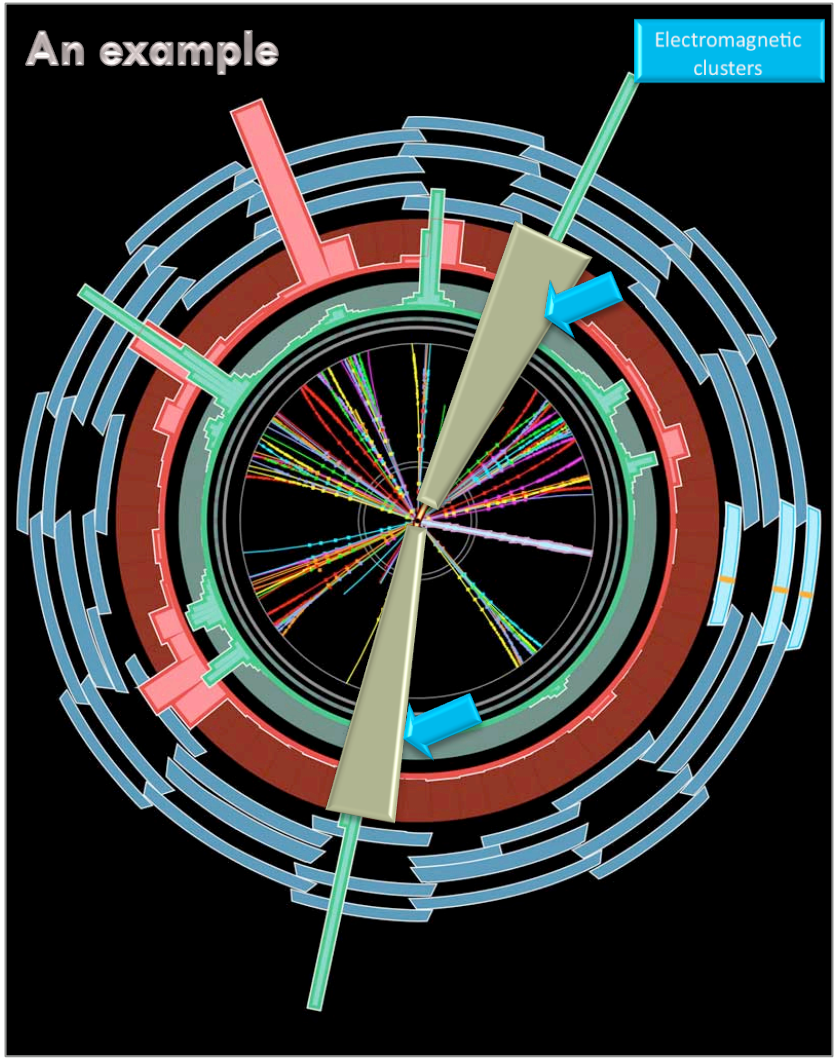
TRIGGER DATA FOR PHYSICS ANALYSIS

Ricardo Gonçalo – Royal Holloway

ATLAS Software Tutorial – 22nd to 24th April 2009

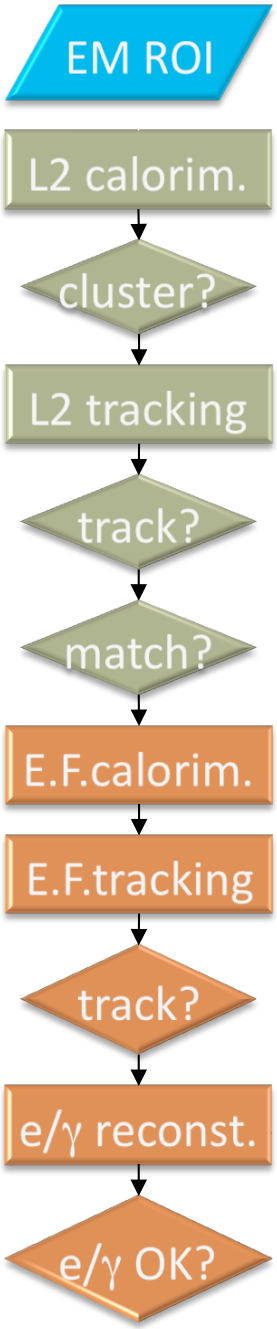
The Trigger



Level1:
Region of Interest is found and position in EM calorimeter is passed to Level 2

