Utility of collecting metadata to manage a large scale conditions database in ATLAS

E J Gallas[1], S Albrand[2], M Borodin[3], and A Formica[4]
on behalf of the ATLAS Collaboration

[1] Department of Physics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford, OX1 3RH, United Kingdom
[2] Laboratoire de Physique Subatomique et Corpusculaire, Université Joseph Fourier Grenoble 1, CNRS/IN2P3, INPG, 53 avenue des Martyrs, 38026, Grenoble, FRANCE
[3] National Research Nuclear University MEPhI, Kashirskoe sh. 31, Moscow, 115409, Russia
[4]CEA/Saclay IRFU/SEDI, 91191 Gif-sur-Yvette , France

E-mail: gallas@cern.ch

Abstract.

The ATLAS Conditions Database, based on the LCG Conditions Database infrastructure, contains a wide variety of information needed in online data taking and offline analysis. The total volume of ATLAS conditions data is in the multi-Terabyte range.

Internally, the active data is divided into 65 separate schemas (each with hundreds of underlying tables) according to overall data taking type, detector subsystem, and whether the data is used offline or strictly online. While each schema has a common infrastructure, each schema's data is entirely independent of other schemas, except at the highest level, where sets of conditions from each subsystem are tagged globally for ATLAS event data reconstruction and reprocessing.

The partitioned nature of the conditions infrastructure works well for most purposes, but metadata about each schema is problematic to collect in global tools from such a system because it is only accessible via LCG tools schema by schema. This makes it difficult to get an overview of all schemas, collect interesting and useful descriptive and structural metadata for the overall system, and connect it with other ATLAS systems. This type of global information is needed for time critical data preparation tasks for data processing and has become more critical as the system has grown in size and diversity.

Therefore, a new system has been developed to collect metadata for the management of the ATLAS Conditions Database. The structure and implementation of this metadata repository will be described. In addition, we will report its usage since its inception during LHC Run 1, how it has been exploited in the process of conditions data evolution during LS1 (the current LHC long shutdown) in preparation for Run 2, and long term plans to incorporate more of its information into future ATLAS Conditions Database tools and the overall ATLAS information infrastructure.

1. Issues in Conditions Management

"Conditions" is a general term in any experiment for information which is not event-wise, reflecting the conditions or states of a system valid for a time interval ranging from very short to infinity. ATLAS has conditions-like data from a wide variety of subsystems ranging, as examples, from beam related quantities from the LHC, to online detector control, configuration and data acquisition systems, to subdetector calibrations and system alignment (both online and offline). The criticality of the data cannot be overstated: ATLAS event-wise data cannot be meaningfully recorded or processed without the conditions data associated with it.

ATLAS stores its conditions in the ATLAS Conditions Database. It is based on the LCG Conditions Database infrastructure [1]. We use the LCG COOL API [2] to interact with the Conditions Database, so it has become a common colloquialism in ATLAS to refer to the database as the "COOL DB", a convention which has been adopted in this document. For robustness, the master copy of the database is deployed in Oracle RAC clusters [3] at CERN, with data needed for grid processing replicated to select ATLAS Tier-1s via Oracle Streams.

The infrastructure's inherent IOV-based (Interval Of Validity) structure makes it ideal for the primary use cases of the COOL DB: Data taking, processing and monitoring. In addition, it has been very useful to have all conditions data in a common infrastructure:

- It enables us to build develop central tools for system-specific conditions management.
- Structural uniformity was a key factor in our successful deployment of Frontier infrastructure [4], making conditions data readily available to jobs on the grid.

The ATLAS Conditions Database, by the end of LHC Run 1 is both large (now many TB of data) and diverse (65 active schemas). These schemas are owned by 17 subsystems, storing data in 3 active instances for LHC Run 1:
1) Simulation and 2) Real Data, both replicated to Tier 1s and 3) Real Data monitoring;
and in 2 domains: 1) Used Online and 2) Used for Offline processing (not used Online).

As a whole these active schemas now own nearly 1400 **folders** (1400 database tables storing the data). These folders vary greatly in content, size and complexity (number of payload columns varying from 1 to 265, and with many orders of magnitude larger variation in their data volumes): their designs are dictated by the nature of the data and the subsystem use cases. Folder evolution over Run 1 has been considerable, as subsystems came to a better understanding of their data. Folder changes under this evolution generally resulted in new folders being defined, with obsolete folders being retained for backwards compatibility during the transition as well as ensuring that potentially useful legacy data not be lost. Thus, many folders are known now to be obsolete in the Run 1 instances.

