

# Table of Contents

<b>4.6 HLT Tutorial.....</b>	<b>1</b>
Goals of this page:.....	1
Contents.....	1
Introduction.....	1
Producing event with Triggers.....	2
A quick look at the code.....	3
Running Trigger paths.....	3
Analyzing Trigger/Offline information.....	3
Getting the L1 information.....	4
Getting the HLT information.....	4
Getting the Offline-reconstructed and other information.....	4
Getting the MC truth information.....	5
Calculate the rate of a new trigger:.....	5
Recipes for producing ntuples.....	5
Review status.....	5

## 4.6 HLT Tutorial

Complete:

Detailed Review status

### Newsbox

This page has been updated for CMSSW 52X and the latest 2012 HLT studies

## Goals of this page:

This page is intended to familiarize you with HLT ideas, software and utilities. In particular, you will learn:

- how to run various trigger paths,
- how to analyze HLT related information.

## Contents

- Introduction
- Producing events with Triggers
  - ◆ A quick look at the code
  - ◆ Running Trigger paths
- Analyzing Trigger/Offline information
  - ◆ L1 Information
  - ◆ HLT Information
  - ◆ Offline-reconstructed Information
  - ◆ MC truth information
  - ◆ Calculate the rate of a new trigger
  - ◆ Recipes
  - ◆ Explanation of branch tags
- Review Status

## Introduction

Why a trigger ?

- Most of events/processes produced at a high-energy hadronic collider are not interesting : an early decision, i.e. online selection, has to be made to avoid running the offline code on millions of uninteresting events

High Level Trigger (HLT) :

- Takes events accepted by the L1, first level of the CMS Trigger
- Decides, based on more elaborated algorithms, whether the event should be kept

Therefore, it is a crucial part of the CMS data flow since it's the HLT algorithms and filters which will decide whether an event should be kept for an offline analysis : any offline analysis depends on the outcome of HLT, i.e. on the HLT efficiency.

Important benchmark ideas :

- Rates (at 2E33) :
  - ◆ 13 MHz -> L1 -> 50kHz
  - ◆ -> HLT -> 150 Hz
- Conceive and run approx. 200 triggers, as efficient as possible
- Reconstruction :
  - ◆ Seeded by L1
  - ◆ As close as possible to offline reconstruction
  - ◆ Regional reconstruction : saving CPU
  - ◆ HLT level : typically :
    - ◇ L2 : Calorimeter and Muon information
    - ◇ L3 : Tracking information

Example of HLT reconstruction : Muons :

- L2 :
    - ◆ Uses L1 seeds : Up to 4 muons provided by Global Muon Trigger (GMT), information : (pT, charge, phi, eta)
    - ◆ "StandAlone" muons : using muon segments, charge clusters, then Outside->In fitting
    - ◆ Filters : pT, invariant-mass
    - ◆ Filters : Calorimeter-based isolation
  - L3 :
    - ◆ Uses L2 seeds
    - ◆ "Global regional muon reconstruction" : using tracker information in muon window
    - ◆ Filters : pT, invariant-mass, Impact-Parameter, track-quality
    - ◆ Filters : Tracker-based isolation
- 

Please do visit Trigger, HLT pages :

- CMS Trigger TDR : [Trigger TDR](#)
- Trigger Studies Group page : [Main page of TSG](#)
- Trigger software guide : [Trigger software guide](#)

Subscribe to hypernews :

- [hn-cms-online-selection@cernNOSPAMPLEASE.ch](mailto:hn-cms-online-selection@cern.ch) : general trigger discussions
- [hn-cms-hlt@cernNOSPAMPLEASE.ch](mailto:hn-cms-hlt@cern.ch) : HLT Software

## Producing event with Triggers

Triggers defined as `path` blocks : sequence of modules and operands :

```
path MyTrigger = {doL1Reco & doL1Seeding & ApplyPrescale & doHltReco, HltCondition}
```

Modules :

- **Reconstruction** : `doL1Reco`, `doHltReco`, etc...
- **Prescale** : `ApplyPrescale`
- **Filter** : `HltCondition`

Operands :

- the `,` or dependency operator, the operand to the right is dependent on the operand on the left (i.e. the right accesses data produced by the left)

- the "&" or sequencing operator, the operand on the left is executed first, followed by the operand on the right, but they are not dependent on each other

Consequences :

1. The result of each operand is a boolean : the final outcome of `MyTrigger` is "reject" or "accept"
2. For a given path, the overall answer is the "AND" of all operands : If ever one of the operands on the left fails, reject AND stop processing : Saving CPU time !

Recommended reading : [PathTriggerBits](#)

The HLT configuration in CMS is stored in a dedicated database system, ConfDB. Use the dedicated GUI to create, manipulate and store trigger path configurations. Use the ConfDB web browser [☞](#) to browse the content of the configuration database and inspect available HLT menus. The command-line tool `edmConfigFromDB` allows you to retrieve (complete or partial) configurations as either ascii or python configuration files to be fed to `cmsRun`.

