Table of Contents

4.2.3 Physics Analysis Toolkit (PAT): Configuration

Detailed Review status

Contents

Introduction

In this section you will find a description of the configuration of the workflow and content of the pat::Candidates **(1)** . You may access this content from the pat::Candidate member functions. To learn more about the content of each pat::Candidate have a look to WorkBookPATDataFormats. The following groups of modules will be described in the following:

- Candidate Configuration: configuration of the content of the basic pat::Candidate.
- Selection Configuration: configuration of selected pat::Candidate collections.
- Disambiguation Configuration: configuration of object disambiguation extra information to be added to the pat::Candidate.

The modules for the creation of the pat::TriggerEvent and the matching of trigger candidates to pat::Candidates is described in SWGuidePATTrigger#Producers. You can easily browse each configuration file exploiting the *edmConfigEditor* as explained on WorkBookConfigEditor.

(1) To learn more about regularly used expressions (like pat::Candidate, pat::Tuple, ...) have a look to the WorkBookPATGlossary.

Candidate Configuration

In this section you will find a description how to (re-)configure the content of each pat::Candidate and the corresponding default configurations. In the standard workflow the pat::Candidate collections are the first object collections to be produced after a pre-creation phase. The basic pat::Candidate collections are further processed in a selection phase and finally in a optional phase of candidate disambiguation. To learn more about the PAT workflow have a look to WorkBookPATWorkflow. You will find the configuration files discussed on this page in the python/producersLayer1 directory of the PatAlgos package.

pat::Photon

Description:

The standard module to produce pat::Photons is the **PATPhotonProducer** module. It produces the patPhotons collection. You can get an automated description of the module and its parameters using the edmPluginHelp tool described on SWGuideConfigurationValidationAndHelp. You can find the module implementation here.

Configurables:

You can find the full list of configuration parameters and the standard configuration of the PATPhotonProducer module here.

Algorithms:

You can find the links to the used algorithms for the implementation of the pat::Photons below. Note that the default implementation of algorithms as recommended and used by the corresponding POG or the Analysis Tools (AT) group are used. Therefore the following links will mostly point to the documentation as provided by these groups.

- WorkBookMCTruthMatch: a workbook description of MC truth matching tools provided by the AT group.
- SWGuidePATMCMatching : a description of the PAT implementation of the MC truth matching tools as described above.
- SWGuidePATTrigger: a complete description of the pat::Trigger.
- SWGuidePATTriggerMatching: a description of the matching of trigger information to offline reconstructed objects.
- SWGuideEgammaSpikeCleaning: a description of the spike cleaning needed for the analysis of photons and electrons.
- SWGuideEgammaIsolation: a description of the standard isolation algorithms for photons.
- Photon_ID_criteria: description of the photon identification.

---

pat::Electron

Description:

The standard module to produce pat::Electrons is the **PATElectronProducer** module. It produces the patElectrons collection. You can get an automated description of the module and its parameters using the edmPluginHelp tool described on SWGuideConfigurationValidationAndHelp. You can find the module implementation here.

Configurables:

You can find the full list of configuration parameters and the standard configuration of the patElectrons module here.

Algorithms:

You can find the links to the used algorithms for the implementation of the pat::Electrons below. Note that the default implementation of algorithms as recommended and used by the corresponding POG or the Analysis

Tools (AT) group are used. Therefore the following links will mostly point to the documentation as provided by these groups.

- WorkBookMCTruthMatch: a workbook description of MC truth matching tools provided by the AT group.
- SWGuidePATMCMatching : a description of the PAT implementation of the MC truth matching tools as described above.
- SWGuidePATTrigger: a complete description of the pat::Trigger.
- SWGuidePATTriggerMatching: a description of the matching of trigger information to offline reconstructed objects.
- SWGuideEgammaSpikeCleaning: a description of the spike cleaning needed for the analysis of photons and electrons.
- SWGuideEgammaIsolation: a description of the standard isolation algorithms for electrons.
- EgIdentification#Electron_ID_criteria: description of the electron identification.

---

pat::Muon

Description:

The standard module to produce pat::Muons is the **PATMuonProducer** module. It produces the patMuons collection. You can get an automated description of the module and its parameters using the edmPluginHelp tool described on SWGuideConfigurationValidationAndHelp. You can find the module implementation here.

Configurables:

You can find the full list of configuration parameters and the standard configuration of the patMuons module here.

Algorithms:

You can find the links to the used algorithms for the implementation of the pat::Muons below. Note that the default implementation of algorithms as recommended and used by the corresponding POG or the Analysis Tools (AT) group are used. Therefore the following links will mostly point to the documentation as provided by these groups.