Folders are defined to be either single-version or multi-version. Multi-version folders are used when different versions of conditions are expected in the same IOV ranges (for example: calibrations which may be refined over time). Currently, the multi-version folders own over 15000 **folder tags** (versions of distinct sets of conditions).

Folder tags are collected across schemas into **global tags** for ATLAS event processing. There are currently over 600 global tags in Run 1 instances, only a fraction of which have ever been used in standard data processing. Others may have been used in non-standard processing, so they cannot be simply deleted. Similarly, just because a folder tag is never included in a global tag does not mean that it has never been used. Thus, many global and folder-level tags are known now to be obsolete in the Run 1 instance, but, like obsolete folders, must be kept for posterity.

ATLAS Global Tagging procedures reached maturity during Run 1. It is now believed that a single global tag for data and simulation, respectively, can be consolidated for any future Run 1 analysis. Work is well underway during **LS1** (the current LHC long shutdown) to realize this outcome, which will greatly simplify the COOL DB for Run 2, as we discuss in later sections.

2. Motivation and Goals

While basing our conditions storage on the LCG infrastructure has served us well in many regards as described in the previous section, the system has a number of limitations:

1) The COOL API has the limitation that one must access the data by schema. This presents difficulties in developing systems which offer any kind of overview of the COOL DB.
2) It is difficult to find information without detailed subsystem-specific knowledge which impacts coordination efforts and makes it difficult for new people adopting subsystem responsibilities in the conditions area to understand the system.
3) The infrastructure does not easily allow us to enhance global content with ATLAS specific information or commonly needed metrics (such as characteristics of folders or data volume associated with specific Tags).
4) The infrastructure does not easily allow us to connect dynamically with other ATLAS systems.

Therefore, a dedicated repository has been developed to collect metadata on ATLAS COOL DB structure to help fill the gap. The goals of this new system are to:

1) enhance functionality of the ATLAS COOL Tag Browser [5], a tool for conditions tag management in ATLAS,
2) collect structural metadata about content, such as about folder channels, columns, rows, volume, and which data changes the most (or the least),
3) understand gaps in IOV coverage (gaps in conditions w/time) in folder tags,
4) easily find which folders use external references and investigate the uniqueness of those references,
5) offer a global view of COOL DB structure and show collected metadata for folders and tags via web-based interfaces,
6) connect conditions references to other ATLAS systems: For example, to identify which conditions are or are not used in event-wise processing,
7) store and display which sets of conditions are current or in preparation, and
8) assist in the general conditions cleanup during LS1 (the current post-Run 1 Long Shutdown) in preparation for LHC Run 2 operations.

The system being described here is an extension of the ATLAS COMA system [6] described at CHEP in 2012. This development has followed the same database and interface design principles (described in that presentation, so they will not be repeated here).

Ties between AMI (ATLAS Metadata Interface) [7] with COMA have broadened into this new area of COOL DB management, sharing information and resources to provide more coherent services to users while optimizing effort and infrastructure. Some of these connections are described explicitly in this document.

3. Schema

A simplified overview of the database schema used to collect metadata about the structure of the COOL DB is shown in Figure 1.

The schema currently contains 15 tables and 4 views, with its design being driven fundamentally by the COOL DB structure: It stores data in tables (called folders), so central to the metadata schema is the Folders table (green tables of Figure 1), which all other tables relate to directly or indirectly. Each folder is owned by a specific Schema (purple), each of which is characterized by its subsystem, instance, and if it is used offline or strictly online. Multi-version folders have one or more Folder Tags (blue) for conditions that require different versions over IOVs. Folder Tags may be included in one or more Global Tags (yellow) when they are designated to be used in event-wise processing. The database has additional tables to store metrics (structural metadata about folders and folder tags) and enhance content related to global tags (which is derived from other ATLAS systems).