Level 2 seeded by Level 1
- Fast reconstruction algorithms
- Reconstruction within RoI

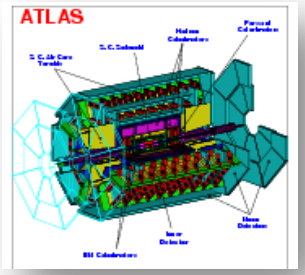
Ev.Filter seeded by Level 2
- Offline reconstruction algorithms
- Refined alignment and calibration



Trigger Configuration + Data

Preparation

Data taking



Configures

TriggerDB
All configuration data

Stores decoded Trigger Menu

Online Conditions Database COOL

Encoded trigger result from all 3 levels

Decoded trigger menu

Reconstruction

Trigger aware analysis

Trigger Result

- passed?, passed through?, prescaled?, last successful step in trigger execution?

Trigger EDM (Features)

- Trigger objects for trigger selection & studies

Trigger Configuration

- Trigger names, prescales, pass throughs

access through ITrigDecisionTool

ESD

AOD

DPD

TAG

With decreasing amount of detail

What's there? – Configuration Data

- L1 Items: name, version, CTP-Id, prescale
  - To come: bunch-group for each item
  - Also to come: Definition of L1Items in terms of thresholds
- HLT Chains: name, version, level, counter, prescale, trigger elements
- Streams: chains feeding into each stream
- Chain-groups: chains belonging to each group
- Bunch-groups: name of each of the 8 BG

FEX — **FE**ature e**X**traction algos produce

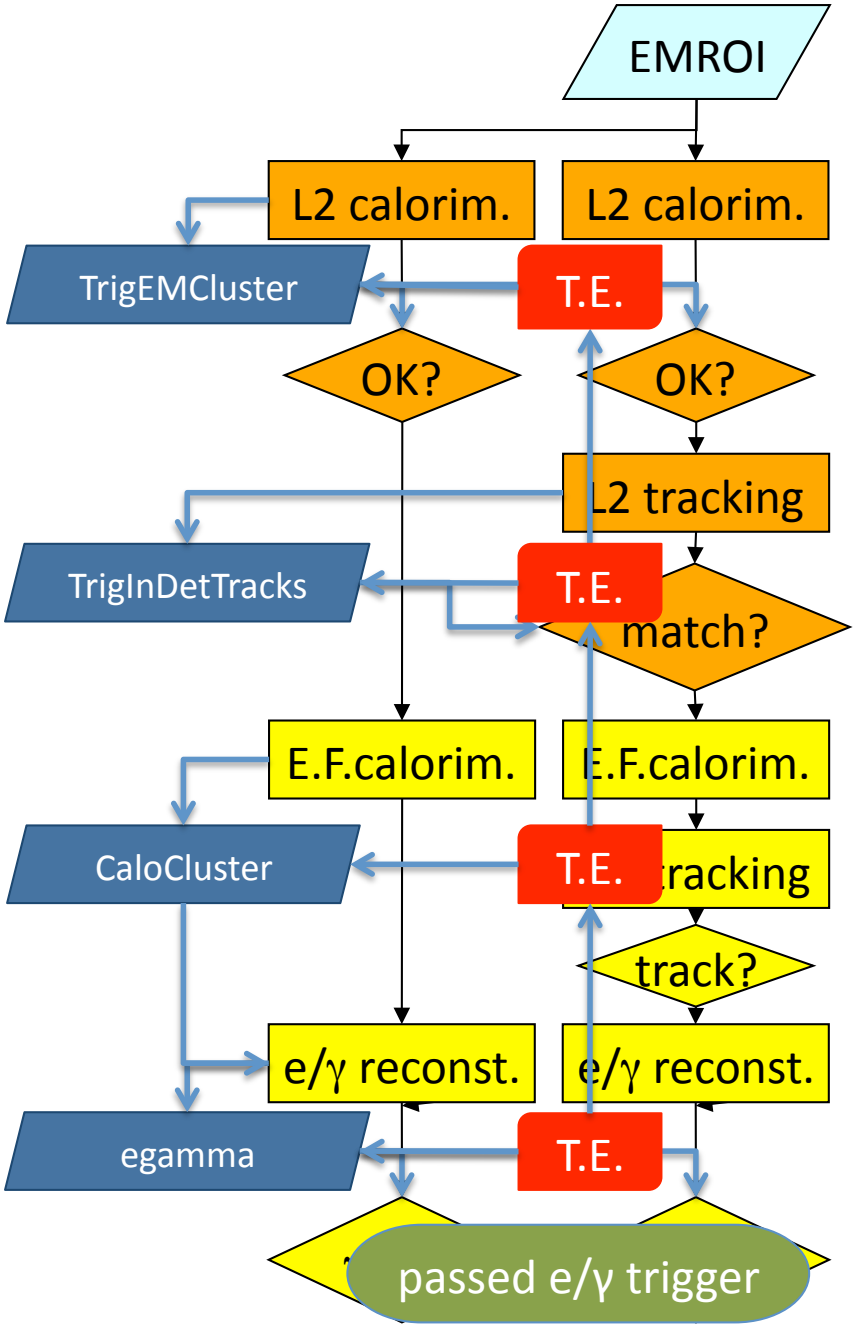
features — Features on which selection in

