

PAPER • OPEN ACCESS

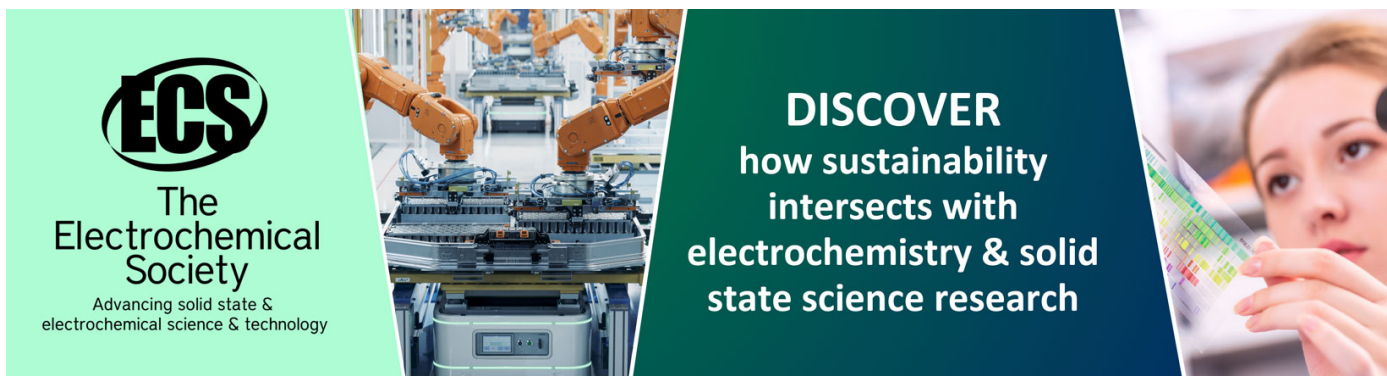
## The Database Driven ATLAS Trigger Configuration System

To cite this article: Carlos Chavez *et al* 2015 *J. Phys.: Conf. Ser.* **664** 082030

View the [article online](#) for updates and enhancements.

You may also like

- [Performance of the ATLAS Trigger System](#)  
C Gabaldon
- [The trigger and timing system of the Double Chooz experiment](#)  
F Beissel, A Cabrera, A Cucoanes et al.
- [Operation of the ATLAS trigger system in Run 2](#)  
The ATLAS collaboration



**ECS**  
The  
Electrochemical  
Society  
Advancing solid state &  
electrochemical science & technology

**DISCOVER**  
how sustainability  
intersects with  
electrochemistry & solid  
state science research

# The Database Driven ATLAS Trigger Configuration System

Carlos Chavez<sup>1</sup>, Michele Gianelli<sup>2</sup>, Alex Martyniuk<sup>3</sup>, Joerg Stelzer<sup>4</sup>,  
Mark Stockton<sup>5</sup>, Will Vazquez<sup>2</sup>

<sup>1</sup> University of Sussex, Brighton, United Kingdom

<sup>2</sup> Royal Holloway University of London, London, United Kingdom

<sup>3</sup> University College London, London, United Kingdom

<sup>4</sup> CERN, CH-1211 Geneva 23, Switzerland

<sup>5</sup> McGill University, Montreal, Canada

E-mail: [martyniu@cern.ch](mailto:martyniu@cern.ch)

**Abstract.** The ATLAS trigger configuration system uses a centrally provided relational database to store the configurations for all levels of the ATLAS trigger system. The configuration used at any point during data taking is maintained in this database. A interface to this database is provided by the TriggerTool, a Java-based graphical user interface. The TriggerTool has been designed to work as both a convenient browser and editor of configurations in the database for both general users and experts. The updates to the trigger system necessitated by the upgrades and changes in both hardware and software during the first long shut down of the LHC will be explored.