## A quick look at the code

Please visit :

- Trigger software guide : [TriggerStudySW](#)
- L1-HLT software guide : [L1Extra](#)

HLT code in [HLTrigger](#) [☞](#)

## Running Trigger paths

Follow instructions from [HLTtable](#)

For input files : go on the [Data Aggregation System](#) page [DAS](#) [☞](#)

## Analyzing Trigger/Offline information

Motivation : Get in one place, on event-by-event basis :

- L1-, HLT-related information
- Offline reconstructed information
- HLT information without filters: "Open HLT" mode

Enables to :

1. Study Trigger efficiencies, as function of offline reconstructed quantities...
2. Get Trigger rejections, overlaps, rates...
3. ...for L1, HLT conditions

Code :

- Package : `HLTrigger/HLTanalyzers` : [HLTAnalyzers](#) [☞](#)
- Driving code : `HLTAnalyzer .h .cc` : [an EDAnalyzer](#)

- Configuration file to run : `HLTrigger/HLTAnalyzers/test/HLTAnalysis.cfg`

More details in HLTAnaManual

## Getting the L1 information

Motivation : Have the information of HLT seeds at disposal

Using `L1Extra` objects, from `MCTruth` or from the L1 Emulator : In the `HLTAnalysis.cfg` file :

```
module HLTAnalyzer = {
  ...
  string l1extramc = l1extraParticles
  ...
}
```

Physics objects	Variables stored	L1Extra Class	Instances
Muons	E, pT, phi, eta, isolation, mip	L1MuonParticle	
EM particles	E, ET, phi, eta	L1EmParticle	"Isolated" and "NonIsolated"
Jets	E, ET, phi, eta	L1JetParticle	"Forward" and "Central"
Taus	E, ET, phi, eta	L1JetParticle	"Tau"
MET	ET, ET(tot), ET(had), phi	L1EtMissParticle	

Branches of variables created per instance.

Code in : `HLTInfo .h .cc`

## Getting the HLT information

How do I get information about Trigger Results ?

- From the `TriggerResults` class : Associate Trigger Path to Decision. `HLTAnalyzer` dynamically creates as many branches as Triggers present, and fill them with corresponding Trigger decision. Trigger branches pop up in form of "TRIGG\_".

Code in : `HLTInfo .h .cc`

If you want to get the names of the paths (e.g. for accessing them by path name rather than by bit number), see the example code in `HLTrigger/HLTAnalyzers/src/HLTrigReport.cc` [↗](#).

See also the hypernews discussion here [↗](#).

## Getting the Offline-reconstructed and other information

Specify in `HLTAnalysis.cfg` the instances of reconstructed objects that you want :

```
module HLTAnalyzer =
  ...
  string muon = "muons"
  string Electron = "pixelMatchGsfElectrons"
  string Photon = "correctedPhotons"
  string recjets = "iterativeCone5CMS.CaloJets"
  string genjets = "iterativeCone5GenJets"
  string recmet = met
```

```
string genmet = genMet
string calotowers = towerMaker
...
}
```

Physics objects	Variables stored	Instances
Muon	E, ET, pT, eta, phi	muons
Electron	E, ET, pT, eta, phi	pixelMatchGsfElectrons
Gamma	E, ET, pT, eta, phi	correctedPhotons
Jet	E, ET, pT, eta, phi	iterativeCone5CMS.CaloJets, iterativeCone5GenJets
MET	pT, phi, ET(sum)	met, genMet
Calo Towers	E, ET, E(em), E(had), E(out), eta, phi	towerMaker
MC truth	Id, Vtx(X,Y,Z), pT	

Code in : HLTGamma, HLTMuon, HLTJets (deals with MET as well)

- Getting also information about the MC truth information about generated particles : pT, Identity, Vertex... Code in HLTMCtruth

Offline collections can be accessed when running simultaneously and RAW+RECO samples. This can be achieved using crab with use\_parent=1 option. Running on a RECO dataset will then allow to access its parent dataset, namely the corresponding RAW dataset.

## Getting the MC truth information

Motivation : Have the generator level information at disposal

Object	Variables stored	Instances
Generated particle	PDG-Identity, particle status, [x,y,z]-component of primary vertex	genParticles
Generated particle	pT, phi, eta	genParticles
	Event pT-hat, i.e. scale	genEventScale

### Calculate the rate of a new trigger:

Please have a look at: How to run OpenHLT

You may also want to take a look at the recipes from the STEAM group

## Recipes for producing ntuples

These versions of the analyzer have the OpenHLT capability included.

For additional tags needed to run on real data consistent with online releases, consult this page.

For the correct GlobalTag to use in a given release consult this page.

## Review status

Reviewer/Editor and Date	Comments
Pedrame Bargassa - 1 Dec 2007	Author

XuanChen - 31 Jul 2014	changed links from cvs to github, replaced dbs by das
------------------------	-------------------------------------------------------

Responsible: Pedrame Bargassa

Last reviewed by: ChristosLeonidopoulos - 22 Feb 2008

---

This topic: CMSPublic > WorkBookHLTTutorial

Topic revision: r201 - 2014-10-29 - MurielVanderDonckt



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)