

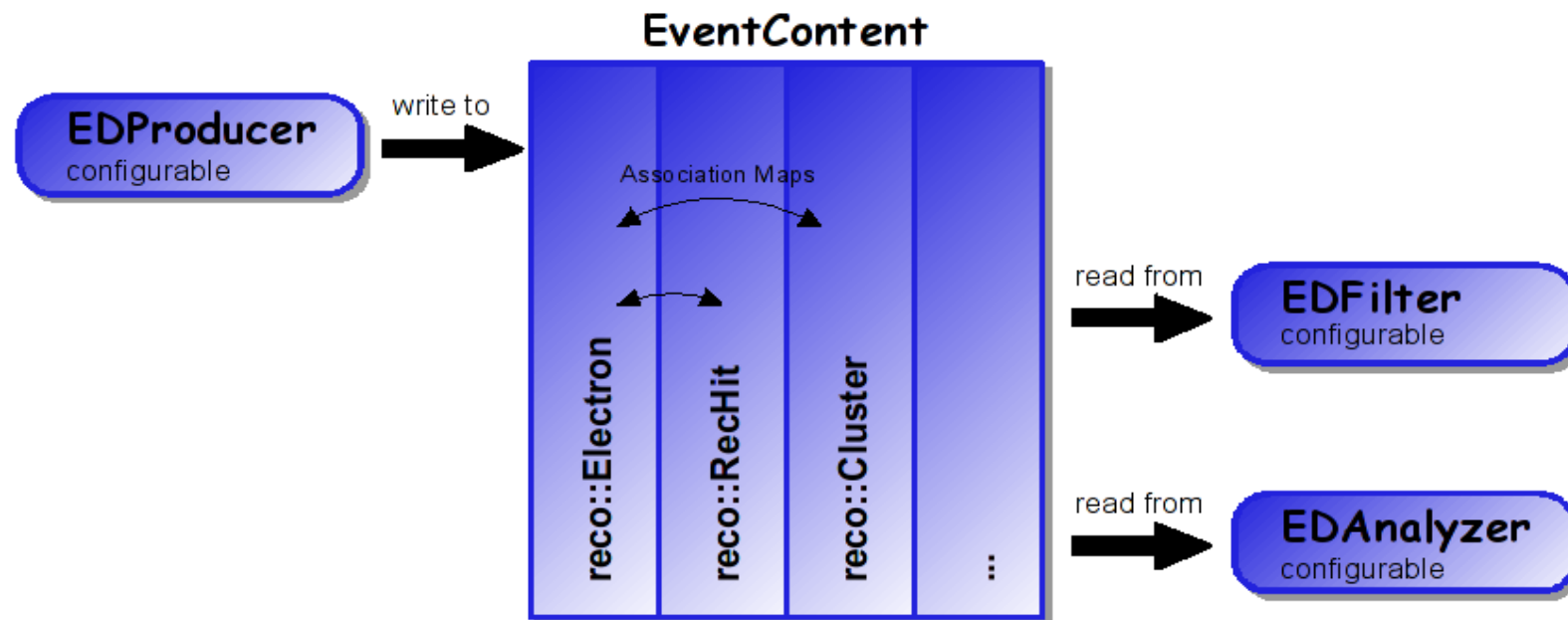
# What is PAT and How to use it

*Roger Wolf*

PAT Tutorial 15.06.2009

# The Event Data Model

- Major concept of CMSSW
- Fully configurable `edm::Modules` communicate via EventContent



- Same file structure (root) for: **Gen-Sim-Digi-Reco-Analysis**
- Single framework for Reconstruction (POGs) and Analysis (PAGs)



# The Physics Analysis Toolkit

## Interface

- b/w Reconstruction & Analysis Level
- simplifies access via **DataFormats**
- canalizes expertise (POG & PAG contacts)
- crossing point between POGs & PAGs ('vertical integration')

## Common Tool

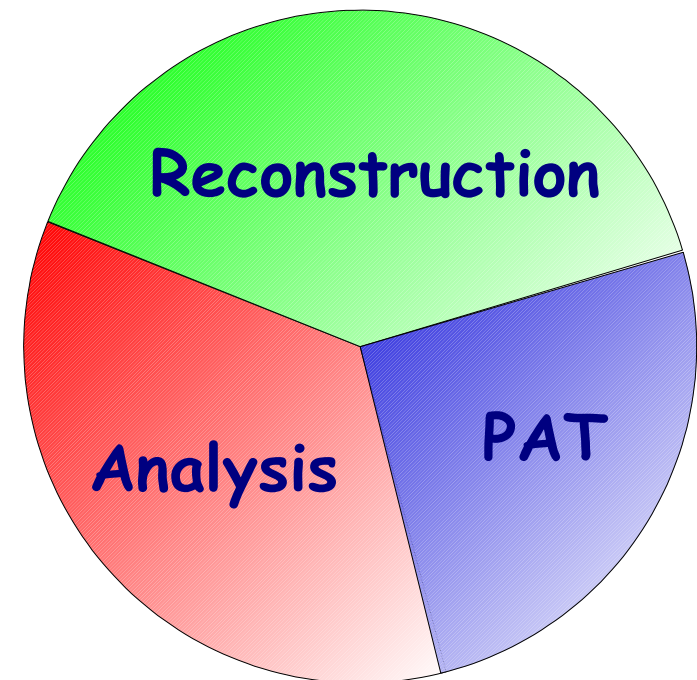
- approved algorithms & sensible defaults
- synergy (everybody can profit from recent developments)
- quick start into analysis for beginners

## Common Format

- facilitates transfer & comparisons
- PAG common configurations
- sustained provenance