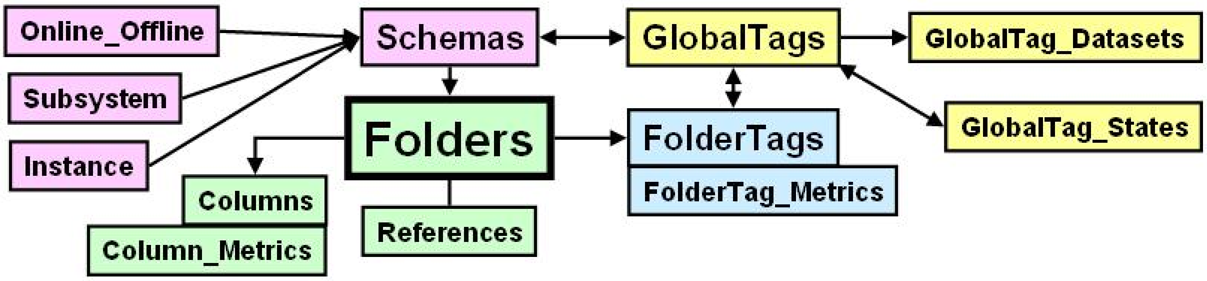


Figure 1. Database schema for ATLAS Conditions Database metadata: Relations between the entities are represented by lines, arrows, or multi-directional arrows for one-to-one, one-to-many, or many-to-many relationships, respectively.

4. Data Loading

Sources of the metadata include the COOL DB itself, derived content from the AMI database, TWiki pages and other documentation, and expert entry via customized interfaces. The conditions infrastructure contains no mechanism to notify external systems of changes (nor is that necessarily desirable), so synchronizing the metadata with the COOL DB is challenging because of the volume of the data and because changes can occur in any of the schemas. This is accomplished, in some cases, by direct queries to underlying COOL DB tables when COOL API methods are not available to retrieve the desired information or when those methods are slower than direct queries.

Cross checks are performed as new data is inserted or updated. Examples of cross checks which have identified inconsistencies include:

- Global Tag Descriptions and Lock Status, which are stored in the COOL DB schema-wise, must be identical across all schemas when that Tag is locked for production processing.
- Folder definitions contain XML which must conform to set standards if the conditions in those folders need to be accessed by Athena (the ATLAS event processing framework).

When inconsistencies are found, experts are notified to correct the issue.

The global tags area is of keenest interest to other areas in the experiment because these distinct collections of conditions are used in data taking and offline processing, so we collect additional metadata in two areas related to global tags:

1) Special tables are defined and populated with data from the AMI database relating which global tags are used to process event-wise datasets: of particular interest is the project name in which that data belongs and the range in dates of processing. An AMI task refreshes the **Global Tag Datasets** table approximately once per day.
2) Special tables are also defined to store **Global Tag State** designations. These designations vary in time as the experiment evolves. They are declarations by experts that particular global tags are designated to be currently the “Best Knowledge” (BK) tag for event-wise processing (State = “Current”) or as the tag under preparation to be Current in the future (“Next”). There are 3 flavors depending on their domain of usage:
   a) Online data taking (HLT): the current tag in use for online data taking
   b) Express Stream processing (ES): used in quasi-real time processing of the latest data
   c) Offline processing (no suffix): used for all offline bulk data processing

   Putting States into a database makes them robustly available to many external systems needing this information (previously stored in AFS files). These States are entered by COOL Tag experts into these tables via an entry interface developed by the AMI Team.

4.1. Structural Metadata

Metadata in this system can in general be classified as either descriptive or structural. An example of the former being the names of folder tags included in a global tag or their properties, while the latter would be a count of those folder tags. The reasoning behind collecting structural metadata (metrics) in this application is generally to gauge the scale or extent of aspects of the COOL DB to help make decisions about which areas of the system would most benefit from optimization.

LS1 offers an excellent opportunity to make changes to the system without worry about affecting real-time data taking. In fact, many known changes to the COOL DB, as well as to the LCG Conditions Database infrastructure, had been postponed until this shutdown to avoid potential disruption to Run 1 operations. Now during LS1, a major cleanup is underway, using the knowledge gained during the Run 1 experience both at the subsystem level as well as to refine the system globally to start Run 2 on the best possible footing.

The three "active" instances of the COOL DB contain all data conditions stored over the last 5 years (including all of Run 1). Some underlying COOL DB tables have become so large that it is unimaginable to add Run 2 data without expecting degradation in performance (some already seen at the end of Run 1). This, combined with the number of obsolete folders and tags form the main motivation to create new instances for Run 2.