## 1. Introduction

The two level trigger system of ATLAS(A Toroidal LHC Apparatus [1]) at the Large Hadron Collider (LHC) at CERN is an essential component of the detector. By only recording events containing interesting collisions, it reduces the ATLAS event rate from the bunch crossing rate of 40 MHz to a lower, more manageable output rate of 1kHz for storage. The trigger [2] itself is a complex highly configurable system comprised of the hardware based Level 1 (LVL1) systems which reduce the rate to  $\approx 100\text{kHz}$ , seeding a software based High Level Trigger (HLT) system. The trigger system has been upgraded for Run-2, to allow for higher output rates [3]. During data taking both the hardware and software components of the system must be quickly and reliably re-configured, to avoid data loss. The precise configuration used at any point during data taking must be recorded accurately to be able to reproduce the trigger behaviour in the recorded data. The ATLAS trigger configuration system uses a relational database, the TriggerDB, to store this configuration data in a stable and accessible manner.

The TriggerTool graphical user interface (GUI) is the sole editor of the TriggerDB. It provides the means to browse the configurations in the TriggerDB, the ability to create new configurations and edit existing ones in a reproducible and self-consistent fashion.

## 2. The ATLAS Trigger System

The proton bunches in Run-2 of the LHC will cross at a nominal rate of 40MHz due to the 25ns bunch spacing. The two level trigger needs to rapidly and efficiently identify and accept



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

interesting physics events, within the constraints of the data-flow and storage systems. A brief description of the trigger system, necessary to set the scope in explaining the trigger configuration, is provided in following section.

### 2.1. The Level 1 Trigger

The LVL1 trigger is comprised of custom electronics close to the ATLAS detector. It makes decisions based on information from the calorimeter and muon systems to reduce the event rate to less than 100kHz.

The ATLAS muon spectrometer (MS) contains trigger chambers: resistive plate chambers in the barrel and thin gap chambers in the end-caps. The LVL1 trigger uses trigger primitive objects (TOBs) from these chambers to search for high transverse momentum ( $p_T$ ) muons. The decision is based on the multiplicity of muons passing six programmable  $p_T$  thresholds. The ATLAS calorimeter systems passes TOBs representing the calorimeter energy deposits to the LVL1 trigger. The calorimeter trigger aims to identify high transverse energy ( $E_T$ ) electrons, photons, jets and hadronically decaying tau leptons from this raw information. It also makes decisions based on events with large missing or total energy. Multiplicity information is produced for a number of programmable  $E_T$  thresholds for each object type.

For Run-2 an additional LVL1 topological trigger system (L1Topo) has been installed to allow for selection on physically meaningful quantities obtained by combination of the TOBs. The L1Topo system receives the same TOBs as the LVL1, from which it can make decisions based on the topological arrangement of the objects. The system can sort input trigger objects into a truncated list (maximum 6 objects) based on their  $E_T$ . Using the sorted or raw lists of objects the L1Topo can then make decisions based on the position or relative position of objects:  $\eta$ ,  $\phi$ ,  $\Delta\eta$ ,  $\Delta\phi$ ,  $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ . Or can make decisions based on the mass ( $M$ ) or transverse-mass ( $M_T$ ) of objects. The L1Topo can also make a decision based on the scalar sum of energy deposits in the detector ( $H_T$ ) or a Kalman-filtered version of missing energy.

Signals from L1Calo, L1Muon and L1Topo are collected in the Central Trigger Processor (CTP) which combines the partial detector information in order to arrive at global reject/accept decision. The CTP's decision requires logical combinations of the trigger objects matching the LVL1 trigger items stored in the LVL1 trigger menu. The trigger menu can be programmed with up to 512 distinct items. These items can have their raw rate reduced by applying a prescale factor to them.