HYPO — **HYPO**thesis algos is based

- Chain:
  - Started, if seed has fired and chain is not **PRESCALED**
  - Stopped **AT STEP**, if a HYPO is not passed
  - Last HYPO passed → **CHAIN PASSED**

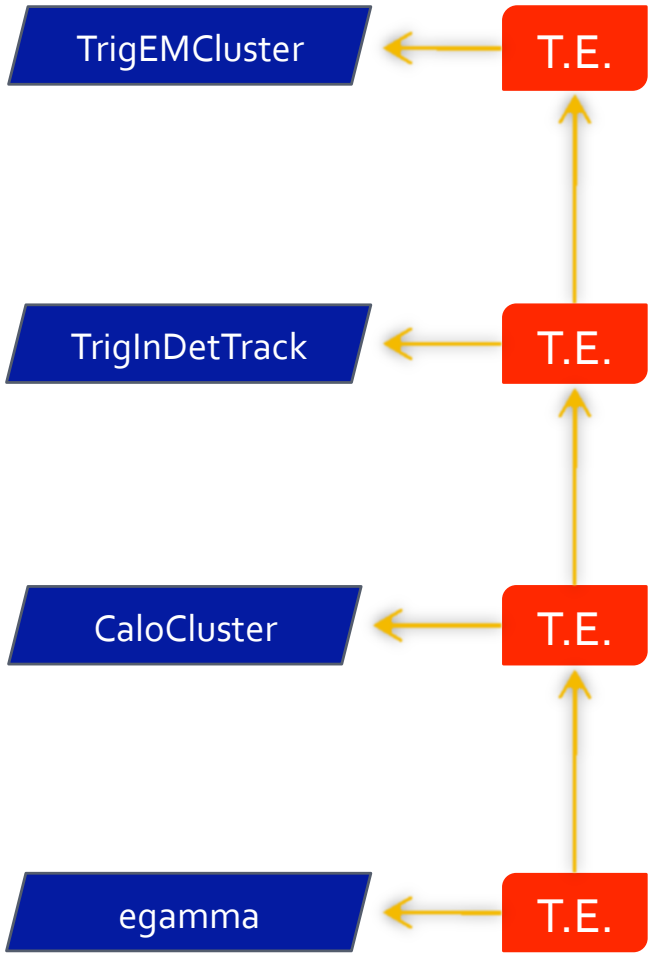
- Event:
  - Passed, if at least one EF chain is passed
  - Put into all streams that are associated with any passed EF chain