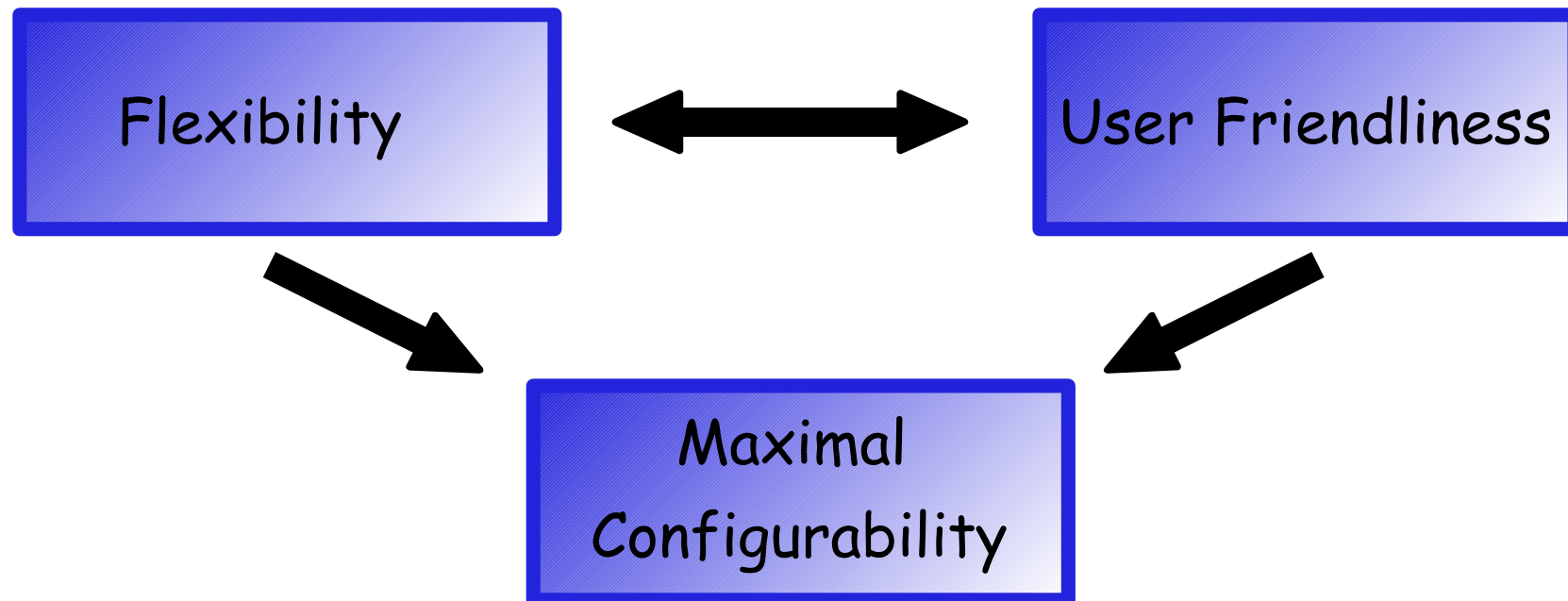
# 'Disclaimer'

- PAT does **NOT** re-invent the wheel
  - prevents re-inventions
  - helps standardize & spread finest knowledge in a collaboration of 3000 physicists
- PAT is **NOT** an antipode to CMSSW
  - fully CMSSW contained
  - prevents multiple frameworks within the framework
- Consequence & Completion of the Event Data Model



# Concepts

- Make use of the modular structure of CMSSW (in full FW/FWLite)
- Provide easy access via member functions in **DataFormats**
- Serve 80% of all analyses in CMS



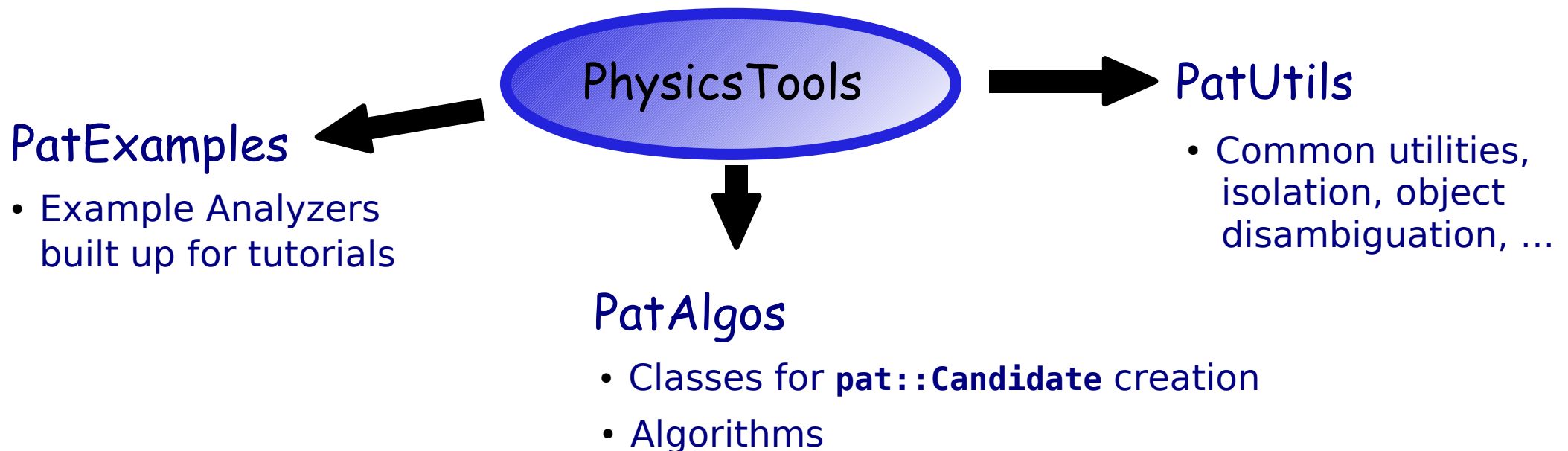


# Code Location

- All code located in the CMSSW domain distributed over two systems

- **DataFormats/PatCandidates**

- Structures and Candidate Classes
- `pat::Photon`, `pat::Electron`, `pat::Muon`, `pat::Jet`, `pat::MET`, ...



# Development

- Antagonism between code development & user's needs
- Provide stability and development (beyond releases)
  - Installation Recipes with new developments/fixes for the user

## CMSSW\_2\_2\_X (with PAT version 2)

**Latest tag:** To check out the latest tags do the following

[Show ▶](#)

**Head version:** To check out the head version of the B22X development branch do the following

[Show ▶](#)

**Important Notes:** For further information about the use of PAT (version 2) have a look below:

[Show ▶](#)

See the corresponding [Release Notes](#) for details.

