



# Trigger Tutorial



Catrin Bernius  
UCL

2nd Artemis workshop  
Thessaloniki  
27./28. September 2007

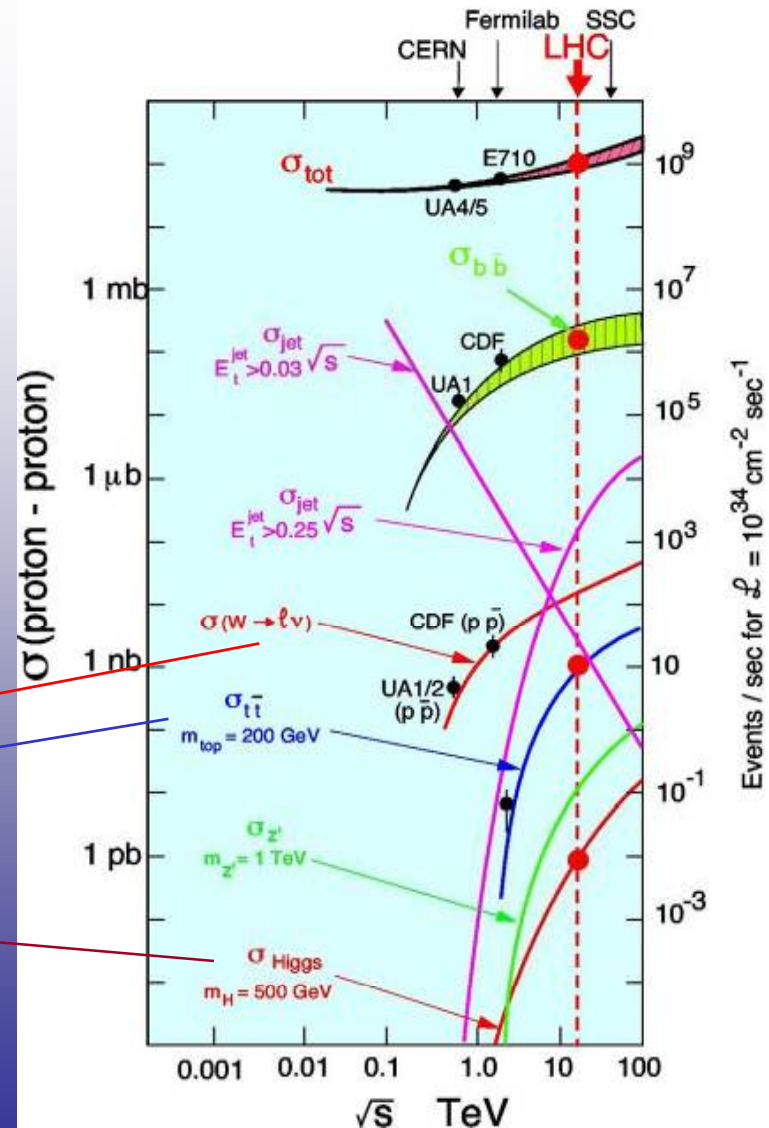
# Introduction & Outline

**Aim of this tutorial:** Retrieve TriggerDecision from AOD file with offline release 12.0.6

- **Part 1: Introduction to...**
  - Trigger Challenges
  - ATLAS Trigger System
    - Level 1, Level 2 and Event Filter
    - Trigger Menu
  