- Trigger information in ESD AOD DPD TAG
  - **TRIGGER DECISION +** decision
  - **TRIGGER FEATURES +** features
  - **TRIGGER NAVIGATION +** T.E.
  - **CONFIGURATION**



What's there? – Features

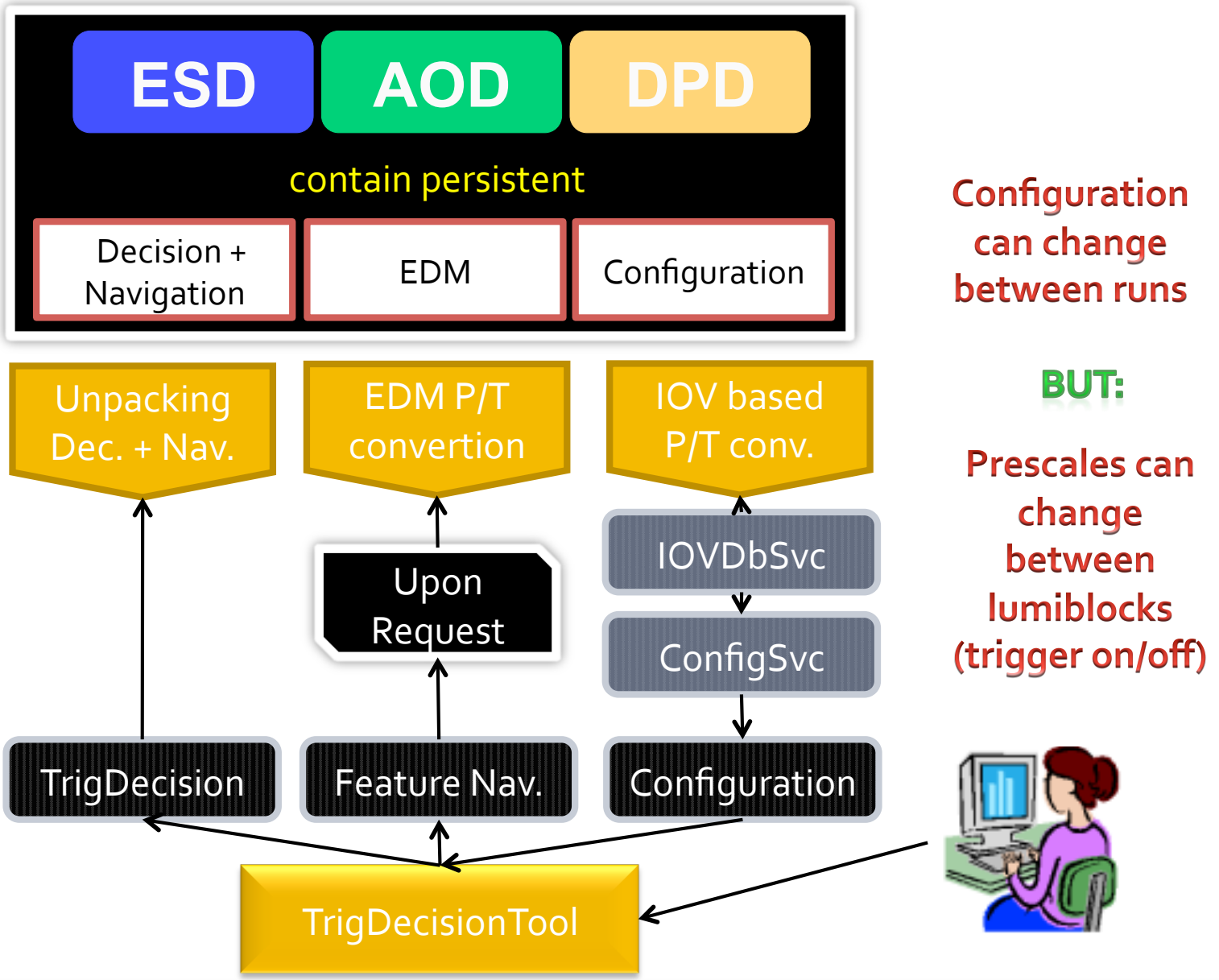
- Features can be retrieved through the "TriggerNavigation" using the TrigDecisionTool
- Features are created by FEX algorithms. They appear in StoreGate in containers according to FEX name. A FEX also creates a "TriggerElement" (TE)
  - A TE is used as handle to the feature
  - A TE has a pass/fail state set by the HYPO corresponding to the FEX
- So the navigation can give you the TE's for all the FEX that run in a chain
  - Or just those that passed the last step in the chain
  - From there you get the features (type templated)
  - This is the correct way to retrieve the features for each RoI