- Several layers of development for the developer

| Branch             | Release      | PATCandidates | PATAlgos  | PATUtils     | Comments           |
|--------------------|--------------|---------------|-----------|--------------|--------------------|
| B22X_Production_v1 | CMSSW_2_2_9  | V03-18-ZZ     | V04-14-ZZ | V03-05-02-ZZ | no development     |
| B2_2_X             | CMSSW_2_2_10 | V03-YY-ZZ     | V05-YY-ZZ | V03-06-ZZ    | development branch |
| HEAD               | CMSSW_3_1_X  | V05-YY-ZZ     | V07-YY-ZZ | V03-YY-ZZ    | development branch |

Have a look at **SWGuidePATRecipes** for more details

# Installation Recipes

- Recommended Installation Recipes
  - Stable and tested to our best knowledge (validation in progress)
  - Regularly updated
  - Can be applied by copy & paste
  - First aim: push these tags into new releases
  - We also provide 'Release Notes'

## CMSSW\_2\_2\_X (with PAT version 2)

**Latest tag:** To check out the latest tags do the following

[Hide](#) 

```
cmsrel CMSSW_2_2_13
cd CMSSW_2_2_13/src
cmsenv
addpkg DataFormats/PatCandidates V03-26-05
addpkg PhysicsTools/PatAlgos V05-05-10
addpkg PhysicsTools/PatUtils V03-06-03
```

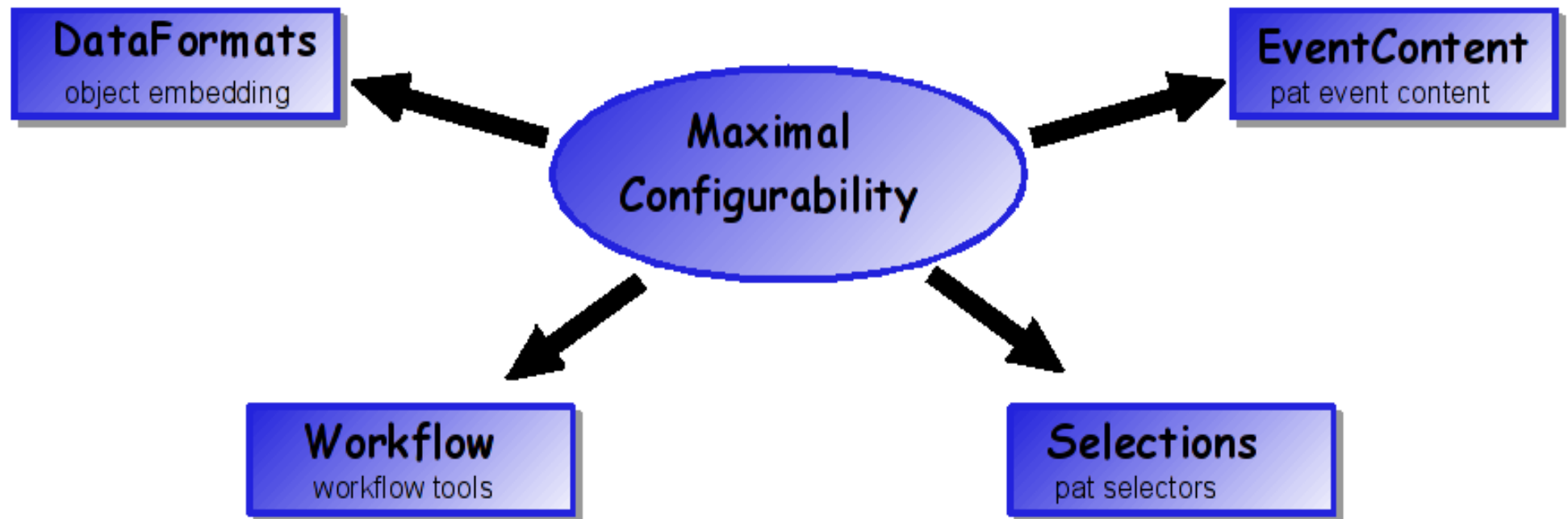
These packages have the following dependencies:

```
addpkg CondFormats/JetMETObjects V01-08-04
addpkg PhysicsTools/RecoAlgos V08-06-16-06-02
addpkg PhysicsTools/PFCandProducer V03-01-16
addpkg RecoMET/Configuration V00-04-02-17
addpkg RecoMET/METAlgorithms V02-05-00-21
addpkg RecoMET/METProducers V02-08-02-17
addpkg DataFormats/METReco V00-06-02-09
addpkg DataFormats/MuonReco V07-02-12-03
addpkg JetMETCorrections/Type1MET VB04-00-02-04
addpkg RecoJets/JetAssociationAlgorithms V01-04-03
addpkg JetMETCorrections/Algorithms V01-08-02-01
addpkg JetMETCorrections/Configuration V01-08-15
addpkg JetMETCorrections/JetPlusTrack V03-02-06
addpkg JetMETCorrections/Modules V02-09-02
```



