every time from scratch
- We'd like an easy way of building and reusing existing binaries
  - Nexus - sw repository to store binary artifacts - not sure if it is open source and deserves some more detailed check
  - git LSF ?
- Breaking up the large rpm file into several smaller chunks
- Include the build of LCG/tdaq-/dqm-common into our own build procedure?