- **Part 2: How to use it...**

# Challenges faced by the ATLAS Trigger

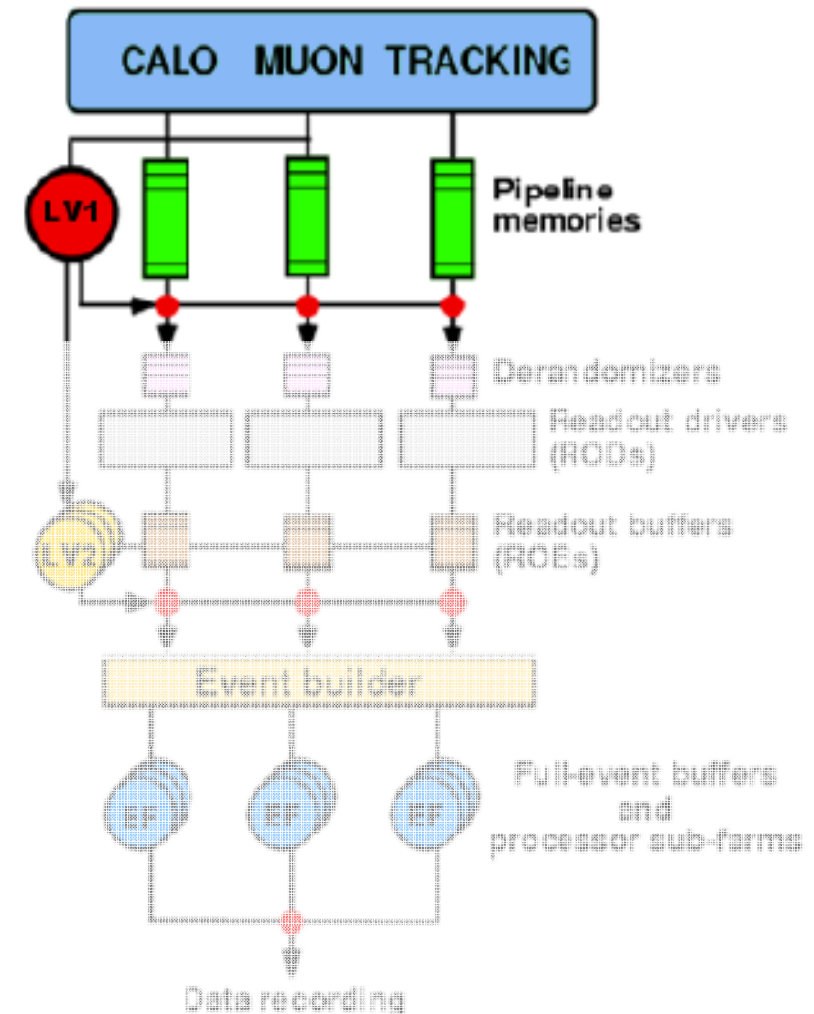
- Rare physics processes with cross sections  $\sim 10^6$  times smaller than total cross section
- 25ns bunch crossing interval (40 MHz) each with  $\sim 23$  interactions at high  $\mathcal{L}$ 
  - $\sim 10^9$  interactions per sec
- BUT: offline storing/processing:  $\sim 200$  Hz
  - $\sim 5$  events per million crossings
- In one second at design luminosity:
  - 40 M bunch crossings
    - $\sim 200$  events can be written out
  - $\sim 2000$  W events
  - $\sim 10$  top events
  - $\sim 0.1$  Higgs events
- Make sure to get the right 200 events!



# The ATLAS Trigger

- Level 1:
  - Hardware based
  - Coarse calorimeter and muon information
  - Latency 2.2  $\mu$ s
  - Output rate  $\sim$ 75 kHz
- Level 2:
  - Software based
  - Only detector sub-regions (RoI) processed, flagged by level 1 as interesting
  - Full detector granularity in RoIs used with fast tracking and calorimetry
  - Latency  $\sim$ 10 ms
  - Output rate  $\sim$ 1 kHz
- Event Filter (EF):
  - Runs after event building
  - Offline reconstruction algorithms using full calibration and alignment information
  - Latency  $\sim$ 1 s
  - Output rate  $\sim$ 200 Hz

High-Level Trigger





# Trigger Menu Table

Possible detailed Trigger menu table for low Luminosity scenario:

<b>Object</b>	<b>Physics coverage</b>	<b>Object name</b>
<b>electrons</b>	<b>Higgs, new gauge bosons, extra dim., SUSY, W/Z, top</b>	<b>e25i, 2e15i, e60</b>
<b>photons</b>	<b>Higgs, SUSY, extra dim.</b>	<b><math>\gamma</math>60, 2<math>\gamma</math>20i</b>
<b>muons</b>	<b>Higgs, new gauge bosons, extra dim., SUSY, W/Z, top</b>	<b><math>\mu</math>20i, 2<math>\mu</math>10</b>
<b>Jets</b>	<b>SUSY, compositeness, resonances</b>	<b>j400, 3j165, 4j110</b>
<b>Jets+missEt</b>	<b>SUSY, leptoquarks</b>	<b>j70+xE70</b>
<b>Tau+missEt</b>	<b>Extended Higgs models (e.g. MSSM), SUSY</b>	<b><math>\tau</math>35i+xE45</b>