We are now in the process of refining folder definitions at the subsystem level and the consolidation of tags, so the Run 2 instances will start off containing a minimum of obsolete data and tags. Data in the largest folders will be retained in the Run 1 instances, with only their definitions being copied into the Run 2 instances: This allows those folders to make a clean start in Run 2. It is critical to retain Run 1 data processing capacity, so modifications are being made to Athena to point algorithms to the correct instance when processing any of Run 1 data.

The metadata system has been very useful in this consolidation process. As an example: Folder payload can be a "reference" to external files, rather than storing data "inline" within the database. But external files have been problematic:

- Online: file movement around the online firewall is problematic, requiring special infrastructure, sometimes need expert intervention, and can cause delays.
- Offline: files must be delivered to worker nodes for jobs on the grid.

Pre-Run 1 thinking was that external references would be the best way to store larger data objects. But after improvements in server performance and network bandwidth, inline storage is decidedly more performant and less problematic than managing file movement. So a dedicated effort has been put forth during LS1 to try to move from the use of external references toward inline storage where ever possible.

Using the metadata, it was easy to identify at coordination level which folders use external references by subsystem (208 in 5 subsystems) and similarly easy to identify which of those are used in current global tags (99 in the current BK tag). Evaluating the uniqueness of their content, it is found that some data did not change as anticipated or was of smaller volume than anticipated. Subsystem experts re-evaluated storage options. In some cases, good reasons were found for external files but in other cases, subsystems agreed that inline payload is better and are in the process of redefining these folders for Run 2 (moving references to inline content).

4.2. Ongoing Improvements

Currently, metadata is completely synchronized with sources approximately once per day since the main program requires about an hour to execute. Work is ongoing to speed up the synchronization process while adding additionally useful metrics. The programs are being split into faster (more critical) and slower (less critical) parts, so that critical components can be updated more often. We are also employing a new API: a RESTful service (Java wrapper) in a

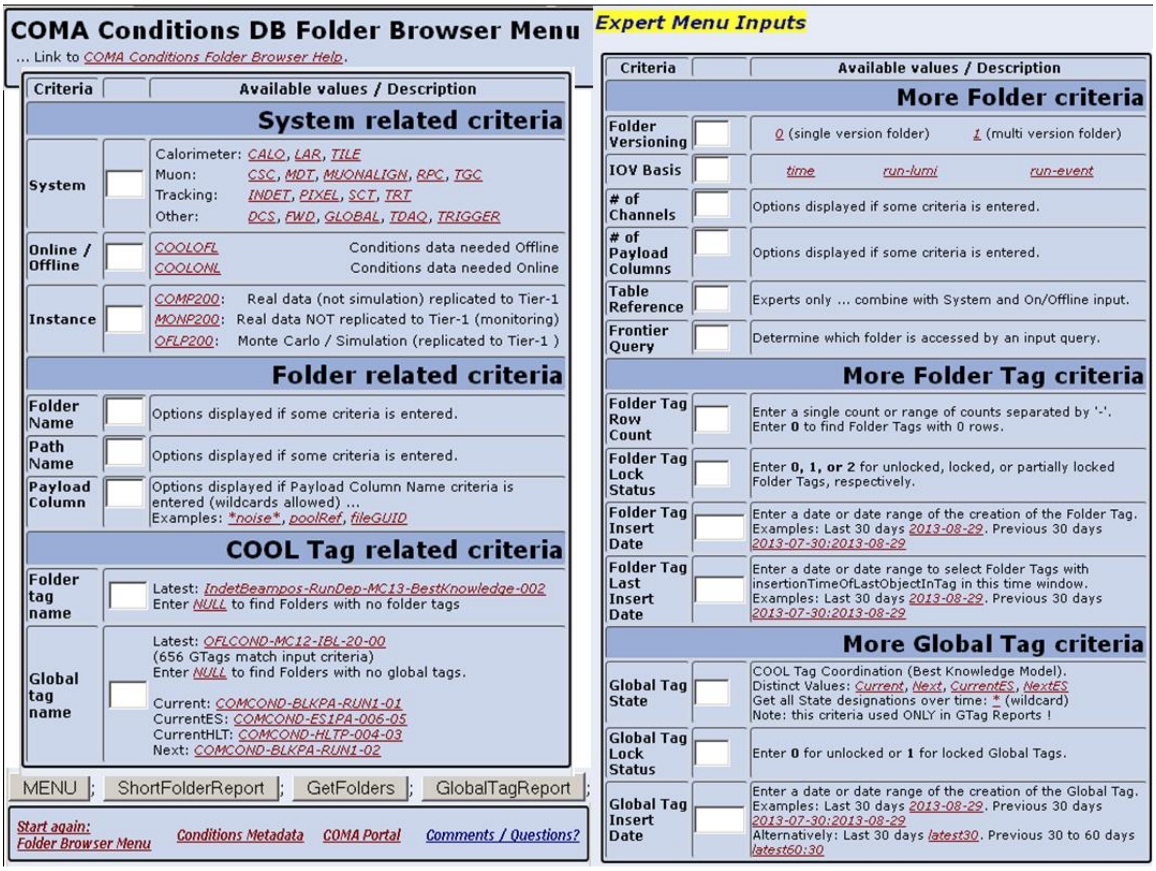
JBoss server, which obtains new metrics (not available via the COOL API) through dedicated direct PL/SQL.