Trigger-aware analysis

- Analysis based on single trigger chain or an 'OR' of a few chains
- Chain definition – algorithms, cuts, multiplicities – do not change during a run, but can change between runs
  - Important for analysis on DPD, where multiple runs are merged
- Prescales at LVL1 or at HLT can change between luminosity blocks
  - A negative prescale means that this trigger is off. This is important for calculating the integrated luminosity

Bringing it Together – TrigDecisionTool



Configuration can change between runs

BUT:

Prescales can change between lumiblocks (trigger on/off)

Migration to new TrigDecisionTool in rel. 15.2.0!!!

- The standard user interface to trigger data for physics analysis
- Major change from release 15.2.0 to:
  - Provide new functionality
  - Simplify the API
  - Increase robustness
  - Allow easy use in both Athena and ARA
- See last tutorial for old version (<15.2.0)
  - http://indico.cern.ch/conferenceDisplay.py?confId=55063
- The two versions coexist in Athena for now (old one won't be supported for long)
  - In different namespaces: old TrigDec::/ new Trig:: and TrigDefs::
- Documentation:
  - Wiki: https://twiki.cern.ch/twiki/bin/view/Atlas/TrigDecisionTool15
  - Examples: `Trigger/TrigAnalysis/TrigAnalysisExamples`

Usage: Just like any other Tool

1. ToolHandle to a TrigDecisionTool in your algorithm header
2. Initialize ToolHandle in constructor as public tool
3. Retrieve tool in initialization
4. Use tool in execute()

```
private:
    ToolHandle<Trig::TrigDecisionTool> m_trigDec;
```

```
MyAlgo::MyAlgo(const std::string &name, …) :
m_trigDec("Trig::TrigDecisionTool/TrigDecisionTool")
{
 declareProperty("TrigDecisionTool", m_trigDec, \
 "The tool to access TrigDecision");
}
```

```
StatusCode sc = m_trigDec.retrieve();
```

```
(*m_log)   << MSG::INFO
           << "L2_mu.* passed/failed: "
           << m_trigDec->isPassed("L2_mu.*")
           << endreq;
```

TrigDecisionTool and ChainGroups

- Important concept in the new tool: ChainGroup
  - They work as an OR of triggers
  - Created from a vector of trigger names, comma-separated list or regular expressions:
  - `const ChainGroup* getChainGroup(const string& names)`

```
m_all   = m_trigDec->getChainGroup(".*");
m_allL1 = m_trigDec->getChainGroup("L1_.*");
m_allL2 = m_trigDec->getChainGroup("L2_.*");
m_allEF = m_trigDec->getChainGroup("EF_.*");
```

- Can be used e.g. to get pass/fail information:

```
(*m_log) << MSG::INFO << "L1: " <<  m_allL1->isPassed() << endreq;
(*m_log) << MSG::INFO << "L2: " <<  m_allL2->isPassed() << endreq;
(*m_log) << MSG::INFO << "EF: " <<  m_allEF->isPassed() << endreq;
```

- Can be created on the fly:

```
// this creates on the fly soem chain groups and check if they passed
(*m_log) << MSG::INFO << "L2_mu.*" << m_trigDec->isPassed("L2_mu.*") << endreq;
(*m_log) << MSG::INFO << "L2_j.*" << m_trigDec->isPassed("L2_j.*") << endreq;
```