Trigger Workshop: [Overview of current menus](#) (Marc-André Dufour)

# PART 2:

## How to use it....

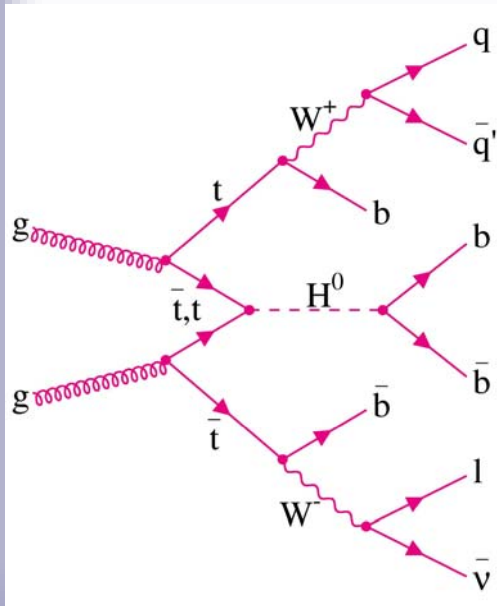
- Introduction to TriggerDecision
- Introduction to UCL ttH analysis code
- How to...
  - Retrieve TriggerDecision from AOD
  - Print out the Trigger menu
  - Print out statistics for one event
  - Write info to Ntuple

# TriggerDecision

- TriggerDecision is a user interface of Trigger code to finding the decision of the trigger on each event. It provides:
  - list of trigger signatures which were running at each trigger level and satisfied by the event
  - methods which return the overall result from each trigger level and for the whole trigger
    - `isDefinedL1,L2,EF()` : true if L1, L2, EF was running, false otherwise
    - `isDefined(std::string) const` : true if the signature (string) was running at any level
    - `isDefined(std::string, unsigned int) const` : true if signature (string) was running at trigger level int
    - `isTriggerPassed()` : checks at EF then L2 then L1 for a satisfied signature
    - `isPassedL1,L2,EF`
    - `isTriggered(std::string)` : true if signature satisfied
    - `isTriggered (std::string, unsigned int) const` : true if signature satisfied at level int
- Contains trigger menu items as specified by the configuration at reconstruction time (for 12.0.6, it is [CSC-06](#) )
- For each event, a TriggerDecision object containing the trigger information for each event is produced by an Algorithm derived class (TriggerDecisionMaker)



# The UCL ttH analysis code



- Tool based analysis representing the different reconstruction steps:

- **TriggerTool**
- bTaggingTool
- PreselectLeptons
- PreselectJets
- WInTool (semileptonic W reconstruction)
- WjjTool (hadronic W reconstruction)
- ttbarRecoTool
- higgsRecoTool

- In ttH analysis: trigger on isolated high  $p_T$  lepton

HG - 5	ttH, H $\rightarrow$ bb	mu20i; e25i; 3b jets from 4L1 jets; follow top Non-isolated muon @L1 and e/ $\mu$ +MET (to be studied)
--------	-------------------------	---

- Trigger Workshop: [Ricardo Goncalo's Talk](#)
- In this tutorial: used dataset is the signal sample
  - ttH (pos & neg leptons) with  $\sim 39000$  events

# Retrieve TriggerDecision from AOD

What we will do now...

- (a) Retrieve the TriggerDecision from an AOD
- (b) Print TriggerMenu
- (c) Print Statistics for each event
- (d) Write information to Ntuple and analyse it

- Coding is already done for you
- Output files will be provided
- Trigger Software Tutorial Page:

<https://twiki.cern.ch/twiki/bin/view/Atlas//TriggerSoftwareTutorialPage>

# Accessing TriggerDecision

- In cmt/requirements:

```
use TrigSteeringEvent      TrigSteeringEvent-00-*      Trigger/TrigEvent
```

- To make class available in TriggerTool.cxx:

```
#include "TrigSteeringEvent/TriggerDecision.h"
```