- WorkBookMCTruthMatch: a workbook description of MC truth matching tools provided by the AT group.
- SWGuidePATMCMatching : a description of the PAT implementation of the MC truth matching tools as described above.
- SWGuidePATTrigger: a complete description of the pat::Trigger.
- SWGuidePATTriggerMatching: a description of the matching of trigger information to offline reconstructed objects.
- SWGuideMuonIsolation : a description of the standard isolation algorithms and a description of the standard content of isoDeposits.
- WorkBookMuonID : a description of the object identification as recommended by the CMS.MuonPOG. You can find a detailed study on muon identification in this note CMS-AN2008/098

---

pat::Tau

Description:

The standard module to produce pat::Taus is the **PATTauProducer** module. It produces the patTaus collection. You can get an automated description of the module and its parameters using the edmPluginHelp tool described on SWGuideConfigurationValidationAndHelp. You can find the module implementation here.

Configurables:

You can find the full list of configuration parameters and the standard configuration of the patTaus module here.

Algorithms:

You can find the links to the used algorithms for the implementation of the pat::Taus below. Note that the default implementation of algorithms as recommended and used by the corresponding POG or the Analysis Tools (AT) group are used. Therefore the following links will mostly point to the documentation as provided by these groups.

- WorkBookMCTruthMatch: a workbook description of MC truth matching tools provided by the AT group.
- SWGuidePATMCMatching : a description of the PAT implementation of the MC truth matching tools as described above.
- SWGuidePATTrigger: a complete description of the pat::Trigger.
- SWGuidePATTriggerMatching: a description of the matching of trigger information to offline reconstructed objects.
- SWGuidePFTauID : a description of the tau discrimination algorithms.

---

pat::Jet

Description:

The standard module to produce pat::Jets is the **PATJetProducer** module. It produces the patJets collection. You can get an automated description of the module and its parameters using the edmPluginHelp tool described on SWGuideConfigurationValidationAndHelp. You can find the module implementation here.

Configurables:

You can find the full list of configuration parameters and the standard configuration of the patJets module here.

Algorithms:

You can find the links to the used algorithms for the implementation of the pat::Jets below. Note that the default implementation of algorithms as recommended and used by the corresponding POG or the Analysis Tools (AT) group are used. Therefore the following links will mostly point to the documentation as provided by these groups.

- WorkBookMCTruthMatch: a workbook description of MC truth matching tools provided by the AT group.
- SWGuidePATMCMatching : a description of the PAT implementation of the MC truth matching tools as described above.
- SWGuidePATTrigger: a complete description of the pat::Trigger.
- SWGuidePATTriggerMatching: a description of the matching of trigger information to offline reconstructed objects.
- SWGuideJetAlgorithm: a description of the various jet algorithms used at CMS.
- WorkBookJetCorrections : a description of the jet energy scale correction procedures.
- SWGuideJetCharge : a description of the jet charge determination.
- SWGuideJetFlavor : a description of the jet flavour determination.
- SWGuideBTagging : a description of the b-tag algorithms.

---

pat::MET

Description:

The standard module to produce pat::MET is the **PATMETProducer** module. It produces the patMets collection. You can get an automated description of the module and its parameters using the edmPluginHelp tool described on SWGuideConfigurationValidationAndHelp. You can find the module implementation here.

Configurables:

You can find the full list of configuration parameters and the standard configuration of the patMETs module here.

Algorithms:

You can find the links to the used algorithms for the implementation of the pat::Jets below. Note that the default implementation of algorithms as recommended and used by the corresponding POG or the Analysis Tools (AT) group are used. Therefore the following links will mostly point to the documentation as provided by these groups.

- SWGuidePATTrigger: a complete description of the pat::Trigger.
- SWGuidePATTriggerMatching: a description of the matching of trigger information to offline reconstructed objects.
- MET Type1 Correction : a description of the MET type1 corrections.
- MET Muon Correction : a description of the MET muon corrections.

User-Defined Isolation for Leptons

reco::Photons, reco::Electrons and reco::Muons contain pre-defined isolation values recommended by the corresponding POGs. Via the pat::Candidate you can access the most common isolation values via the following member functions:

- `trackIso()`: isolation in the tracker.
- `ecalIso()`: isolatzion in the ECAL.
- `hcalIso()`: isolation in the HCAL.
- `caloIso()`: combined ECAL and HCAL isolation.