### 2.2. The High Level Trigger

The High Level Trigger (HLT) is implemented in software and runs on a farm of around 20,000 cores. The HLT trigger menu defines a list of physics signatures which are the result of a chain of algorithm sequences running on data originating from an accepted LVL1 item. Each sequence or step in the chain is executed sequentially. Each sequence is comprised of a logical combination of one or more trigger elements. The elements are produced by running algorithms on the elements of the proceeding step. The initial step refines decisions within "Regions of Interest" (RoIs), detector regions defined in terms of  $\eta$  and  $\phi$  flagged by the decision at LVL1 as interesting. At any point, if an algorithm rejects the RoI then the chain stops, no further processing is done. If all RoIs are rejected then the event is rejected. Past the configurable event building (EB) step the whole event is assembled in a HLT node, all further algorithms now have access to the full information of the collision event. The algorithmic processing of trigger elements is performed by the HLT steering software, within the ATLAS ATHENA software environment. Similarly to LVL1 the prescales can be applied to these chains to reduce their rate. The HLT reduces the event rate to around 1kHz and has a time budget of a few seconds in average to process the event, more for events which are accepted and less for events to be rejected.

### 3. Configuring the ATLAS Trigger

The configuration of the ATLAS trigger system needs to coherently describe the settings of both the hardware and software sides of the trigger system described in the previous section. To define the state of the LVL1 and L1Topo systems the configuration contains:

- The definitions of the muon/calorimeter threshold for each TOB.
- LVL1 Trigger Menu: A list of LVL1 trigger items against which logical combinations of TOBs are compared by the CTP.
- L1Topo Trigger Menu: A list of definitions for the sorting and decision algorithms used in the L1Topo system.
- The prescales applied to the items.

In addition to the items, there are several global settings such as LVL1 random trigger rates. Moreover, a representation of the human-readable menu definition for use by the CTP hardware is required. This is produced by the Trigger Menu Compiler and also forms part of the configuration data.

For the configuration of the HLT software the following information is needed:

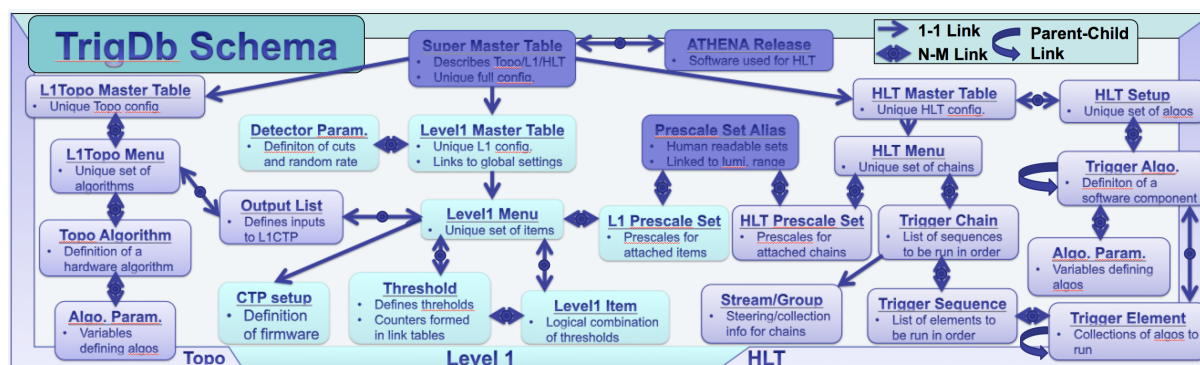
- HLT Trigger Menu: A list of HLT trigger chains, a list of partial decisions required in order for the chain to accept the event.
- The algorithms, other software components and their settings/configurable parameters.
- The prescale associated with each chain.

As an algorithm's performance can change between versions of the software, the database also stores the release used to run the configuration, to allow complete reproduction of the trigger behaviour.

