Evolution of ATLAS conditions data and its management for LHC Run-2

Michael Böhler[1], Mikhail Borodin[2], Andrea Formica[3], Elizabeth Gallas[4], and Voica Radescu[5] for the ATLAS collaboration

[1] Physikalisches Institut, Albert-Ludwigs-Universität, Hermann-Herder-Str. 3, 79104 Freiburg, Germany
[2]National Research Nuclear University MEPhI, Kashirskoe sh. 31, Moscow, 115409, Russia
[3]CEA/Saclay IRFU/SEDI, 91191 Gif-sur-Yvette , France
[4] Department of Physics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford, OX1 3RH, UK
[5] Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Im Neuenheimer Feld 226, 69120 Heidelberg, Germany

E-mail: michael.boehler@cern.ch

Abstract. The ATLAS detector at the LHC consists of several sub-detector systems. Both data taking and Monte Carlo (MC) simulation rely on an accurate description of the detector conditions from every subsystem, such as calibration constants, different scenarios of pile-up and noise conditions, size and position of the beam spot, etc. In order to guarantee database availability for critical online applications during data-taking, two database systems, one for online access and another one for all other database access, have been implemented.

The long shutdown period has provided the opportunity to review and improve the Run-1 system: revise workflows, include new and innovative monitoring and maintenance tools and implement a new database instance for Run-2 conditions data. The detector conditions are organized by tag identification strings and managed independently by the different sub-detector experts. The individual tags are then collected and associated into a global conditions tag, assuring synchronization of various sub-detector improvements. Furthermore, a new concept was introduced to maintain conditions over all different data run periods into a single tag, by using Interval of Validity (IOV) dependent detector conditions for the MC database as well. This allows on the fly preservation of past conditions for data and MC and assures their sustainability with software evolution.

This paper presents an overview of the commissioning of the new database instance, improved tools and workflows, and summarizes the actions taken during the Run-2 commissioning phase in the beginning of 2015.

1. Introduction

The ATLAS [1] detector conditions data encompass a wide variety of information which characterize the state of all ATLAS subsystems during specific intervals essential for data taking and/or event processing. This information is stored in the ATLAS database (DB) system based on *LCG Conditions DB infrastructure* [2]. The databases are separated into two categories: one explicitly used for event processing of data collected by the ATLAS experiment, in the following referred to as 'data', and another one used for generating Monte Carlo simulations of the events expected to be recorded by the experiment, in the following referred to as 'MC'.

Based on the experience of the LHC data taking period from 2009-2013 (Run-1), several features of the ATLAS conditions database have been exploited, and customized ATLAS tools for conditions entry, manipulation, collection, and verification have been developed. In order to cope with the challenges of Run-2 a complete new DB instance for data conditions has been implemented explicitly for Run-2 data conditions. The MC conditions DB instance from Run-1 will still be used for Run-2. The MC conditions are now arranged in a structure of intervals of validity (IOV) in order to achieve an on the fly preservation of past MC conditions. This assures their sustainability with software evolution and reduces the amount of duplicated information.

This paper gives an overview of the management of the detector conditions, updated tools and workflows, the implementation of a new conditions database instance for Run-2 data, and the IOV based MC conditions.

2. Concept of global conditions tags

There is a long list of conditions data which need to be recorded and provided during data processing for precise event reconstruction and an accurate MC description:

- alignment
- beam position
- magnetic field map
- cabling
- calibration
- corrections
- channel offset
- detector status
- noise
- pulse shapes
- timing
- ...

This is just a small subsample of hundreds of parameters which are stored in the ATLAS conditions DB. Since the data describes the conditions of several subsystems, the conditions management is grouped logically by physical subsystem, closely following the description of the ATLAS detector:

- Calorimeter: The calorimeter conditions are grouped in LAR for liquid-argon electromagnetic sampling calorimeters, TILE for tile calorimeters, and CALO for global calorimeter conditions.
- Muon spectrometer: The muon related conditions are grouped as their subsystems: Monitored Drift Tube chambers (MDT), Cathode-Strip Chambers (CSC), Resistive Plate Chambers (RPC), Thin Gap Chambers (TGC). The muon spectrometer alignment conditions are collected in MUONALIGN.
- Tracking: The innermost pixel detector conditions are arranged in PIXEL, the silicon micro-strip detector conditions in SCT, the transition radiation tracker conditions in TRT, and global inner detector conditions e.g. inner detector alignment and beam spot conditions are grouped in INDET.
- Other: The last category contains the Trigger and Data Acquisition (TDAQ) system conditions data, further trigger conditions (TRIGGER), the conditions of the forward detectors (FWD), data from the Detector Control System (DCS) and GLOBAL conditions, e.g. detector status defects.