We are also looking into expanding the schema to include bookkeeping details of changes made by subsystem experts using ATLAS specific tools. These tools, generally in python, are outside the LCG infrastructure, so they can add metadata content directly as experts execute those tools.

5. Interfaces

Any metadata within the system can be made available programatically via pyAMI, the AMI python client. We describe here a number of web based interfaces which have been deployed to browse and report the metadata, giving users an overview of the COOL DB and display information about its definition and structure.

The COMA Conditions DB Folder Browser is a dynamic web interface allowing users to find folders and tags by applying any of a wide variety of selection predicates shown in Figure 2. The browser itself is dynamic, with the ability to apply the user's selection criteria and show remaining possibilities of other selection criteria. At each iteration, the menu shows the number of folders and tags meeting the criteria.



COMA Conditions DB Folder Browser Menu

... Link to COMA Conditions Folder Browser Help.

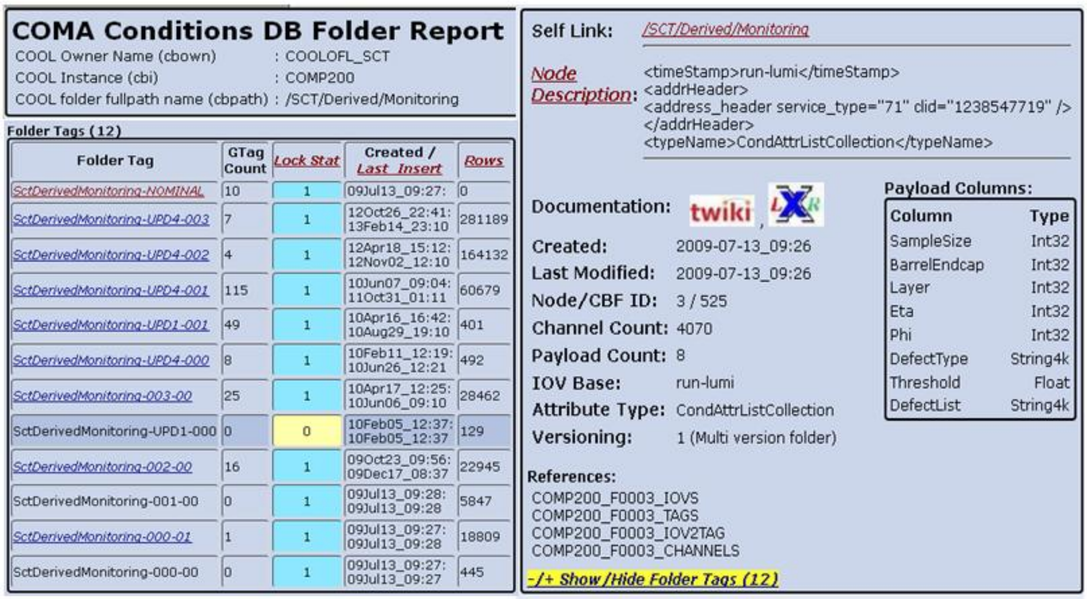
Criteria		Available values / Description
		System related criteria
System	□	Calorimeter: CALO, LAR, TILE Muon: CSC, MDT, MUONALIGN, RPC, TGC Tracking: INDET, PIXEL, SCT, TRT Other: DCS, FWD, GLOBAL, TDAQ, TRIGGER
Online / Offline	□	COOLOFL Conditions data needed Offline COOLONL Conditions data needed Online
Instance	□	COMP200: Real data (not simulation) replicated to Tier-1 MONP200: Real data NOT replicated to Tier-1 (monitoring) OFLP200: Monte Carlo / Simulation (replicated to Tier-1)
		Folder related criteria
Folder Name	□	Options displayed if some criteria is entered.
Path Name	□	Options displayed if some criteria is entered.
Payload Column	□	Options displayed if Payload Column Name criteria is entered (wildcards allowed) ... Examples: *noise*, poolRef, fileGUID
		COOL Tag related criteria
Folder tag name	□	Latest: IndetBeampos-RunDep-MC13-BestKnowledge-002 Enter NULL to find Folders with no folder tags
Global tag name	□	Latest: OFLCOND-MC12-IBL-20-00 (656 GTags match input criteria) Enter NULL to find Folders with no global tags. Current: COMCOND-BLKPA-RUN1-01 CurrentES: COMCOND-ES1PA-006-05 CurrentHLT: COMCOND-HLTP-004-03 Next: COMCOND-BLKPA-RUN1-02