Accessing features created by a chain

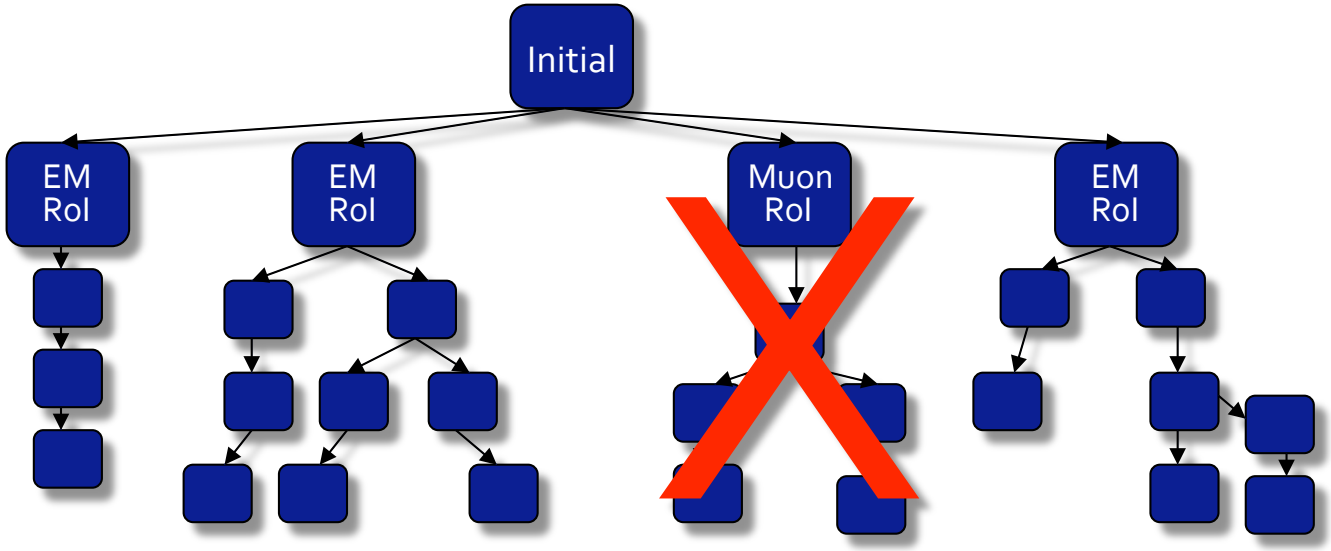
- Use chains or chain groups to access trigger features produced by each trigger/group
  - FeatureContainer gives access to Trigger Elements and trigger objects for simple or complex triggers

```
FeatureContainer f = m_trigDec->features("L2_e15");
FeatureContainer::combination_const_iterator it;

for (it = f.getCombinations().begin(); it !=
    f.getCombinations().end(); ++it) {
    std::vector< Feature<TrigRoiDescriptor> > initRois =
                          cIt->get<TrigRoiDescriptor>("initialRoI");
}
```

  - Combinations allows to retrieve e.g. combinations of objects satisfying combined triggers like EF_tau16_2j23

Trigger information slimming



- Needed for producing custom DPDs (e.g. Physics DPDs or user DPDs)
- **Feature Removal** – remove objects according to type and FEX
- **Pruning** – remove entire RoIs (see example above)
- **Squeezing** – compress navigation information (no information loss)
- **Slimming** – remove chains belonging to groups or streams
- More info in: https://twiki.cern.ch/twiki/bin/view/Atlas/HLTTrigNavigationSlimming