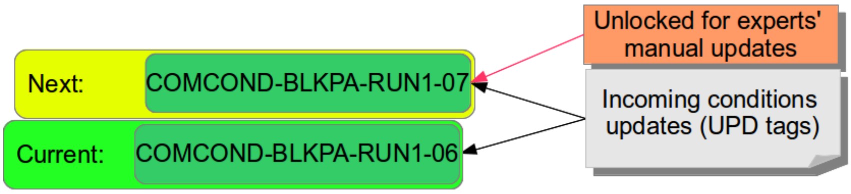

Figure 1. Dynamic conditions collection of the *Current-Next-Mechanism*. Whereas the *Current tag* can only be updated automatically for future IOVs, the *Next tag* can be modified manually by conditions experts.

Some of these conditions, e.g. cabling configurations of sub-detector components and specific noise thresholds, are immutable values and have to be stored once in the DB and never changed. Other conditions like alignment parameters need to be updated on the fly. These conditions are stored in (UPD) tags, which can be updated for new IOVs.

The detector conditions are organized by tag identification strings and managed independently by the different sub-detector experts. The individual tags are then collected and associated to a global conditions tag (GT) to assure that improved conditions descriptions of the various sub-detectors are synchronized. This allows the user to choose a specific conditions setup for the complete ATLAS detector by selecting one specific global conditions tag name. The so-called 'Best Knowledge' (BK) global tag describes the latest stable conditions configuration. This model gives experts a well-ordered framework for tag management, provides a standardization over time, and ensures the reproducibility of data processing.

Since a running experiment demands a dynamic conditions collection, conditions for new IOVs must be added continuously and automatically; on the other hand, there must be a mechanism to allow conditions updates from subsystem experts as well. This is guaranteed by the *Current-Next-Mechanism*. The *Current* global conditions tag is used as "in production" for ATLAS data processing and it is updated for new IOVs automatically, whereas the *Next* global conditions tag gets the same automatic updates from the calibration loop, but can be modified by system experts, since it is used as "development" tag (see Fig.1).

In preparation for Run-2, the status of the global conditions tag was added to the COMA (Conditions/Configuration Meta-data for ATLAS) DB [3]. This allows tracking of the evolution of a given GT, when it is set to *Next* or *Current*. This information is added to the COMA DB via the ATLAS Meta-data Interface (AMI) [4, 5] interface. A dedicated web page *COMA Global Tags* was added to the AMI web portal such that the tag coordinators can perform this DB insertion via an easy-to-use web interface.

3. Workflow of global tags coordination

The global conditions tags are collections of subsystem conditions. For real data conditions, the global conditions tags need to be updated when non-UPD conditions need to be updated. Since no continuous update of MC conditions is foreseen, the MC global conditions tag needs to be rebuilt whenever new conditions are necessary for MC production.

First, the subsystem experts are notified by the tag coordination that a new global conditions tag is planned to be built, and that new conditions can be associated to it. This communication is documented via the ticketing system JIRA [6]. Two conditions monitoring web interfaces are available to visualize the current status of the conditions database: *CoolTagBrowser* [7] and the COMA TagReport. When all subsystem conditions tags are available they can be added to the global conditions tags. Dedicated Python scripts developed by ATLAS are used to perform the necessary DB manipulations. The global tag and all associated subsystem tags are locked, so that all conditions data are frozen for a given global conditions tag. The newly created global