MENU ; ShortFolderReport ; GetFolders ; GlobalTagReport ;
Start again: Folder Browser Menu
Conditions Metadata
COMA Portal
Comments / Questions?
Expert Menu Inputs

Criteria		Available values / Description
		More Folder criteria
Folder Versioning	□	0 (single version folder) 1 (multi version folder)
IOV Basis	□	time run-lumi run-event
# of Channels	□	Options displayed if some criteria is entered.
# of Payload Columns	□	Options displayed if some criteria is entered.
Table Reference	□	Experts only ... combine with System and On/Offline input.
Frontier Query	□	Determine which folder is accessed by an input query.
		More Folder Tag criteria
Folder Tag Row Count	□	Enter a single count or range of counts separated by '-'. Enter **0** to find Folder Tags with 0 rows.
Folder Tag Lock Status	□	Enter **0, 1, or 2** for unlocked, locked, or partially locked Folder Tags, respectively.
Folder Tag Insert Date	□	Enter a date or date range of the creation of the Folder Tag. Examples: Last 30 days 2013-08-29. Previous 30 days 2013-07-30:2013-08-29
Folder Tag Last Insert Date	□	Enter a date or date range to select Folder Tags with insertionTimeOfLastObjectInTag in this time window. Examples: Last 30 days 2013-08-29. Previous 30 days 2013-07-30:2013-08-29
		More Global Tag criteria
Global Tag State	□	COOL Tag Coordination (Best Knowledge Model). Distinct Values: Current, Next, CurrentES, NextES Get all State designations over time: * (wildcard) Note: this criteria used ONLY in GTag Reports !
Global Tag Lock Status	□	Enter **0** for unlocked or **1** for locked Global Tags.
Global Tag Insert Date	□	Enter a date or date range of the creation of the Global Tag. Examples: Last 30 days 2013-08-29. Previous 30 days 2013-07-30:2013-08-29 Alternatively: Last 30 days latest30. Previous 30 to 60 days latest60:30

Figure 2. Conditions DB Folder Browser

When the user narrows the selection sufficiently for his/her use case, they can then generate reports about those folders or tags. An example of a folder report is shown in Figure 3 which shows general properties of the selected folder and its payload, its folder tags and their association with global tags, and offers links to related COMA reports and external documentation.



COMA Conditions DB Folder Report

COOL Owner Name (cbown) : COOLOFL_SCT
COOL Instance (cbi) : COMP200
COOL folder fullpath name (cbpath) : /SCT/Derived/Monitoring

Folder Tags (12)

Folder Tag	GTag Count	Lock Stat	Created / Last Insert	Rows
SctDerivedMonitoring-NOMINAL	10	1	09Jul13_09:27:	0
SctDerivedMonitoring-UPD4-003	7	1	12Oct26_22:41: 13Feb14_23:10	281189
SctDerivedMonitoring-UPD4-002	4	1	12Apr18_15:12: 12Nov02_12:10	164132
SctDerivedMonitoring-UPD4-001	115	1	10Jun07_09:04: 11Oct31_01:11	60679
SctDerivedMonitoring-UPD1-001	49	1	10Apr16_16:42: 10Aug29_19:10	401
SctDerivedMonitoring-UPD4-000	8	1	10Feb11_12:19: 10Jun26_12:21	492
SctDerivedMonitoring-003-00	25	1	10Apr17_12:25: 10Jun06_09:10	28462
SctDerivedMonitoring-UPD1-000	0	0	10Feb05_12:37: 10Feb05_12:37	129
SctDerivedMonitoring-002-00	16	1	09Oct23_09:56: 09Dec17_08:37	22945
SctDerivedMonitoring-001-00	0	1	09Jul13_09:28: 09Jul13_09:28	5847
SctDerivedMonitoring-000-01	1	1	09Jul13_09:27: 09Jul13_09:28	18809
SctDerivedMonitoring-000-00	0	1	09Jul13_09:27: 09Jul13_09:27	445