Tools to Investige the Trigger Setup

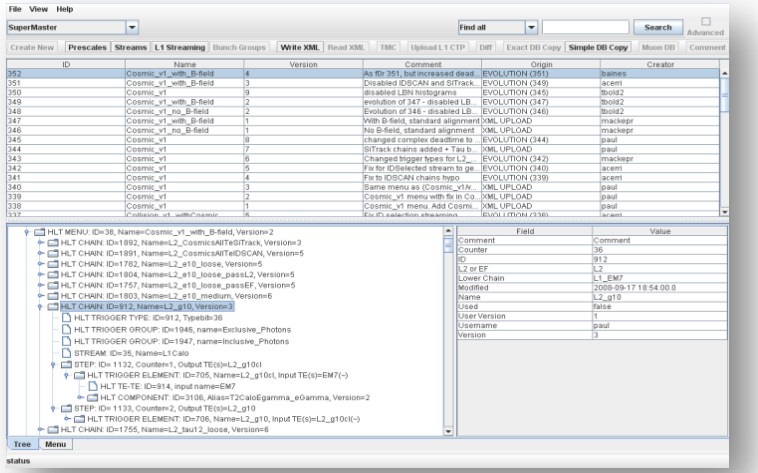
- Run-summary page: trigger chains, prescales, rates – web based
  - Query single run
- AtlCoolTrigger.py: same, but command line
  - Views many runs
- TriggerTool: Java based GUI to browse the TriggerDB (replica) for all information
- http://trigconf.cern.ch as web front-end to TriggerTool and AtlCoolTrigger.py

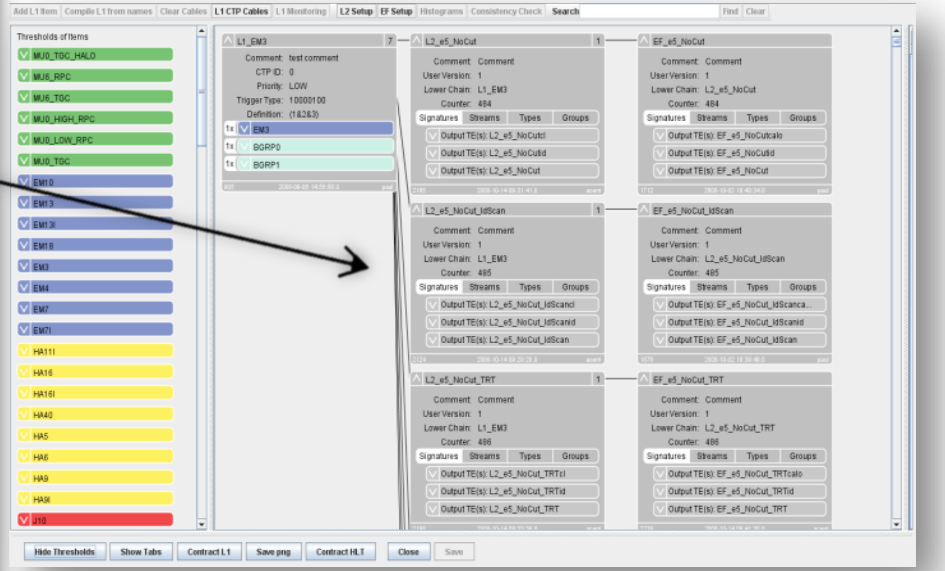
Trigger Menu Listing

- Triger Menu and L1 rates stored in COOL, HLT rates coming. Quick access via
  - Run summary pages (WEB based)
  - http://atlas-service-db-runlist.web.cern.ch/atlas-service-db-runlist/query.html
    - Trigger names, rates
  - AtlCoolTrigger.py (command line tool)
    - AtlCoolTrigger –r 91000-99000 (many run summary)
    - AtlCoolTrigger –v –m –r 90272 (single run menu)
    - Prints keys, trigger menus, streams, allows diff-ing of menus in different runs

TriggerTool

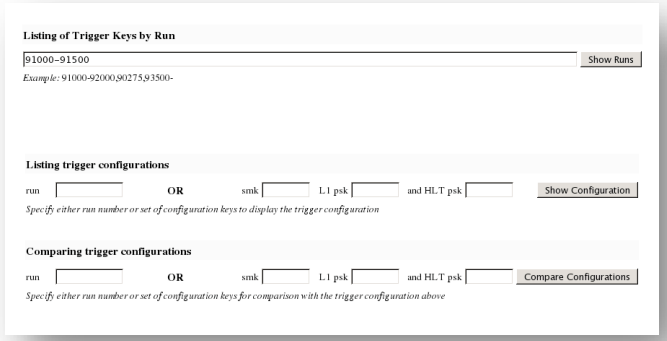
- Java based front end to TriggerDB, launch from the web (Java web-start): http://www.cern.ch/triggertool
  - Overview of all trigger configurations
  - Detailed and convenient investigation of trigger menus
    - Trigger definition L1->L2->EF: prescales, threshold algorithms, selection criteria, streaming information, etc.
- Possibility to compare different trigger configurations