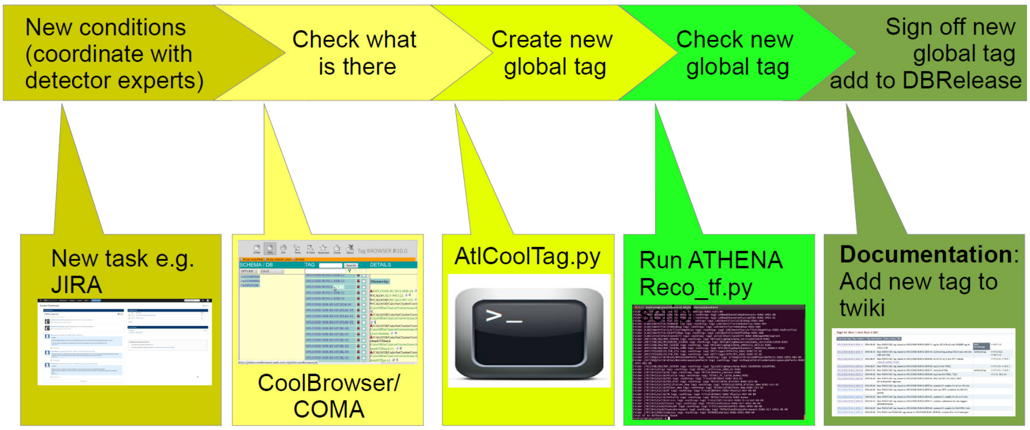

Figure 2. Workflow of global tags coordination. After conditions updates are requested, subsystem experts provide new conditions and upload them to the conditions DB. Tag coordination can monitor these updates with the *CoolTagBrowser* and the COMA Tag Report. A new GT is then build with dedicated Python scripts. After technical validation GT is signed-off.

conditions tag is tested first on the DB level for consistency and then by running a typical ATLAS data reconstruction or MC production job for a technical validation. In the case of MC global tags, a physics validation is also requested, where MC samples with larger statistics are produced and distributions of physics parameters are compared in detail. After any kind of deviation from the previous conditions is understood, the global conditions tag is signed-off and can be used for large scale MC production. The complete workflow is sketched in Fig. 2.

4. ATLAS conditions for data

Experience has been gained from the Run-1 conditions DB instance *COMP200*. It worked reliably and impressive physics results have been achieved in Run-1. Nevertheless, a detailed review of the existing DB instance showed that there are some folders which are large, not only due to Run-1 conditions data volume, but also test beam and commissioning conditions are stored there. Restructuring was needed because the large amount of old, partially obsolete conditions data made that instance hard to maintain for the long term, and the historically-grown structure made it difficult for incoming experts to understand. It was then decided to gain from the Run-1 experience and build up a complete new DB instance dedicated to Run-2 data. A folder classification was set up, such that every subsystem expert had to classify every existing folder tag as *Run1Only*, *Obsolete*, or *AllRuns* category. For the last category *AllRuns*, the conditions structure was then copied to the new DB instance called *CONDBR2*.

Since the schema of the conditions was not changed at all, only small changes to the software were needed. Whenever data events are processed by the ATLAS reconstruction software (ATHENA), the *IOVDB Service* needs to distinguish whether the event belongs to Run-1 or Run-2, so that it can look up the conditions either in *COMP200* or in *CONDBR2*. This mechanism is sketched in Fig. 3.

Three different global conditions tags need to be maintained for ATLAS data taking: for the High Level Trigger (HLT), for the prompt express stream (ES1) and for the first pass bulk processing (BLK) as shown in Fig. 4. This concept was used in Run-1, it has proven to be very reliable, and it provides most precise and up-to-date conditions information for prompt data reconstruction.

The High Level Trigger provides a fast online reconstruction of the complete event. Based