Self Link: /SCT/Derived/Monitoring

Node Description:

```
<timeStamp>run-lumi</timeStamp>
<addrHeader>
<address_header service_type="71" clid="1238547719" />
</addrHeader>
<typeName>CondAttrListCollection</typeName>
```

Documentation: twiki, LXR

Created:	2009-07-13_09:26
Last Modified:	2009-07-13_09:26
Node/CBF ID:	3 / 525
Channel Count:	4070
Payload Count:	8
IOV Base:	run-lumi
Attribute Type:	CondAttrListCollection
Versioning:	1 (Multi version folder)

Payload Columns:

Column	Type
SampleSize	Int32
BarrelEndcap	Int32
Layer	Int32
Eta	Int32
Phi	Int32
DefectType	String4k
Threshold	Float
DefectList	String4k

References:
COMP200_F0003_IOVS
COMP200_F0003_TAGS
COMP200_F0003_IOV2TAG
COMP200_F0003_CHANNELS

-/+ Show/Hide Folder Tags (12)

Figure 3. Conditions DB Folder Report

6. Conclusions

Metadata about the ATLAS Conditions Database structure has been aggregated into a dedicated system, providing unique information and services to experts and users. It has proven to be useful in the post-LHC Run 1 Conditions Database cleanup efforts in preparation for LHC Run 2. It is part of a broader integrated ATLAS Metadata program which shares information and infrastructure with other ATLAS Metadata systems to deliver coherent and integrated services to its community.

Every moderate to large scale experiment needs to efficiently store and access conditions-type data. When this grows in size and diversity as the ATLAS Conditions Database has, collecting metadata about its structure has proven to be useful in many respects.

References

[1] Valassi A, Duellmann D, Amorim A, Barros N, Franco T, Klose D, Pedro L, Schmidt SA, Tsulaia V (2004) “LCG Conditions Database Project Overview”, *CHEP 2004, Interlaken, Switzerland*. `https://indico.cern.ch/contributionDisplay.py?contribId=447&sessionId=6&confId=0`

[2] Trentadue R, Clemencic M, Dykstra D, Frank M, Front D, Kalkhof A, Loth A, Nowak M, Salnikov A, Valassi A, Wache M (2012) “LCG Persistency Framework (CORAL, COOL, POOL): Status and Outlook in 2012 ”, *J. Phys.: Conf. Ser.* **396** 052067. `http://iopscience.iop.org/1742-6596/396/5/052067`

[3] Oracle Database, `http://www.oracle.com`

[4] Barberis D, Bujor F, de Stefano J, Dewhurst AL, Dykstra D, Front D, Gallas E, Gamboa CF, Luehring F, Walker R (2012) “Evolution of grid-wide access to database resident information in ATLAS using Frontier”, *J. Phys.: Conf. Ser.* **396** 052025. `http://iopscience.iop.org/1742-6596/396/5/052025`

[5] Sharmazanashvili A, Batiashvili G, Gvaberidze G, Formica A (2013) “A tool for Conditions Tag Management in ATLAS” `https://indico.cern.ch/contributionDisplay.py?contribId=287&confId=214784`.

[6] Gallas EJ, Albrand S, Fulachier J, Lambert F, Pachal K E, Tseng J C L, Zhang Q (2012) “Conditions and configuration metadata for the ATLAS experiment”, *J. Phys.: Conf. Series* **396** 052033. `http://iopscience.iop.org/1742-6596/396/5/052033`

[7] Albrand S, Doherty T, Fulachier J and Lambert F (2008) “The ATLAS METADATA INTERFACE” *J. Phys.: Conf. Series* **119** 072003. `http://www.iop.org/EJ/article/1742-6596/119/7/072003` and Albrand S, Fulachier J, Lambert F, Aidel O (2013) “Looking back on 10 years of the ATLAS Metadata Interface” this conference. `https://indico.cern.ch/contributionDisplay.py?contribId=260&confId=214784`