Web Interface to COOL and the TriggerDB

- Web interface http://trigconf.cern.ch
  - Runs TriggerTool on the server, result presented as dynamic html pages

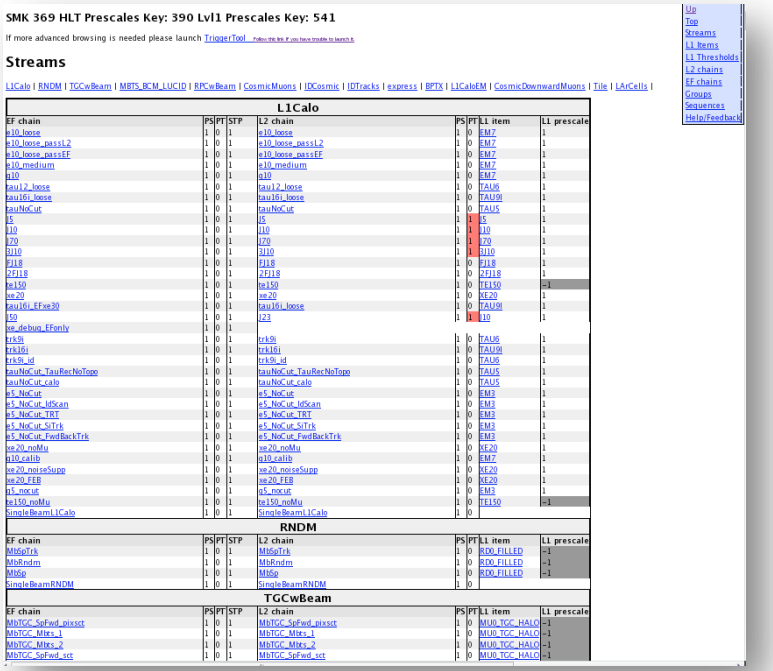


1. Search run-range

2. Run list

Also with simple comparison functionality

3. Trigger configuration (browsable) (definition, algorithms, selection cuts)



Scripts to Check Pool File Content

- checkTrigger.py AOD.pool.root
  - Runs over ESD/AOD/DPD and presents detailed (chain-wise) counts of the trigger decision
- checkTriggerConfig.py -d AOD.pool.root
  - Runs over ESD/AOD/DPD and presents detailed trigger configuration(s) in the file
  - Shows multiple configurations, in DPD possible/likely

Additional Information

General Trigger info: https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerUserPages

Tutorials: https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerSoftwareTutorialPage

Trigger Event Data Model (EDM) : https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerEDM

TrigDecisionTool – all releases/versions:
Twiki: https://twiki.cern.ch/twiki/bin/view/Atlas/TrigDecisionTool

Athena examples:
TrigAnalysisExamples package:
http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDevDirectory/AtlasOffline/latest_doxygen/InstallArea/doc//TrigAnalysisExamples/html/
TrigDecisionChecker (pre-15.X.0 for the moment):
http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/Trigger/TrigValidation/TrigValAlgs/TrigValAlgs/TrigDecisionChecker.h?revision=1.2&view=markup

ARA example:
ARA: https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerARA14

Trigger Configuration: https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerConfiguration?topic=TrigDecisionTool
Run summary pages: http://atlas-service-db-runlist.web.cern.ch/atlas-service-db-runlist/query.html
(include link to start up the TriggerTool)

Trigger information slimming: https://twiki.cern.ch/twiki/bin/view/Atlas/HLTTrigNavigationSlimming

TriggerMenu working group: https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerPhysicsMenu

Help! Hypernews forum hn-atlas-TriggerHelp@cern.ch