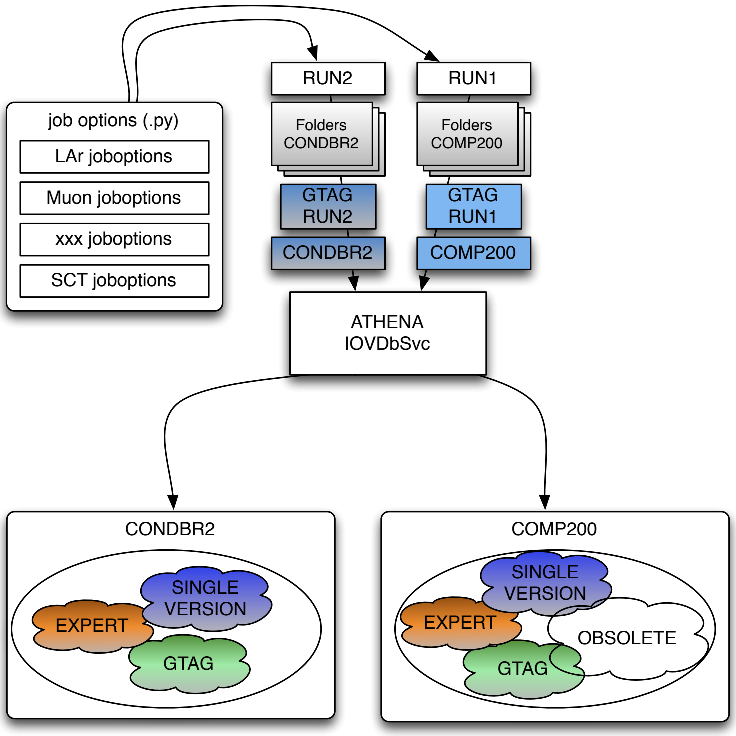

Figure 3. Software access of the new DB model with two separated DB instances for Run-1 and Run-2, respectively.

on the online reconstruction, a collision event is selected for further reconstruction and stored on disk or tape. The HLT explicitly uses online conditions stored in the online DB. If the event cannot be reconstructed by the HLT due to any kind of failure, the event cannot be recorded and is lost. Therefore the online DB has to have maximal availability.

The offline reconstruction processed at the ATLAS Tier-0 (the computing farm based next to the ATLAS experiment at CERN) has two further reconstruction phases:

- Prompt express stream processing: A dedicated global conditions tag (CONDBR2-ES1PA-2015-01 in Fig.4) is used in this step. It processes a subsample of 10% of the total data stream and provides, together with a dedicated calibration stream, new conditions data which is loaded to the offline conditions DB and then used in the next step from the BLK tags.
- First pass bulk reconstruction processing: The folder tags associated to the global conditions tags used for BLK processing, are either non-UPD tags, if their conditions does not need to be updated continuously, or UPD tags, which receive the automatic conditions updates derived in the calibration loop.

Figure 4 shows which GT accesses the online and offline DB instances. Furthermore, it highlights which GT is used at each point of the data preparation workflow.

5. ATLAS conditions in Monte Carlo production

The MC conditions must not access the online conditions DB. The online conditions are exclusively reserved for online applications like the High Level Trigger. This is a very important