- Receive Trigger objects from StoreGate:

```
m_TriggerDecisionKey="MyTriggerDecision";

// retrieve latest TriggerDecision from Storegate
const TriggerDecision* trigDec = NULL;
sc = m_storeGate->retrieve(trigDec, m_TriggerDecisionKey);
if ( sc.isFailure() ) {
    mLog << MSG::WARNING << "Failed to retrieve
        TriggerDecision" << endreq;
    return StatusCode::SUCCESS;
} else {
    mLog << MSG::INFO << "TriggerDecision successfully retrieved"
    << endreq;
```

# Print out TriggerMenu

- Print out the trigger menu to see which ones have fired and which ones are available

```
trigDec->print();
```

- Set the output level to INFO in the job options (TriggerTool\_jobOptions.py)
  - Print method checks the debug level and suppresses any messages if it is less than MSG::INFO

```
MessageSvc.OutputLevel = INFO
```

- To see the output of the TriggerDecision, set the overall message level to INFO
- This is done in:
  - TriggerTool\_1.cxx
  - TriggerTool\_1.h
- Output in log\_1.out

# Caveats in the Trigger menu in 12.0.6

- **Jets:** at HLT, e.g. 2j120 which is started at L1 by 2J45
  - Due to limitation of trigger menu items at L1 (hardware trigger)
  - Jet trigger currently start with very low thresholds which are not the actual L1 trigger thresholds due to late readines of slice for 12.0.X

ttHSelection.Tr...	INFO trigger	L1	L2	EF
ttHSelection.Tr...	INFO j160 trigger	1	0	0
ttHSelection.Tr...	INFO 2j120 trigger	1	0	0
ttHSelection.Tr...	INFO 3j65 trigger	1	0	0
ttHSelection.Tr...	INFO 4j50 trigger	1	1	1

- **B-jets:** trigger runs currently in accept all mode in HLT, no selection cuts applied

ttHSelection.Tr...	INFO bjet35 trigger	1	1	1
--------------------	---------------------	---	---	---

- **Forward Jets:** trigger not working properly in 12.0.X, no events will be selected by it (will be fixed in release 13)

ttHSelection.Tr...	INFO fj30 trigger	0		
--------------------	-------------------	---	--	--

# Caveats in the Trigger menu in 12.0.6 (cont)

- **Muons:** MuCPTI provides only the highest L1 muon threshold fired
  - e.g.: event with 50GeV muon, L1\_MU40 fired (nearest threshold), but L1\_MU08, L1\_MU10 etc. not passed.
  - To get correct trigger decision, all higher thresholds need to be checked

```
TriggerDecision... INFO TriggerDecision: L1 Trigger Item L1_MU06 did not pass
TriggerDecision... INFO TriggerDecision: L1 Trigger Item L1_MU08 did not pass
TriggerDecision... INFO TriggerDecision: L1 Trigger Item L1_MU10 did not pass
TriggerDecision... INFO TriggerDecision: L1 Trigger Item L1_MU11 is fulfilled
TriggerDecision... INFO TriggerDecision: L1 Trigger Item L1_MU20 did not pass
TriggerDecision... INFO TriggerDecision: L1 Trigger Item L1_MU40 did not pass
```

- Isolation is currently not implemented for muons in release 12
- **Electrons:** level 2 spacepoint bug
  - Problem in reconstruction where tracks aren't formed at L2 due to missing code
  - Leads to non-flat efficiencies in eta at L2
  - Can only be fixed by re-reconstruction from RDO's
  - <http://indico.cern.ch/getFile.py/access?contribId=52&sessionId=4&resId=0&materialId=slides&confId=16155> (slide 5)

# Caveats in the Trigger menu in 12.0.6 (cont)

- **Mixed Triggers:** currently can't deal with mixed triggers as eg. J45+tau15i
  - Need to add those by hand

```
If (trigDec->isTriggered("EF_jet45") && trigDec->isTriggered("EF_tau15i"))
```

- **Prescaled Triggers:** currently no prescaled factors applied
  - Look at <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerPhysicsMenu>

# Print Statistics for each event

- we want to know if the event was selected by the various triggers and get some statistics
- Set-up counters and figure out if a given trigger has fired
  - `isTriggered()` : has event passed a given trigger item
  - `isDefined()` : is given trigger signature for given trigger level run in this event? (Level1 = 1, Level2 = 2, EventFilter =EF)
  - `print()` : print list of triggers plus trigger decision per event
- This is again already done for you in:
  - `TriggerTool_2.cxx`
  - `TriggerTool_2.h`
- Output can be found in `log_2.out`



