First use of LHC Run 3 Conditions Database infrastructure for auxiliary data files in ATLAS

L Aperio Bella¹, D Barberis², W Buttinger¹, A Formica³, E J Gallas⁴, L Rinaldi⁵ and G Rybkin⁶
on behalf of the ATLAS collaboration

¹ CERN
² Dipartimento di Fisica, Università di Genova, and INFN Sezione di Genova, Via Dodecaneso 33, I-16146 Genova, Italy
³ CEA/Saclay IRFU/SEDI, 91191 Gif-sur-Yvette, France
⁴ Department of Physics, Oxford University, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, United Kingdom
⁵ Dipartimento di Fisica, Università di Bologna, and INFN Sezione di Bologna, Via Irnerio 46, I-40126 Bologna, Italy
⁶ Laboratoire de l’Accélérateur Linéaire, Université Paris-Sud, CNRS/IN2P3, Orsay, France

E-mail: Grigori.Rybkine@cern.ch

Abstract. Processing of the large amount of data produced by the ATLAS experiment requires fast and reliable access to what we call Auxiliary Data Files (ADF). These files, produced by Combined Performance, Trigger and Physics groups, contain conditions, calibrations, and other derived data used by the ATLAS software. In ATLAS this data has, thus far for historical reasons, been collected and accessed outside the ATLAS Conditions Database infrastructure and related software. For this reason, along with the fact that ADF are effectively read by the software as binary objects, this class of data appears ideal for testing the proposed Run 3 conditions data infrastructure now in development. This paper describes this implementation as well as the lessons learned in exploring and refining the new infrastructure with the potential for deployment during Run 2.

1. Usage of auxiliary data files in ATLAS

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. The auxiliary data files it uses consist of a variety of calibrations, alignments, efficiencies, weights and other useful constants, which are produced by experts using the physics data and are essential for user analysis in the context of ATLAS distributed computing infrastructure. These files are currently stored in a simple file system structure under an AFS (Andrew File System) dedicated area, that we call Calibration Area, at CERN and then propagated into the CVMFS (CERN Virtual Machine File System [2]) storage area, which is accessible from external sites via Squid Proxy, and into the High Level Trigger (HLT) farm, located near the ATLAS experiment, for on-line processing.
1.1. ADF data flow
The experts provide in general refinement of the constants stored in the ADF files several times per year, and the directory structure allows to link these files to a specific software version that should be used to read the constants. The size can vary considerably depending on the type of constants, in general, from several kilobytes to several tens of megabytes. The constants are used during the analysis via a centralised tool that is capable of associating the correct set of files needed by a specific software version (resolving automatically the file path for every needed entry).

This tool gives as well the possibility to override the default settings and to access older sets of constants if requested by clients.

1.2. The calibration area on AFS
The total volume of the calibration area is 2 GB of data. This represents \(\approx 0.2\%\) of data accessed via the Conditions Database infrastructure. The calibration area files are not updated very frequently, and in general their validity spans a large range in time. For the moment this IOV (Interval Of Validity) structure is taken into account adding the time information in the name of the sub-directories inside a given package when needed, or even at the level of the file names. Every system has defined its own way to handle the internal file dependencies for a given package. The file type used is extremely heterogeneous: ASCII files (XML or TXT) and ROOT files can be used by the experts depending on their needs.

1.3. Present limitations
The calibration area today is certainly functional, however, several aspects of its design and implementation limit its scalability for the future:

- Directory structure determined essentially by the package experts
- Difficult to handle centrally and time-consuming to create a given calibration area release by the ASG (Analysis Software Group) manager
- Every system has adopted some model for tagging and versioning
- Needs to be exported to Point 1 — ATLAS detector and trigger computing farm location

2. Managing the auxiliary data files using the new LHC Run 3 conditions server
A new system would make the file storage more uniform among different packages: provide a central way for tagging and versioning the files, simplify the creation of a calibration area release, and facilitate on-line synchronisation.

The ATLAS and CMS experiments are exploring a common solution to manage conditions data with the time scale of Run 3, year 2020, shown schematically in figure 1.

This solution foresees three main components: a back-end to make persistent the conditions data objects and the related meta-data like interval ranges, tags and global tags (for the moment based on Oracle), an intermediate web server providing access to the storage layer and delivering REST functions to clients, and finally a set of client libraries (Python and C++) in order to use the REST API from the data processing frameworks. The data model at the level of Oracle consists today of a simple set of tables to handle the meta-data, and all conditions data objects are stored as files (BLOBs in the database language). A more detailed description can be found in the CHEP 2015 article [3].

This data model seems well adapted to contain the ADF. The proposed solution is to store the analysis ADF in the conditions database like any other conditions, and then to dump a directory structure in CVMFS (and in the on-line area for HLT) in an automatized way in order to keep the file access as it is today.
3. Prototype to manage auxiliary data files in the conditions database

For this prototype we have been gathering a subset of the ADF to verify the possibility to map the existing directory structure in the calibration area into the Tag and IOV based structure of the conditions data model. Some additional tables were added in order to store in the database the complete URI of the files uploaded. The strategy adopted in this first prototype was to map every file for a given calibration package in a Tag inside Oracle (see figure 2). Other possible mappings will be explored in future, to finally decide what is the optimal way of storing the data with the minimal amount of changes to be performed at the level of the software producing and or accessing these data.

Figure 2. Mapping of current calibration area to ADF dumped from Conditions Database.

3.1. Python client for analysis ADF

In order to provide the calibration experts with easy tools to manage their own data via the REST API, some special developments have been done based on an existing prototype for the conditions database (see figure 3). At client level we have established a minimal set of functions that are needed:
1) **calibcli add** `<package> <path> <file> <description>`
Add a file for a given software package and in a given path. The path will be used later to migrate the file under AFS. This action does not upload data, but it is used only to create the relevant meta-data for later management of the specified auxiliary data file.

2) **calibcli commit** `<package> <local file path> <dest path> <file tag extension> <dest file>`
Commit a local file into the conditions database. The file will be uploaded in this case, and stored in a tag using by default the IOV of 0.

3) **calibcli tag** `<package> <package tag> <file tag extension> <description>`
This will allow to associate a tag to a given package. This tag will reference all files uploaded for that package.

![Diagram](image)

**Figure 3.** ADF management via Conditions Database.

**4. Summary**
We have presented a proposal for investigating the management of ADF using the new Conditions Database architecture. A command line client to import the data into the new system was developed and query commands to navigate the data files will be added. This conditions area proved well suited for the new architecture. We are looking for other conditions areas that may immediately benefit from this architecture that we hope to implement with the CMS experiment for the LHC Run 3, planned for year 2020.

**References**

[1] The ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider *JINST* **3** S08003 doi:10.1088/1748-0221/3/08/S08003

[2] De Salvo A *et al* 2012 Software installation and condition data distribution via CernVM File System in ATLAS *J. Phys.: Conf. Series* **396** 032030 doi:10.1088/1742-6596/396/3/032030

[3] Barberis D *et al* 2015 Designing a future Conditions Database based on LHC experience *J. Phys.: Conf. Series* **664** 042015 doi:10.1088/1742-6596/664/4/042015