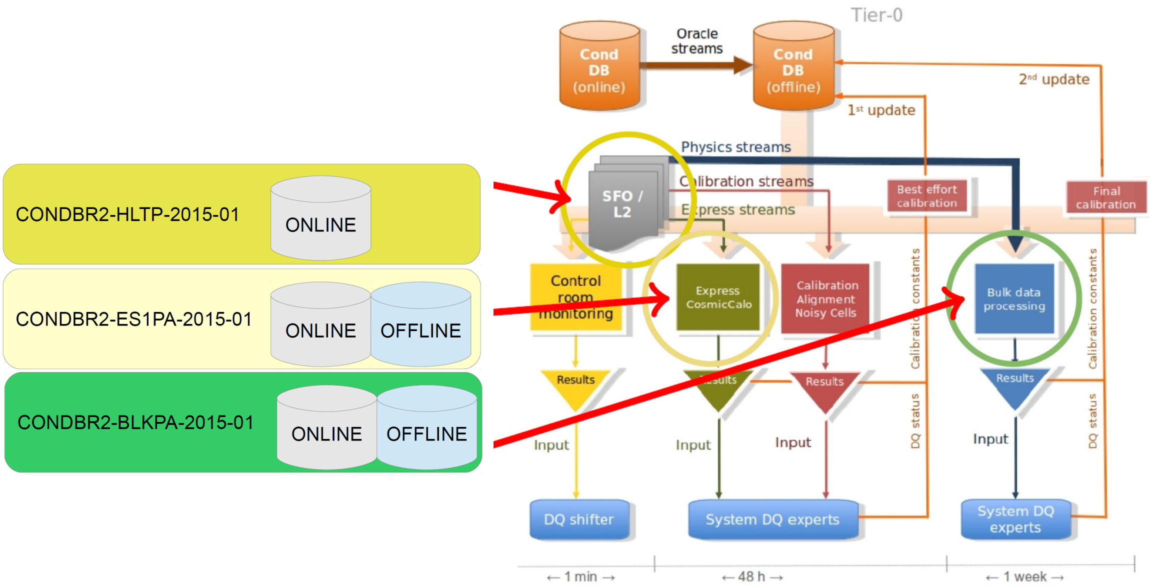

Figure 4. The three different global conditions tags needed for ATLAS data processing - the *HLTP* tag used by the High Level Trigger, the *ES1PA* tag used by the express stream, and the *BLK* tag used in the bulk reconstruction needed for the ATLAS data processing - are shown on the left, indicating the access of the online and the offline conditions DBs. On the right side it is highlighted which GT is used at which step of the ATLAS data preparation schema.

aspect to guarantee the DB availability for critical online applications during data-taking. Since the original ATLAS conditions database was based on the "online" DB and the separation between online and offline was introduced after some time, most of the MC conditions have been moved from online to offline DB. A small subset of the MC conditions could not be moved during Run-1, because of software dependencies. The long shutdown period has provided the opportunity to finalize the relocation of the folders from the online DB to the offline DB: necessary conditions were copied from the online DB to the MC offline DB and corresponding subsystem tags were associated to all global conditions tags. The online dependency was solved by removing the online folders from the new MC global conditions tags. This simplifies the coordination of the MC conditions. After the restructuring all MC conditions are hosted on the offline servers and no online folders need to be maintained for Run-2 MC conditions anymore. Though this solution breaks the backward compatibility of software-conditions interplay, since old software tries to access online folders which are no longer available in new conditions tags, this was unavoidable because the strict separation of online DB from grid production was more important than keeping the oldest software releases synchronized with the global tag evolution.

The most important improvement for the MC global conditions tag was the introduction of the IOV dependence. This allows the collection of conditions designed for completely different environments in one single global conditions tag, just separated by the IOV range (either by run number or by time stamp). The concept of IOV dependent MC conditions also permits any new global conditions tag to contain the IOV specific conditions from every earlier runs and periods. This introduces a conditions preservation "on the fly", meaning that only the conditions for new IOVs are updated while the past IOVs are kept unchanged. In Fig. 5 it is shown how all MC conditions describing the different data taking periods are collected together.

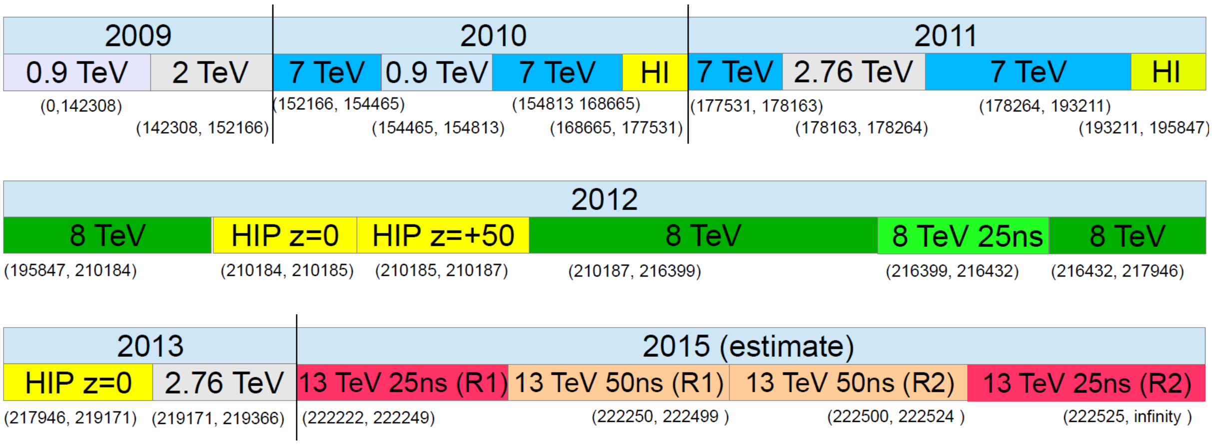

Figure 5. IOV dependent MC conditions data. A time-line split in three lines (for better visualization) shows for each period three pieces of information: (1) year when the corresponding data was recorded, (2) center of mass energy of pp collisions, or the Heavy Ion setup (3) valid IOV Range (here based on the run number).