#### 3.1. The TriggerDB

The configuration data is stored in a relational ORACLE SQL database known as the TriggerDB. An overview of the schema for the TriggerDB is shown in Fig. 1. All tables have an integer primary key (PK), which corresponds to a unique combination of a name and version for the object and any dependent objects being stored. The top level table follows this dependency structure, referring to the configurations of the LVL1 and HLT systems. The PK of this table, known as the super master key (SMK) uniquely describes the bulk of the configuration. Frequently changing configuration parameters, such as the sets of prescales for the CTP and HLT are identified by two additional prescale set keys. This triplet of TriggerDB keys uniquely defines the trigger setting used to take the data. Prescale sets to be used for a specific set of conditions, e.g. during physics runs or cosmic running, can be attached to a prescale set alias. This gives the collection a human readable description, as well as describing the instantaneous luminosity range in which the keys should be used.

The supermaster table defines the three menu configurations through foreign-key relationships to the respective MASTER tables. The relational structure of the tables below the master tables store the objects of the menus and link them in a hierarchical (parent-child) order through many-to-many link tables. For example, on the HLT side, the TRIGGER MENU table is the parent of the TRIGGER CHAIN table. Below this the tables for the trigger sequences, trigger elements, algorithms and finally the configurable parameters of the algorithms are linked. ATHENA software modules are associated by parent-child relations (e.g. an algorithm with a private tool); the component-to-component link table allows this connection to be stored. As well as algorithms and tools used by trigger chains, certain infrastructure components are also required to run the trigger software. These are linked directly from the HLT master table, via the HLT SETUP. The performance of the HLT software is also dependent on the particular ATHENA



**Figure 1.** An overview of the TiggerDB schema. Each box represent data base table. Rations between the tables are expressed by bi-directional arrows in case of N-to-M and one directional arrows for 1-to-1. Some minor tables have been removed or combined for simplicity. For more details see the text.

release installed. The SMK is therefore linked to a RELEASE table where this information is recorded. For the L1Topo side, below the TRIGGER MENU a description of the required algorithms and their parameters are stored. A list of outputs from the L1Topo algorithms used as input to the LVL1 system are linked directly between the menus. For the LVL1, below the TRIGGER MENU table are the tables for the trigger items, the thresholds and the threshold definitions. Further link tables from the threshold table define the CTP cabling information and the configuration of the CTP monitoring. The output of the Trigger Menu Compiler is stored in the CTP SMK and CTP FILES tables.

Two primary concepts drive the design of the TriggerDB schema. Firstly, there should be no duplication of records in the database. Secondly, the TriggerDB maintains every configuration ever used to take the data. The use of many-to-many link tables allows the database to assemble configuration primitives into many functional configurations without the need to replicate the primitives. Records within the database are never deleted, thus maintaining the history of all configurations. The TriggerTool graphical user interface (GUI) is the sole means of manipulating the database, and has been designed with these concepts at its heart.

### 3.2. Interactions with the TriggerDB

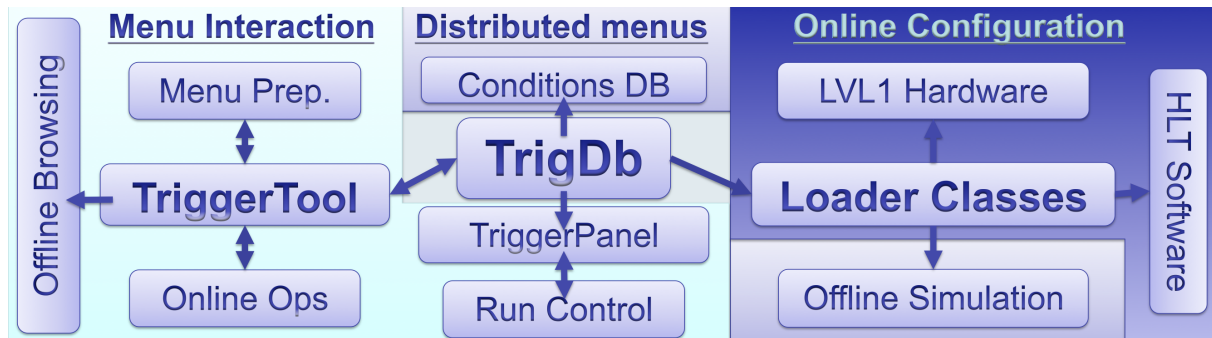
The TriggerDB lies at the centre of the ATLAS trigger configuration system. A significantly simplified description of the full system is shown in Fig. 2. The read-write online instance of the TriggerDB is hosted on a Oracle CERN main server. The information stored is replicated to a read-only account for browsing offline.