# Analyse Ntuple

- Ntuple analysis.ntuple.trigger1/2/3/4.root has some info about the trigger
  - trigger items available
  - PDG id and truth  $p_T$  of leptons (electrons & muons)
- makeHistos.C provides access to the variable of the ntuple
- Look at summary of trigger statistics
- Get turn on curve for ttH relevant trigger e25i and mu20i

# Print Statistics Summary

- Print trigger summary for all events
- Include myTriggerOutput.C in makeHist.C

# Example for mixed trigger

- Currently can't deal with mixed triggers, not supported by the current trigger steering (but will be from release 13 on)
- To include mixed triggers anyway, add those by hand in analysis code:
- Example:  

```
(trigDec->isTriggered("EF_tau10i") &&  
  trigDec->isTriggered("EF_mu6"))
```
- Edit makeHist.C

# Example for pre-scaled trigger

- Currently no pre-scaled factors applied in trigger menu (have to be studied)
- To apply pre-scales in analysis:
  - Ask if trigger has fired
  - Only accept 1/(pre-scale factor) of the events
- Example: mu6 trigger, apply prescale factor of 5
  - Declare variables:
    - `int m_counter; m_counter=0;`
    - `int m_prescale; m_prescale=5;`
  - In execute:
    - ```
if (trigDec->isTriggered("EF_mu6")) {  
    m_counter+=1;  
    if (m_counter%m_prescale == 0) m_EFmu6pre+=1;  
}
```

More Advanced Option...

# Re-run the hypothesis on AOD

- For more detailed information:

<https://twiki.cern.ch/twiki/bin/view/Atlas/TrigTutorialRerunHypo>

- Do this if you want to:
  - optimise given trigger items
  - test new optimisations on physic channels
  - develop new triggers
- First look at reconstructed csc produced data and then re-optimize the trigger selection cuts
- To do this, run the TriggerSteering asking to re-run the hypothesis for your 'favourite' trigger menu item(s).
- Note: if the trigger is **re-run** on an ESD or AOD file which already contains a TriggerDecision object, TriggerDecisionMaker will create a new TriggerDecision object and store it with the key "MyTriggerDecision+ ".

# Retrieving the last TriggerDecision (cont)

```
m_TriggerDecisionKey="MyTriggerDecision";
std::string new_TD = m_TriggerDecisionKey;
std::string lastTD = m_TriggerDecisionKey;

bool old_td = false;
do {
    if (m_storeGate->contains<TriggerDecision>( new_TD )) {
        mLog << MSG::DEBUG << "TriggerDecision with key " << new_TD << " found in
            StoreGate; looking for " << new_TD+"<<" << endreq;
        old_td = true;                // flag for do...while loop
        lastTD = new_TD;              // keep last TD key before updating new
        new_TD = new_TD + "+";       // update StoreGate key
    } else {
        old_td = false;              // to stop loop
        mLog << MSG::DEBUG << "No TriggerDecision with key " << new_TD << " found
            in StoreGate; using key " << lastTD << endreq;
    }
} while (old_td);

// retrieve latest TriggerDecision
const TriggerDecision* trigDec = NULL;
sc = m_storeGate->retrieve(trigDec, lastTD);
if ( sc.isFailure() ) {
    mLog << MSG::WARNING << "Failed to retrieve TriggerDecision" << endreq;
    return StatusCode::SUCCESS;
} else {
    mLog << MSG::INFO << "TriggerDecision successfully retrieved" << endreq;
```

# Summary

- Triggering is challenging and an essential part of any physics analysis
- The trigger software is usable in data production for AOD's
  - Still some problems, many of them will be fixed in release 13



# Useful Links

- Wiki page for TriggerDecision:  
<https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerDecision>
- Trigger Software Tutorial Page:  
<https://twiki.cern.ch/twiki/bin/view/Atlas//TriggerSoftwareTutorialPage>
- Hypernews  
<https://hypernews.cern.ch/HyperNews/Atlas/get/TriggerHelp.html>
- TriggerUserPages  
<https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerUserPages>

I would like to thank....

Nikos Konstandinids

Simon Dean

Adam Davison