Apart from that you may have access to more detailed isolation information (especially for the reco::Muon) via the corresponding reco::Candidate member functions. In parallel PAT offers the possibility to exploit a flexible user-defined isolation for photons, electrons muons, taus and generic particles (i.e. tracks). You can access it via the member function `userIsolation(pat::IsolationKey key)` as defined under WorkBookPATDataFormats#UserIsolation. Corresponding isolation values need to be filled during the creation process of the pat::Candidate collection in the *PSet* `userIsolation` in the cfi file of the corresponding pat::Candidate collection. This *PSet* may consist of one or more additional *PSets* with the following names:

PSet	IsolationKey
`tracker`	`pat::TrackerIso`
`ecal`	`pat::EcalIso`
`hcal`	`pat::HcalIso`
`calo`	`pat::CaloIso`
`user`	`pat::User1Iso`
	`pat::User2Iso`
	`pat::User3Iso`
	`pat::User4Iso`
	`pat::User5Iso`

where `user` stands for a *std::vector* of *PSets* which in the current implementation can have a maximal length of 5. The *PSets* can be of the following two types:

SimpleIsolator:

Show Hide

```
cms.PSet(
   src = cms.InputTag("edm::ValueMap"),
)
```

where `edm::ValueMap` is expected to be an `edm::ValueMap` associating a predefined isolation value to the original reco::Candidate.

IsoDepositIsolator:

Show Hide

```
cms.PSet(
   src = cms.InputTag("edm::ValueMap"),
   deltaR = cms.double(0.3),
   mode = cms.string("mode"),
   veto = cms.double(0.01),
  threshold = cms.double(0.05)
)
```

where `edm::ValueMap` is expected to be an `edm::ValueMap` associating a predefined set of isoDeposits to the original reco::Candidate and the other parameters have the following meaning:

parameter	type	meaning
`deltaR`	double	size of the isolation cone
`veto`	double	size of a potential veto cone
`threshold`	double	size of a given threshold value (in GeV)
`mode`	string	mode to sum up the isoDeposits

The parameter `deltaR` is a key parameter for calculating user-defined isolation values from isoDesposits, while the other parameters are optional. The parameter `mode` allows for the following configurations:

- `sum`: absolute sum of isoDeposits in cone.
- `sumRelative`: relative sum of isoDeposits in cone.

- `max`: absolute maximum of isoDeposits in cone.
- `maxRelative`: relative maximum of isoDeposits in cone.
- `sum2`: absolute sum of isoDeposits in cone squared.
- `sum2Relative`: relative sum of isoDeposits in cone squared.
- `count`: number of isoDeposits in cone.

Note:

User defined isolation is also available for *pat::GenericParticles* e.g. for tracks.

Selection Configuration

There is a string based PatCandidateSelector to apply selections to the pat::Candidate collections in an intuitive way. The string configuration can have all kinds of functions and all member functions of the corresponding pat::Candidate as input. For a description of the string parser have a look at SWGuidePhysicsCutParser.

The PatCandidateSelector produces the selectedPatCandidate collections, which form the standard pat::Candidate collections if object disambiguation is not applied. Per default the input to the PatCandidateSelector are the patCandidates, but it can be any kind of pat::Candidate collection. In the standard workflow of pat::Candidate creation the selectedPatCandidate collections are produced directly after the creation of the basic pat::Candidates. For a detailed description of the standard workflow of pat::Candidate production have a look at WorkBookPATWorkflow.

You will find the configuration files for the selection of pat::Candidates in the python/selectionLayer1 directory of the PatAlgos package. Links to the default selection strings for the most important pat::Candidate collections are listed below:

- selectedPatPhotons
- selectedPatElectrons
- selectedPatMuons
- selectedPatTaus
- selectedPatJets

Note: there is no PATCandidateSelector for pat::MET (this might change in future).

Disambiguation Configuration

Due to the organisation of the physics object reconstruction in CMS an energy deposition in the CMS detector may be reconstructed as different objects at the same time. E.g. an energy deposit in the electromagnetic calorimeter may be reconstructed as an electron, a photon and a jet with electromagnetic fraction close to 1 at the same time. There is a configurable PatCandidateCleaner to apply some extra information for candidate disambiguation to the most important pat::Candidate collections during later analysis steps. For a more detailed description of the PatCandidateCleaner have a look at WorkBookPATWorkflow.

The PatCandidateCleaner produces the cleanedPatCandidate collections, which form the standard pat::Candidate collections including object cross cleaning. Per default the input to the PatCandidateCleaner are the selectedPatCandidates, but it can be any kind of pat::Candidate collection. In the standard workflow of pat::Candidate production the cleanedPatCandidate collections are produced after the selectedPatCandidate collections have been created. For a detailed description of the standard workflow of pat::Candidate creation have a look at WorkBookPATWorkflow.

You will find the configuration files for candidate disambiguation in the python/cleaningLayer1 directory of the PatAlgos package. Links to the default configuration for the most important pat::Candidate collections are listed below:

- cleanPatPhotons
- cleanPatElectrons
- cleanPatMuons
- cleanPatTaus
- cleanPatJets

Note:

There is no PATCandidateCleaner for pat::MET.

Review status

Show Hide

Reviewer/Editor and Date (copy from screen)	Comments
RogerWolf - 26 Aug 2010	Review for PAT Sept Tutorial

Responsible: RogerWolf

This topic: CMSPublic > WorkBookPATConfiguration
Topic revision: r16 - 2013-07-02 - SudhirMalik