At the beginning of the data taking, custom configuration loader classes are used to read the TriggerDB and are responsible for configuring the LVL1/L1Topo and HLT components. The offline trigger simulation use the same classes to access the read-only account or separate instances of the TriggerDB. In this way MC simulation or offline reprocessing can access the same configurations used online.

## 4. The TriggerTool Graphical User Interface

### 4.1. Overview

The TriggerTool GUI to the TriggerDB is a Java project developed in the NetBeans IDE. It is split into two main packages, the TriggerTool package which holds the GUI components of the tool and the TrigDb package which holds only the classes describing the database tables



**Figure 2.** An overview of the interactions with the TriggerDB within the ATLAS trigger system. The TriggerTool is used to edit/browse its contents. The TriggerPanel is used to switch database keys in the ACR, the keys selected are stored in the conditions database. The loader classes are used to port the configurations from the database to the trigger systems.

(ORM) and the SQL queries used to access the TriggerDB. This functionality split allows other tools to reuse the queries used in the TriggerTool to access the TriggerDB without running the TriggerTool itself. In total both packages are formed of around 220 classes.

#### 4.2. Connection to TriggerDBs

The latest versions of the TriggerTool are installed both on the network in the ATLAS Control Room (ACR) and the CERN network (lxplus). Once launched the user is presented with a TriggerTool login window offering a variety of database connection options dependent on the user roles and privileges. The ACR version has read-write access to the main TriggerDB instance, whereas the lxplus version can access the read-only replica account of the TriggerDB. Different access privileges are assigned to users:

- **User:** In this mode the user can only access the read-only replica of the main DB. All editing features are disabled in this mode.
- **Shifter:** Shifters in the ACR use this mode. It has access to the main read-write instance of the TriggerDB, but editing options are limited to those necessary to perform typical actions required from the shifter.
- **Expert:** This mode offers full access to the TriggerTools editing features. It is only accessible to a limited number of trigger experts.
- **DBA:** The Database Administrator mode offers the same rights as Expert, but also allows editing of roles offered to other users.

The user information in the main online database is stored in a separate table. When connecting to a private database users are given Expert rights.