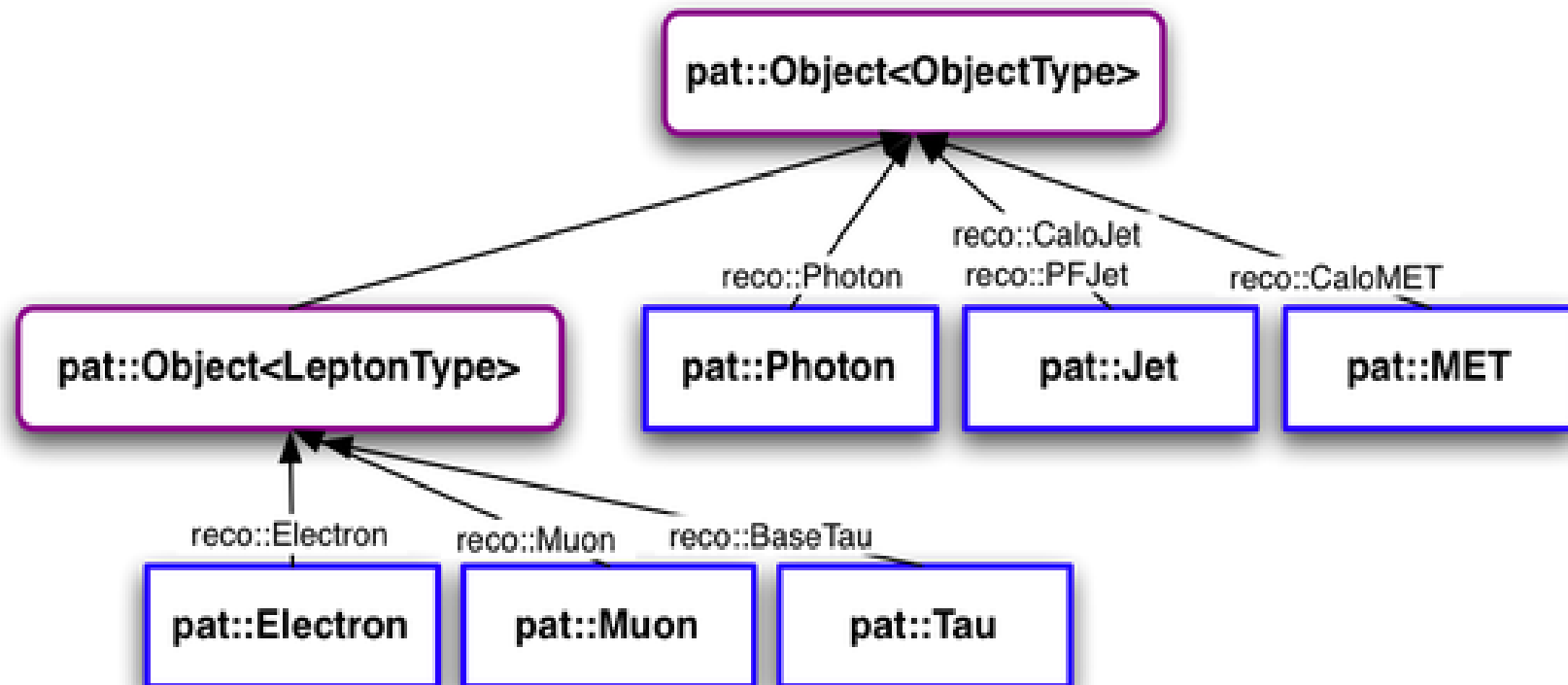
# Maximal Configurability

- Sustain flexibility and user friendliness by maximized configurability
- This configurability is four-fold



# Data Formats

- **DataFormats** in **PatCandidates**



- All **pat::Objects** inherit from their corresponding **reco::Candidates**
- A **pat::Object** is a **reco::Object** (plus more)

# Data Formats

- Find further documentation on **SWGuidePATDataFormats**

## PAT: DataFormats

### Contents:

- ↓ [Trigger Data Formats](#)
- ↓ [Main Data Formats](#)
  - ↓ [pat::Photon](#)
  - ↓ [pat::Electron](#)
  - ↓ [pat::Muon](#)
  - ↓ [pat::Tau](#)
  - ↓ [pat::Jet](#)
  - ↓ [pat::MET](#)
- ↓ [Helper Classes](#)
  - ↓ [pat::JetCorrFactors](#)
  - ↓ [pat::CandKinResolution](#)

### Access:

[Hide](#) ▾

The labels of all jet collections, which are created during the standard PAT workflow(s) are given below:

| Data Type        | Module Label       | Instance Label | Process Label |
|------------------|--------------------|----------------|---------------|
| patJetCollection | allLayer1Jets      | None           | PAT           |
| patJetCollection | selectedLayer1Jets | None           | PAT           |
| patJetCollection | cleanedLayer1Jets  | None           | PAT           |

For a description of each listed module have a look at [SWGuidePATConfiguration](#).

### Member Functions:

You can find a list of all member functions at [pat::Jet](#)

[Main Page](#) | [Modules](#) | [Namespaces](#) | **[Classes](#)** | [CVS Directory](#) | [Package Documentation](#) | [WorkBook](#) | [Offline Guide](#)

[Alphabetical List](#) | [Class List](#) | [Class Hierarchy](#) | [Class Members](#)

[pat::Jet](#)

# How to access pat::Objects

- For an example with an **EDAnalyzer** have a look at:  
**SWGuidePATExamplesBasic**
- Main Configuration File:

```
import FWCore.ParameterSet.Config as cms
```

```
process = cms.Process("Test")
```

```
process.source = cms.Source("PoolSource",
    fileName = cms.untracked.vstring(
        '/store/user/rwolf/ttbar/patTuple_PATv2_ttbar_tauola_1.root'
    )
)
```

Load some data file

```
process.MessageLogger = cms.Service("MessageLogger")
```

```
process.analyzeBasicPat = cms.EDFilter("PatBasicAnalyzer",
    photonSrc = cms.untracked.InputTag("selectedLayer1Photons"),
    electronSrc = cms.untracked.InputTag("selectedLayer1Electrons"),
    muonSrc = cms.untracked.InputTag("selectedLayer1Muons"),
    tauSrc = cms.untracked.InputTag("selectedLayer1Taus"),
    jetSrc = cms.untracked.InputTag("selectedLayer1Jets"),
    metSrc = cms.untracked.InputTag("layer1METs")
)
```

Load the most important  
**pat::Candidate** collections  
into the analyzer

```
process.TFileService = cms.Service("TFileService",
    fileName = cms.string('analyzePatBasics.root')
)
```

```
process.p = cms.Path(process.analyzeBasicPat)
```

# How to access pat::Objects

- PatBasicAnalyzer Module:

```
#include "DataFormats/PatCandidates/interface/Electron.h"
#include "DataFormats/PatCandidates/interface/Photon.h"
#include "DataFormats/PatCandidates/interface/Muon.h"
#include "DataFormats/PatCandidates/interface/Tau.h"
#include "DataFormats/PatCandidates/interface/Jet.h"
#include "DataFormats/PatCandidates/interface/MET.h"
```