The new releases of the *CoolTagBrowser* (release 11 and newer) also contain a IOV per channel display, such that the IOV ranges, in data as well as in MC conditions, can be monitored easily with the same tool used for monitoring the GT building process. A screen shot of *CoolTagBrowser* release 11.08 is shown in Fig. 6. The IOV ranges per channel are listed in the table on the bottom in the middle and visualized in the chart in the right column.

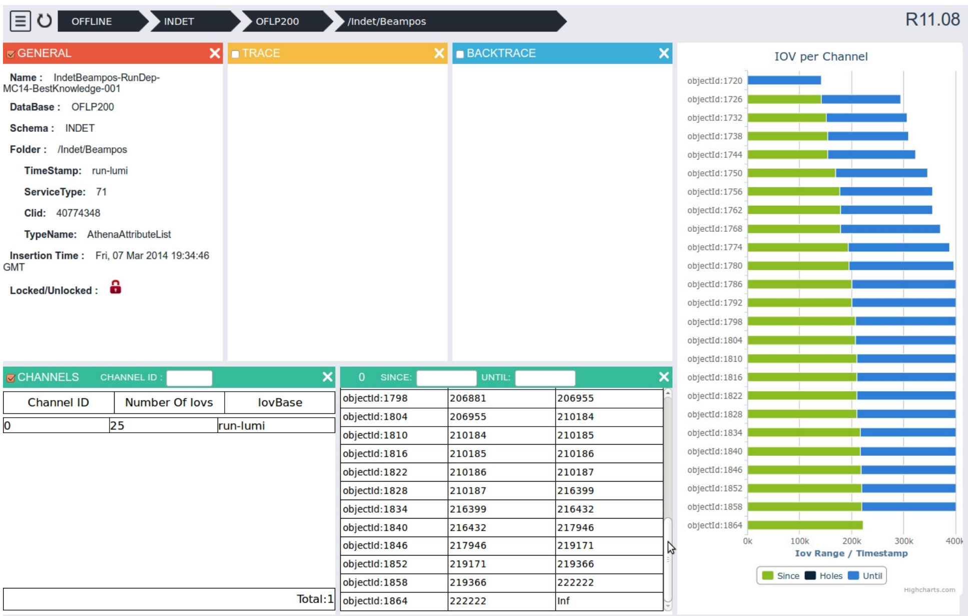

Figure 6. IOV ranges are displayed in the latest *CoolTagBrowser* version 11.08. The information is published as IOV range per channel on the bottom of the screen shot, as a table in the middle, and as a chart in the right column.

6. Summary and conclusion

The detector conditions are needed both for data reconstruction and MC production. The long shutdown period has provided the opportunity to review and improve the Run-1 system: A new data conditions DB has been implemented, which contains the three different global conditions tags needed for the ATLAS prompt data processing scheme: the HLT global tag for the High Level Trigger, the ES1 global tag for the express stream, and the BLK global tag for the first pass bulk processing. The workflows have been reviewed and improved as documented for the conditions tag production workflow. The monitoring tools have been improved, e.g. *CoolTagBrowser* can display the IOVs and the *COMA tag browser* was used for the schema classification, needed to copy only classified folder tags to the new conditions DB.

The IOV dependent MC global conditions tag allows one single tag for different environments and provides on the fly preservation of past conditions for data and MC and assures their sustainability with software evolution.

References

[1] ATLAS Collaboration 2008 *JINST* **3** S08003
[2] A. Valassi *et al.* 2008 *In Nuclear Science Symposium Conference Record* NSS 08. IEEE, pages 3021-3028
[3] E J Gallas *et al.* on behalf of the ATLAS Collaboration 2014 *J. Phys.: Conf. Ser.* **513** 042020
[4] Albrand S, Doherty T, Fulachier J and Lambert F 2008 *J. Phys.: Conf. Ser.* **119** 072003
[5] J Fulachier *et al.* on behalf of the ATLAS Collaboration 2014 *J. Phys.: Conf. Ser.* **513** 042019
[6] JIRA v6.3.13 webpage: https://www.atlassian.com/licensing/jira-agile
[7] A Sharmazanashvili *et al.* on behalf of the ATLAS Collaboration 2014 *J. Phys.: Conf. Ser.* **513** 042040