#### 4.3. The Main Panel

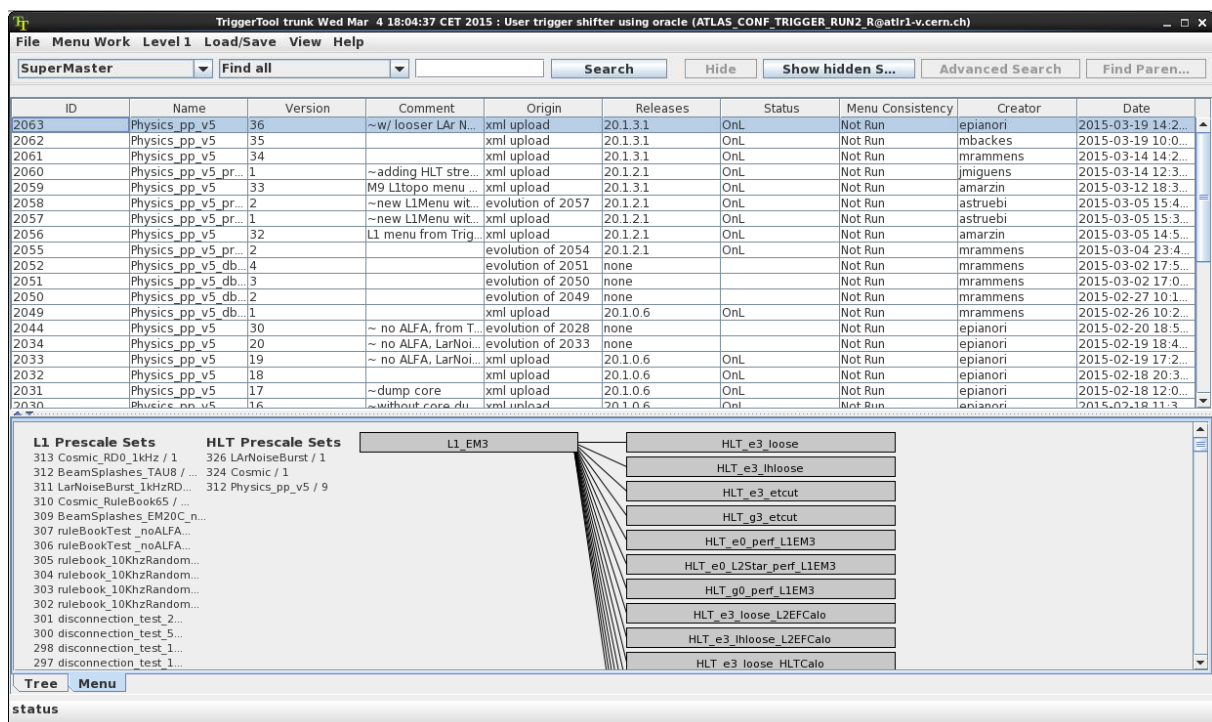
When the user has gained access to their desired database they are shown the main panel, an example of which is presented in Fig.3. The table in the top section of the panel shows the configurations currently contained in the connected TriggerDB. A minimal set of information about those configurations is also shown, including: the configuration name and version, a comment, where the key originated from, who made it, when and with what ATHENA release. This view can be filtered by searching for any property of the configurations represented.

Users can upload new configurations from an XML file. A coherent set of files has to be specified in order to obtain a functional configuration. When a new configuration is added to



the TriggerDB only the components that do not exist yet in the database are uploaded. Any that are identical to already existing ones are reused by linking to the new configuration. As a consequence repeated upload of identical configuration results in no actions on the database and the key of the existing configuration is reported to the user. If the configuration is new to the database, then a new SMK is produced.

After selection of the configuration more detailed information is loaded in the bottom panel. By default a graphical representation of the LVL1 items and their connections to the HLT chains is shown alongside the available prescale keys. A more detailed view is available in the second tab where an entire tree of the hierarchical TriggerDB structure can be inspected.



**Figure 3.** The MainPanel of the TriggerTool in Shifter mode. The top half shows the available super master keys in the connected database. More detailed menu content and prescale set information is shown in the bottom half for the selected configuration.