Make the data formats known to the module

```
PatBasicAnalyzer::PatBasicAnalyzer(const edm::ParameterSet& iConfig):
    histContainer_()
    photonSrc_(iConfig.getUntrackedParameter<edm::InputTag>("photonSrc")),
    elecSrc_(iConfig.getUntrackedParameter<edm::InputTag>("electronSrc")),
    muonSrc_(iConfig.getUntrackedParameter<edm::InputTag>("muonSrc")),
    tauSrc_(iConfig.getUntrackedParameter<edm::InputTag>("tauSrc")),
    jetSrc_(iConfig.getUntrackedParameter<edm::InputTag>("jetSrc")),
    metSrc_(iConfig.getUntrackedParameter<edm::InputTag>("metSrc"))
{
}
```

Read in the InputTags

```
// get jet collection
edm::Handle<edm::View<pat::Jet> > jets;
iEvent.getByLabel(jetSrc_, jets);
```

Receive the jet collection by label

```
// loop over jets
size_t nJets=0;
for(edm::View<pat::Jet>::const_iterator jet=jets->begin(); jet!=jets->end(); ++jet){
    if(jet->pt()>50)
        ++nJets;
}
```

Loop the jet collection

# How to access pat::Objects

- For an examples with a FWLiteAnalyzer have a look at:

## SWGuidePATExamplesFWLite

```
// load framework libraries
gSystem->Load( "libFWCoreFWLite" );
AutoLibraryLoader::enable();
```

Never forget to enable the AutoLibraryLoader

```
// loop the events
unsigned int iEvent=0;
fwlite::Event event(inFile);
for(event.toBegin(); !event.atEnd(); ++event, ++iEvent){
    // break loop after end of file is reached
    // or after 1000 events have been processed
    if( iEvent==1000 ) break;

    // simple event counter
    if(iEvent>0 && iEvent%100==0){
        std::cout << " processing event: " << iEvent << std::endl;
    }
}
```

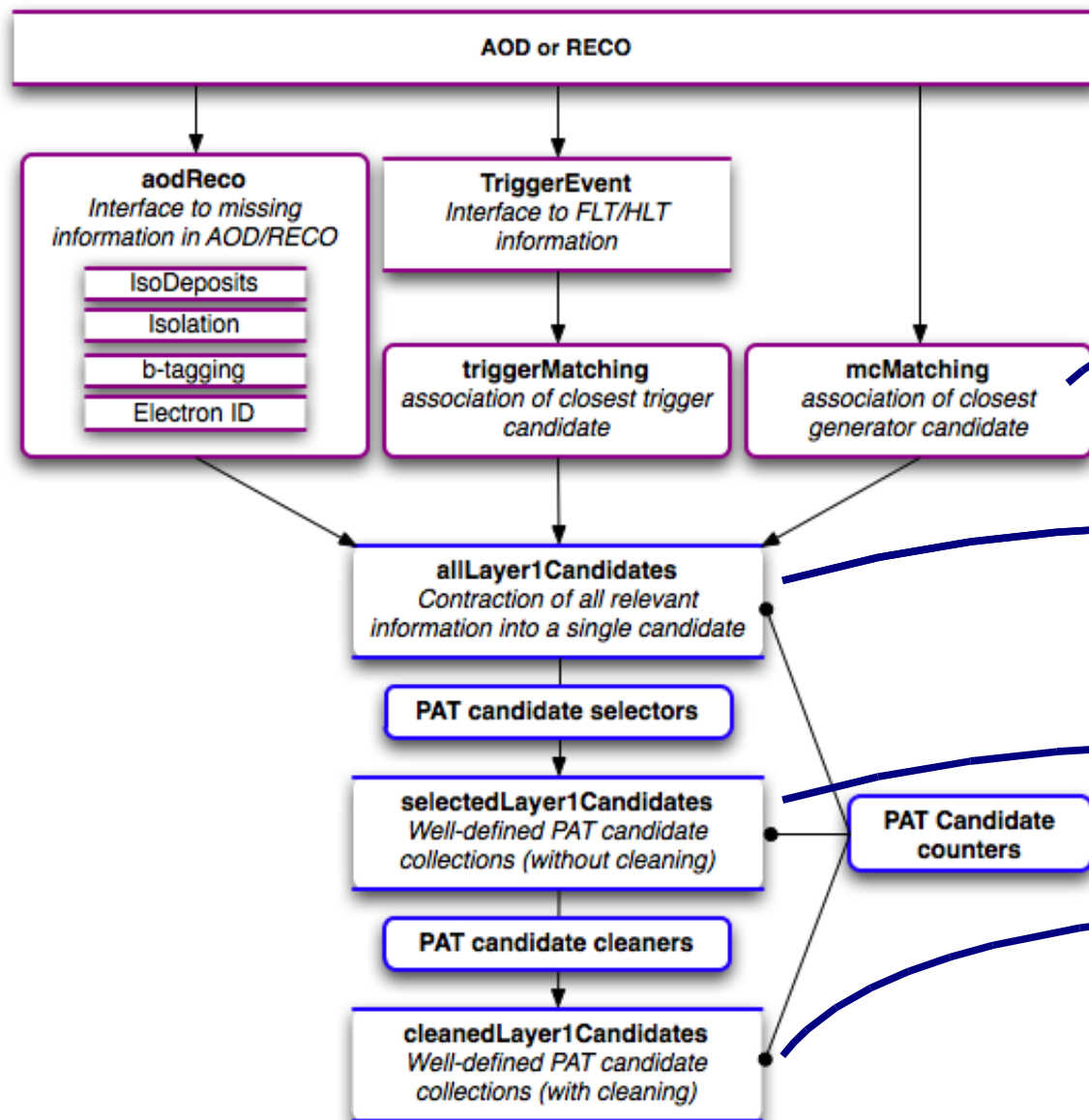
Loop the events of an input file

```
// fwlite::Handle to to muon collection
fwlite::Handle<std::vector<pat::Muon> > muons;
muons.getByLabel(event, "selectedLayer1Muons");

// loop muon collection and fill histograms
for(unsigned i=0; i<muons->size(); ++i){
    muonPt_ ->Fill( (*muons)[i].pt() );
    muonEta_ ->Fill( (*muons)[i].eta() );
    muonPhi_ ->Fill( (*muons)[i].phi() );
}
}
// close input file
inFile->Close();
```

Receive the jet collection by label

# Workflow



- Have a look at:  
**SWGuidePATWorkflow**

Pre-Production steps

Basic input collection

Main collection (w/o cleaning)

Main collection (with cleaning)

- Resembled by the structure of the python directory

# AOD/Reco Steps

- **Pre-pat::Candidate** Production steps on RECO/AOD input
- It happens that 'old'-CMSSW version input files lack finest newest, objects/algorithms
  - BTag algorithms
  - Tau discriminators
  - Electron isolation from RecHits
  - Electron Id, ...