#### 4.4. The Overview Panel

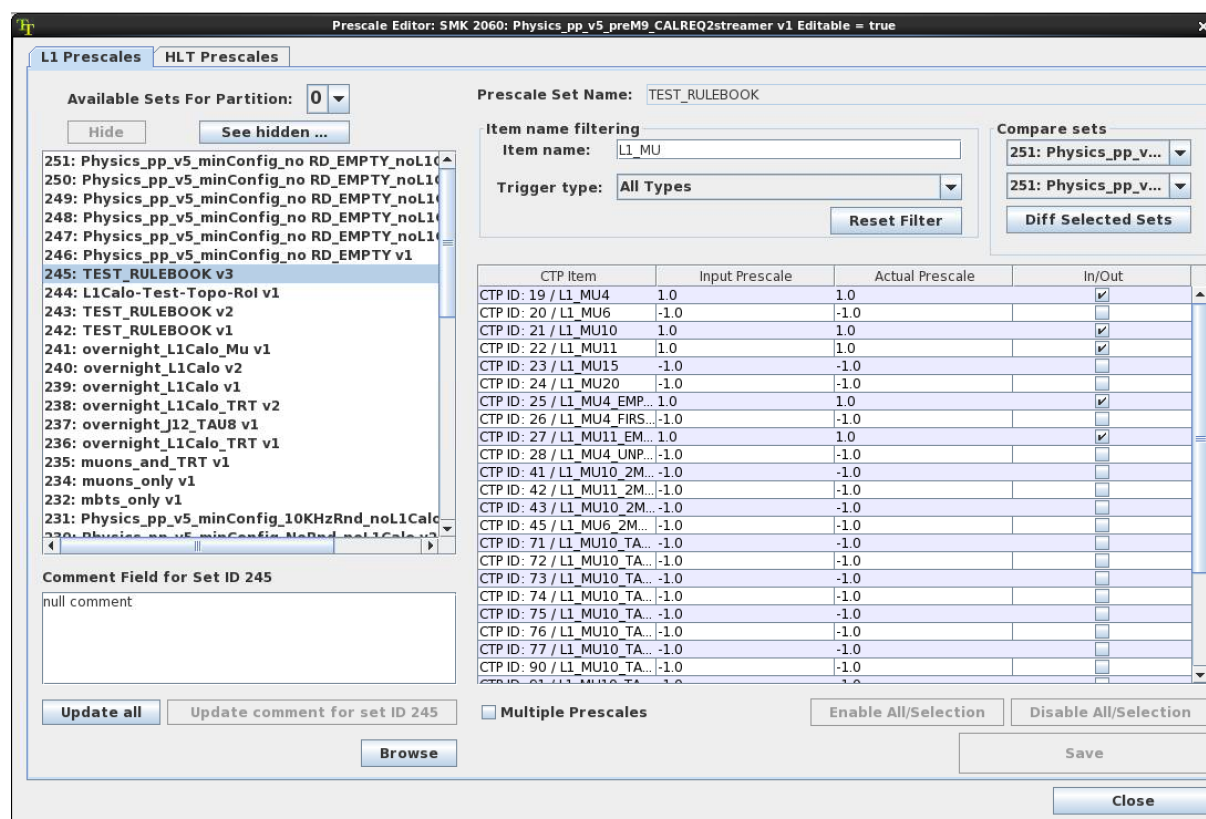
By double-clicking on a SMK in the main panel an overview panel is shown. This loads the full configuration into memory and allows the user to browse or edit all configuration parameters. The information is displayed as a set of connected graphical objects showing the logical triggering flow from all the L1Topo/LVL1 items through to the HLT chains that are seeded by them. Initially only the name of each item or chain is shown, but they can be expanded to give full editable information on each component of the respective menus.

Most frequently, the following information is edited using the overview panel. Changes can be made here to the random trigger rates, deadtime and bunch group settings of the LVL1 system. The full HLT setup is also accessible, allowing for editing of the individual parameter level of ATHENA algorithms stored in the configuration. Using the panel, experts can alter chains, add them, or remove them completely from the menu. Any changes made to the menu

are stored in memory until the user demands the recording of the altered configuration back to the TriggerDB. All changes made are then saved to the connected instance of the TriggerDB and a new SMK returned.

#### 4.5. The Prescales Panel

Due to frequent updates of the prescale sets they are treated separately from rest of the configuration. Separate editing panels in the TriggerTool are provided to work with them. All prescale sets for the HLT and LVL1 which are usable with a given SMK are shown in the prescales panel, shown in Fig. 4. This panel contains two tabs, one for the LVL1 and one for the HLT. The user is shown in each level in the top left, a list of available prescale sets associated to this SMK. Selecting one of these keys loads the details of that key into the table on the right of the panel. This table can be conveniently filtered using information contained in it (e.g. in the LVL1 prescales table by the LVL1 item name). The table can be used to edit the prescale rate or enable or disable prescales, a task simplified by the check-boxes labeled in/out in each row. When changes are complete the table can be saved as a new prescale set and associated with the SMK. The panel also offers the ability to show the differences between two prescale sets. The prescales difference panel only displays the items which differ between the two sets.