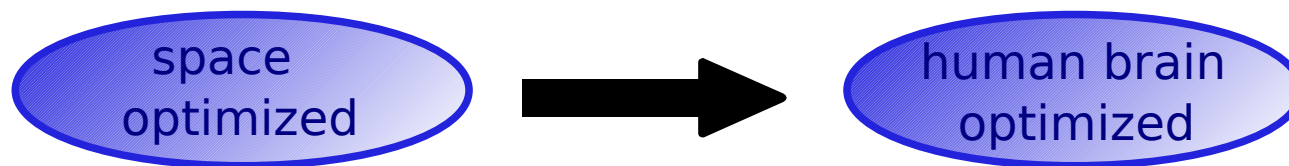
```
# Sequences needed to deliver external information for objects
# You can remove modules from here if you don't need these features
patAODExtraReco = cms.Sequence(
    #patBTagging +           # Empty sequences not supported yet
    patElectronId +
    patElectronIsolation +
    patJetMETCorrections +
    patJetTracksCharge +
    #patMuonIsolation +     # Empty sequences not supported yet
    #patPhotonID +         # Empty sequences not supported yet
    patPhotonIsolation +
    #patTauDiscrimination + # Empty sequences not supported yet
    patPFCandidateIsoDepositSelection +
    patPFTauIsolation
)
```

This sequence fills up this gap



# AOD/Reco Steps

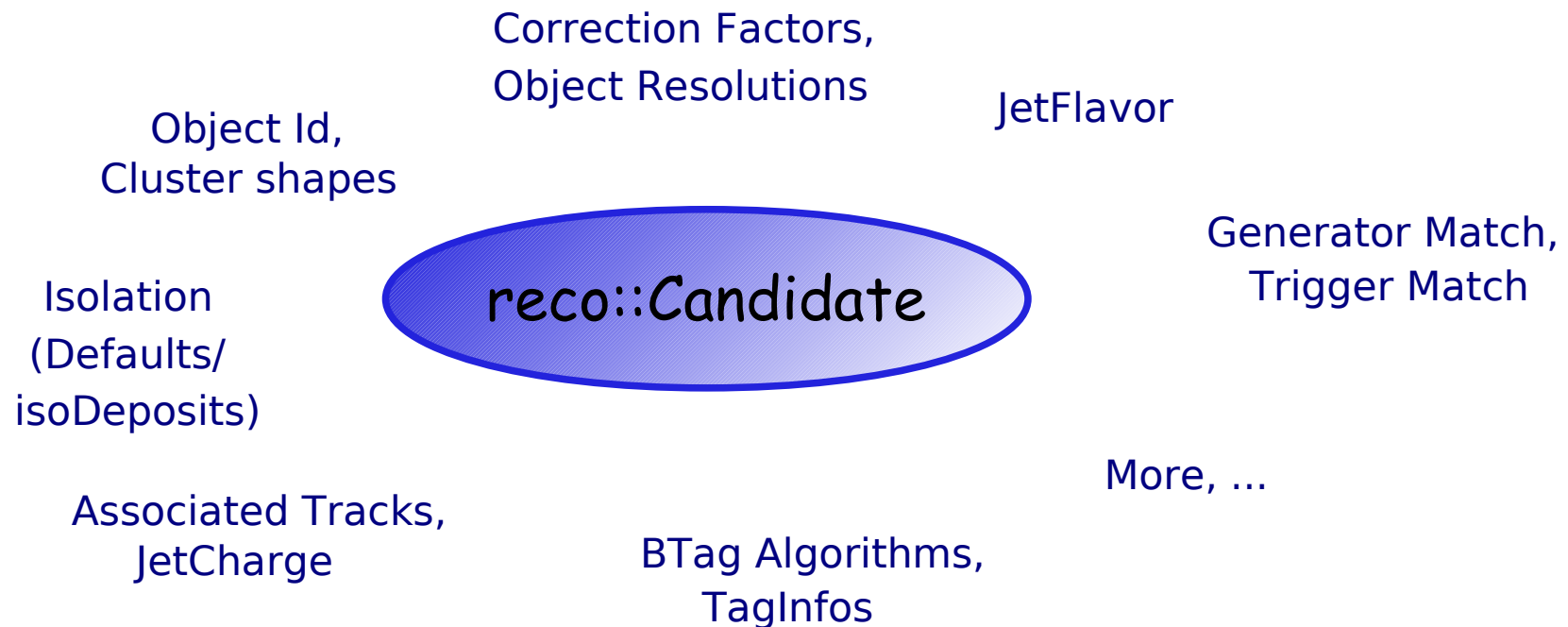
- Other pre-production Steps are missing by construction
- Re-organization of the **TriggerEvent** content:



- Receiving fines/best corrections for Jets & MET:
  - Fill structures, which can be folded into the **pat::Candidate** persistently
  - Adding of more than one set of jet corrections is possible
- Candidate Matching to Trigger/Generator Objects:
  - Search for match within radius in  $\Delta R$  (closest in  $\Delta R$ , OR highest in  $p_T$ )
- **NOTE:** interface to POGs/PTs **NO extra PAT inventions!!!**

# pat::Candidate Creation

- All this extra information is folded into the **pat::Candidates**



- Each **pat::Candidate** is a **reco::Candidate+more**

# pat::Candidate Creation

- Information can be made persistent (embedded) or kept as reference
  - All persistent data is easily accessible in FWLite
  - Flexible size management of a single **pat::Candidate** class
  - Fully transparent for the user
- The size of the **pat::Candidate** depends on the choice of the user

**RECO**

~500kb/evt

**AOD**

~100kb/evt

**PAT**

~6-60kb/evt

- Results in **allLayer1Candidates** (basic candidate collection of PAT)