**Figure 4.** The prescales editor of the TriggerTool in Shift mode. Displayed is the edit functionality for the LVL1 prescales, the HLT prescales are accessible in the second tab. The top left panel shows the available LVL1 prescale keys, along with a brief description. The prescale values for the currently chosen set are displayed on the right. In this example, the filtering is applied to show an overview of all muon trigger items.

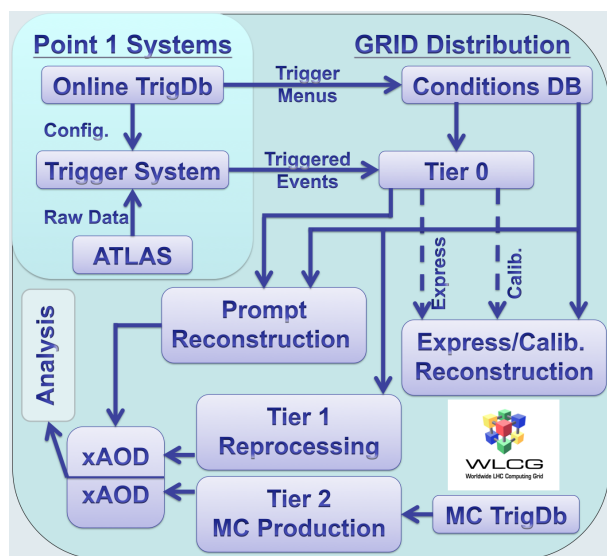


#### 4.6. Additional Features

The TriggerTool also offers a series of additional less widely used features. For instance the TriggerTool allows the user to obtain a full account of differences between two configurations or invoke the Trigger Menu Compiler in order to produce the firmware needed by LVL1 hardware.

### 5. TriggerDB Distribution

All configurations used to define the ATLAS trigger system are automatically replicated to the offline read-only TriggerDB instance, as in Fig. 5. Further replications to GRID sites are performed using Frontier technology. Additionally, the subset of the configuration critical to the analyses is saved in a conditions database (COOL) for each luminosity block. From the conditions database these records are copied to the derived data formats like ESD or AOD. A separate TriggerDB exists for MC production, storing trigger configurations broadly representing long periods of data taking.



**Figure 5.** Schematic view of distribution of trigger configuration information through various databases and to the clients starting from online TriggerDB instance (top left) down to the analysis (bottom left).

### 6. Summary

The ATLAS trigger system reduces the data-rate from the LHC by selecting interesting events to be recorded. The relational trigger database, TriggerDB, lies at the core of the configuration system used to describe the state of the hardware and software components of the system. It has been designed to store configurations with minimal duplication and no deletion of information.

The TriggerTool offers the sole access to the TriggerDBs. Its graphical interface allows users to browse, edit and create whole trigger configurations. The design of the TriggerTool and DB have been altered for Run-2 of the LHC to accommodate the new features of the ATLAS trigger such as the level 1 topological trigger. The TriggerDB schema has been improved to allow quicker uploads, alongside changes incorporating workarounds from Run-1 as more structured features.

### 7. References

#### References

- [1] ATLAS Collaboration, “ATLAS Experiment at the CERN Large Hadron Collider”, JINST **3** S08003, 2008
- [2] ATLAS Collaboration, “Expected Performance of the ATLAS Experiment: Detector, Trigger and Physics”, CERN-OPEN-2008-020, arXiv:hep-ex/0901:0512, 2009
- [3] Y. Nakahama (2015) “The ATLAS Trigger System: Ready for Run-2”, Proceedings of CHEP2015 J. Phys.: Conf. Ser.