# pat::Candidate Creation

- The current default size should pretty much look like the old PAT tuple, of the Summer08 production:

maximally  
36kB/evt

| Collection                                      | items/event | kb/event | kb/item | plot % |
|-------------------------------------------------|-------------|----------|---------|--------|
| recoGenParticles_genParticles__HLT              | 766.55      | 16.18    | 0.02    | 29.3%  |
| CaloTowersSorted_towerMaker__RECO               | 440.02      | 13.74    | 0.03    | 24.9%  |
| recoTracks_generalTracks__RECO                  | 105.41      | 10.46    | 0.10    | 19.0%  |
| patJets_selectedLayer1Jets__TEST                | 6.38        | 6.05     | 0.95    | 11.0%  |
| patElectrons_selectedLayer1Electrons__TEST      | 1.26        | 3.09     | 2.45    | 5.6%   |
| patPhotons_selectedLayer1Photons__TEST          | 2.80        | 2.97     | 1.06    | 5.4%   |
| patMuons_selectedLayer1Muons__TEST              | 1.41        | 1.60     | 1.13    | 2.9%   |
| recoVertexs_offlinePrimaryVertices__RECO        | 1.07        | 0.70     | 0.66    | 1.3%   |
| patMETs_selectedLayer1METs__TEST                | 1.00        | 0.22     | 0.22    | 0.4%   |
| patTaus_selectedLayer1Taus__TEST                | 0.38        | 0.07     | 0.18    | 0.1%   |
| pathHemispheres_selectedLayer1Hemispheres__TEST | 2.00        | 0.06     | 0.03    | 0.1%   |
| recoPdfInfo_genEventPdfInfo__HLT                | 1.00        | 0.02     | 0.02    | 0.0%   |
| recoBeamSpot_offlineBeamSpot__RECO              | 1.00        | 0.01     | 0.01    | 0.0%   |
| triggerTriggerEvent_hltTriggerSummaryAOD__HLT   | 1.00        | 0.00     | 0.00    | 0.0%   |
| int_genEventProcID__TEST                        | 1.00        | 0.00     | 0.00    | 0.0%   |
| ints_genParticles__HLT                          | 1.00        | 0.00     | 0.00    | 0.0%   |
| double_genEventScale__HLT                       | 1.00        | 0.00     | 0.00    | 0.0%   |
| edmTriggerResults_TriggerResults__HLT           | 1.00        | 0.00     | 0.00    | 0.0%   |
| double_genEventWeight__HLT                      | 1.00        | 0.00     | 0.00    | 0.0%   |
| EventMetaData + EventHistory                    | 1.00        | 0.11     | 0.11    | 0.2%   |

- For the default configuration have a look at **SWGuidePATConfiguration**

# selectedLayer1Candidates

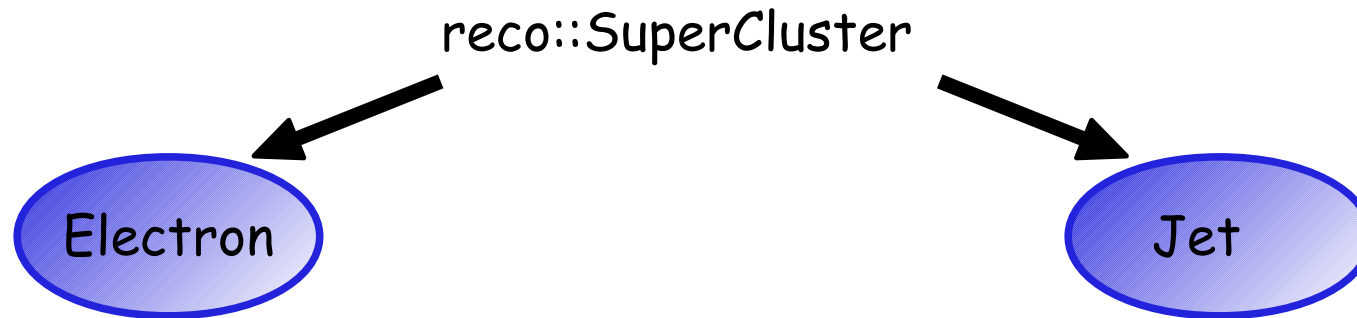
- Simple selection via string parser:

```
# module to select Muons
# See https://twiki.cern.ch/twiki/bin/view/CMS/SWGuidePhysicsCutParser
# on how to use the cut-string
#
# These are only dummy cuts
selectedLayer1Muons = cms.EDFilter("PATMuonSelector",
    src = cms.InputTag("allLayer1Muons"),
    cut = cms.string('pt > 0. & abs(eta) < 12.')
)
```

- Can use all kind of functions and all kind of member functions of the **pat::Candidate**
- For more details have a look at **SWGuidePhysicsCutParser**
- Results in **selectedLayer1Candidates** (std candidate collection of pat::Candidates w/o object cleaning)
- For the choice of standard selection strings have a look at **SWGuidePATConfiguration**

# cleanLayer1Candidates

- What is object (cross) cleaning?



- A set of **PATCandidateCleaners** can help to resolve this double counting if desired
- Results in **cleanLayer1Candidates** (std candidate collection of `pat::Candidates` with object cleaning)
- For the standard configuration of these cleaners have a look at **SWGuidePATConfiguration**

# EventContent

- Compose the `pat::EventContent` to your will:

```
patEventContentNoLayer1Cleaning = [
    'keep *_selectedLayer1Photons_*_*',
    'keep *_selectedLayer1Electrons_*_*',
    'keep *_selectedLayer1Muons_*_*',
    'keep *_selectedLayer1Taus_*_*',
    'keep *_selectedLayer1Jets_*_*',
    'keep *_layer1METs_*_*',
    'keep *_selectedLayer1PFPParticles_*_*'
]
```

```
patEventContent = [
    'keep *_cleanLayer1Photons_*_*',
    'keep *_cleanLayer1Electrons_*_*',
    'keep *_cleanLayer1Muons_*_*',
    'keep *_cleanLayer1Taus_*_*',
    'keep *_cleanLayer1Jets_*_*',
    'keep *_layer1METs_*_*',
    'keep *_cleanLayer1Hemispheres_*_*',
    'keep *_cleanLayer1PFPParticles_*_*'
]
```

```
patExtraAodEventContent = [
    # GEN
    'keep recoGenParticles_genParticles_*_*',
    'keep *_genEventScale_*_*',
    'keep *_genEventWeight_*_*',
    'keep *_genEventPdfInfo_*_*',
    # RECO
    'keep recoTracks_generalTracks_*_*',
    'keep *_towerMaker_*_*',
    'keep *_offlineBeamSpot_*_*',
    'keep *_offlinePrimaryVertices_*_*',
    # TRIGGER
    'keep edmTriggerResults_TriggerResults_*_HLT',
    'keep *_hltTriggerSummaryAOD_*_*'
]
```

plus more...

- Add EventContent to the output module:

```
# Output module configuration
from PhysicsTools.PatAlgos.patEventContent_cff import patEventContent
process.out = cms.OutputModule("PoolOutputModule",
    fileName = cms.untracked.string('PATLayer1_Output.fromAOD_full.root'),
    # save only events passing the full path
    SelectEvents = cms.untracked.PSet( SelectEvents = cms.vstring('p') ),
    # save PAT Layer 1 output
    outputCommands = cms.untracked.vstring('drop *', *patEventContent ) # you
)
process.outpath = cms.EndPath(process.out)
```

# EventContent

- Tools will help you estimate the size of your `pat::Tuple`:

## **patMuons\_selectedLayer1Muons\_\_TEST (1.4 items/event)**

| Datamember              | kb/event | kb/item | plot % | compressed |
|-------------------------|----------|---------|--------|------------|
| isoDeposits_            | 0.796    | 0.563   | 49.8%  | ok         |
| muMatches_              | 0.370    | 0.262   | 23.1%  | ok         |
| combinedMuon_           | 0.091    | 0.064   | 5.7%   | ok         |
| standAloneMuon_         | 0.088    | 0.062   | 5.5%   | ok         |
| isolations_             | 0.019    | 0.014   | 1.2%   | ok         |
| triggerMatches_         | 0.011    | 0.008   | 0.7%   | ok         |
| genParticleRef_         | 0.007    | 0.005   | 0.4%   | ok         |
| vertex_.fCoordinates.fZ | 0.007    | 0.005   | 0.4%   | ok         |
| vertex_.fCoordinates.fY | 0.007    | 0.005   | 0.4%   | ok         |
| phi_                    | 0.006    | 0.005   | 0.4%   | ok         |
| eta_                    | 0.006    | 0.005   | 0.4%   | ok         |
| vertex_.fCoordinates.fX | 0.006    | 0.005   | 0.4%   | ok         |

For more details have a look at **SWGuidePATEventSize**

- Embedding object information will help you to drop branches



# Workflow Tools

- Tools will help you to configure the pat::Workflow:

- addJetCollections:** (patLayer1\_fromAOD\_jetSuite\_full)

```
addJetCollection(process,cms.InputTag('JetPlusTrackZSPCorJetICone5'),
    'JPTc',
    doJTA      = True,
    doBTagging = True,
    jetCorrLabel = ('IC5', 'JPT'),
    doType1MET = True,
    doL1Cleaning = True,
    doL1Counters = True,
    genJetCollection = cms.InputTag("iterativeCone5GenJets")
)
```

- switchJetCollections:** (patLayer1\_fromAOD\_sisCone\_full)

```
switchJetCollection(process,
    cms.InputTag('sisCone5CaloJets'), # Jet collection
    doJTA=True,                       # Run Jet-Track ,
    doBTagging=True,                  # Run b-tagging
    jetCorrLabel=('SC5', 'Calo'),     # example jet co
    doType1MET=True,                  # recompute Type
    genJetCollection=cms.InputTag("sisCone5GenJets"))
```

Choose between supported jet algo's, pflow, JPT, calo, user defined

- More Tools: switch  $\tau$  collections, tcMET, trigger configuration, ...

# Support

- For more information on support have a look at **SWGuidePAT**:

## Support

In this section you can find the links to a all kind of support, which you might want to make use of. The **Starting Point** for any question or request might be the [Physics Tools HN](#). In the first place more people than you might have the same question as you and may profit from the public answer. Moreover people might have had a similar question already before and a query of the list might already be of help.

### PAT core developers:

Find a list of the most important developers below:

[Show](#) ▶

### POG contacts:

Find a list of POG contact persons below:

[Show](#) ▶

### PAG contacts:

Find a list of PAG contact persons below:

[Show](#) ▶

- Lecturers & Tutors
- Hypernews
- Community
- POG/PAG contacts
- Developers



# Documentation

- **SWGuidePAT** Main documentation page
- **SWGuidePATRecipes** Installation recipes
- **SWGuidePATExamples** Tutorials and examples to get started
- **SWGuidePATDataFormats** pat::Candidate description
- **SWGuidePATConfiguration** Module configuration
- **SWGuidePATEventSize** Tools for event size estimate
- **SWGuidePATWorkflow** PAT workflow description
- **SWGuidePATTools** Description of workflow configuration tools
- And last but not least: This Tutorial...

# Future

- End of June the switch to **CMSSW\_3\_1\_0** foreseen
  - Still one change in collection names ('layer1' is obsolete)
  - Improved configuration file structure (more flexible standard re-configuration tools/cff's)
  - Less tags (better integration into the following releases)
- **ConfigBrowser**
  - Process configuration via click and menu
  - Less error prone / much improved user friendliness
- Change of paradigm
  - From 'experimental' and 'task force' philosophy to fully established tool with many users
  - Increased and well defined connection with PTs/PAGs/POGs (**service credits**)
  - **Help in development is always welcome/needed!!!**
  - Internal note for common description and reference

# Exercises

(1) Go to **SWGuidePAT** and install newest PAT tags with CMSSW\_2\_2\_13

(2) Go through beginner's tutorial at **WorkBookPAT**

- Run standard configuration & check event content with **edmDumpEventContent**
- Switch from iterativeCone5 to sisCone5 jets
- Add JPT jets

**Lunch Break!!!**

(3) Run an **EDAnalyzer** on PAT/RECO, following **WorkBookExamplePATBasic**

(4) Run **FWLiteAnalyzer** on PAT, following **WorkBookPATExampleFWLite**