Gaudi and FCCSW

Tutorials for Snowmass 2021

Sep 22, 2020
CERN-EP

Largely based on J Faltova presentation at Oct 2019 FCCSW workshop

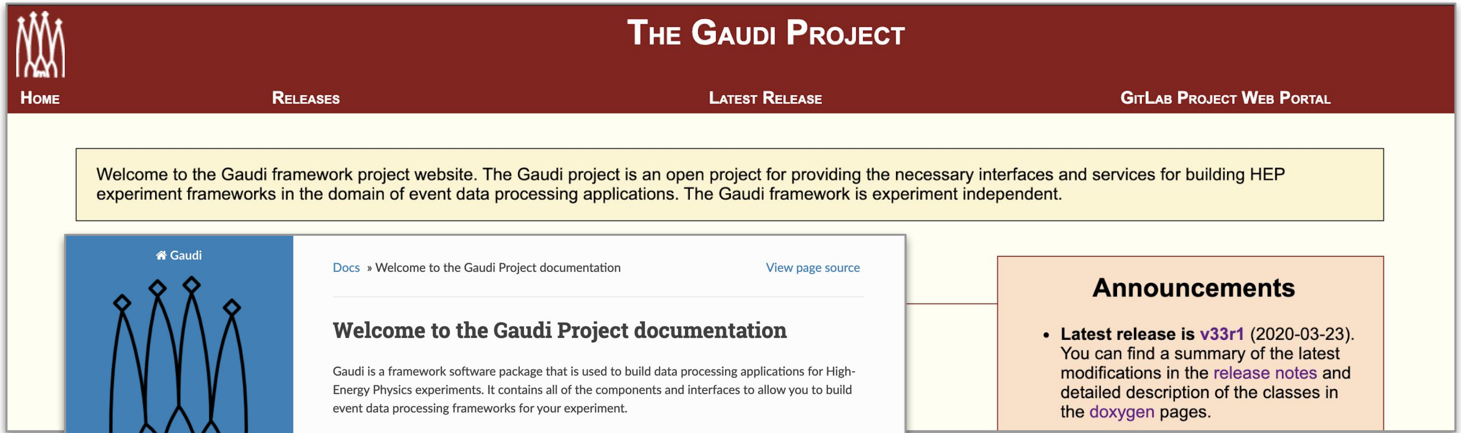
Gaudi

- The Gaudi project

  "Open project for providing the necessary interfaces and services for building HEP experiment frameworks in the domain of event data processing applications. The Gaudi framework is experiment independent"
- Originating from LHCb. Used by
  - ATLAS, Daya Bay, GLAST (Fermi Gamma-ray Space Telescope)
  - FCC, Key4HEP
- Data processing framework designed to manage experiment workflows
  - Separate data and algorithms; well defined interfaces
  - User's code encapsulated in Algorithm's, Tool's / Interface's, Service's
  - Different persistent and transient views of data
- C++, with Python configuration

Gaudi links / doc

- http://gaudi.web.cern.ch/gaudi/



THE GAUDI PROJECT

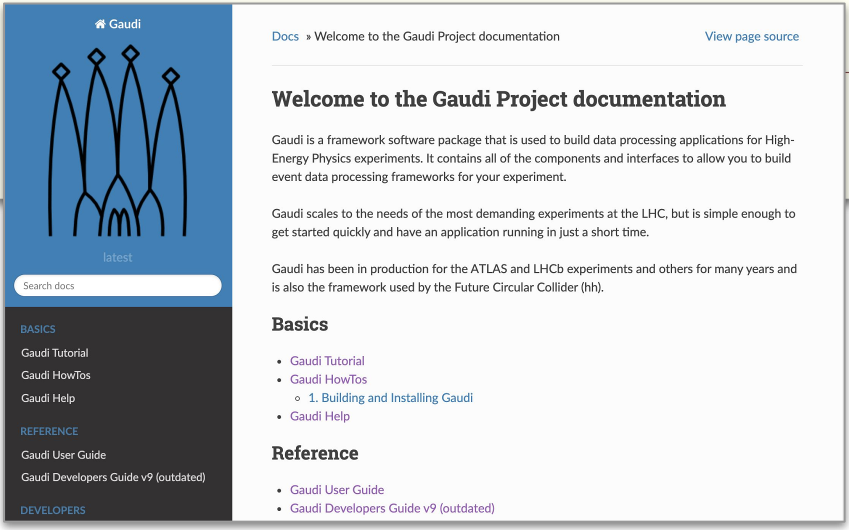
HOME RELEASES LATEST RELEASE GITLAB PROJECT WEB PORTAL

Welcome to the Gaudi framework project website. The Gaudi project is an open project for providing the necessary interfaces and services for building HEP experiment frameworks in the domain of event data processing applications. The Gaudi framework is experiment independent.

Announcements

- **Latest release is v33r1** (2020-03-23). You can find a summary of the latest modifications in the release notes and detailed description of the classes in the doxygen pages.

- ReadTheDocs



Gaudi

latest

Search docs

BASICS
- Gaudi Tutorial
- Gaudi HowTos
- Gaudi Help

REFERENCE
- Gaudi User Guide
- Gaudi Developers Guide v9 (outdated)

DEVELOPERS

Docs » Welcome to the Gaudi Project documentation View page source

Welcome to the Gaudi Project documentation

Gaudi is a framework software package that is used to build data processing applications for High-Energy Physics experiments. It contains all of the components and interfaces to allow you to build event data processing frameworks for your experiment.

Gaudi scales to the needs of the most demanding experiments at the LHC, but is simple enough to get started quickly and have an application running in just a short time.

Gaudi has been in production for the ATLAS and LHCb experiments and others for many years and is also the framework used by the Future Circular Collider (hh).

Basics

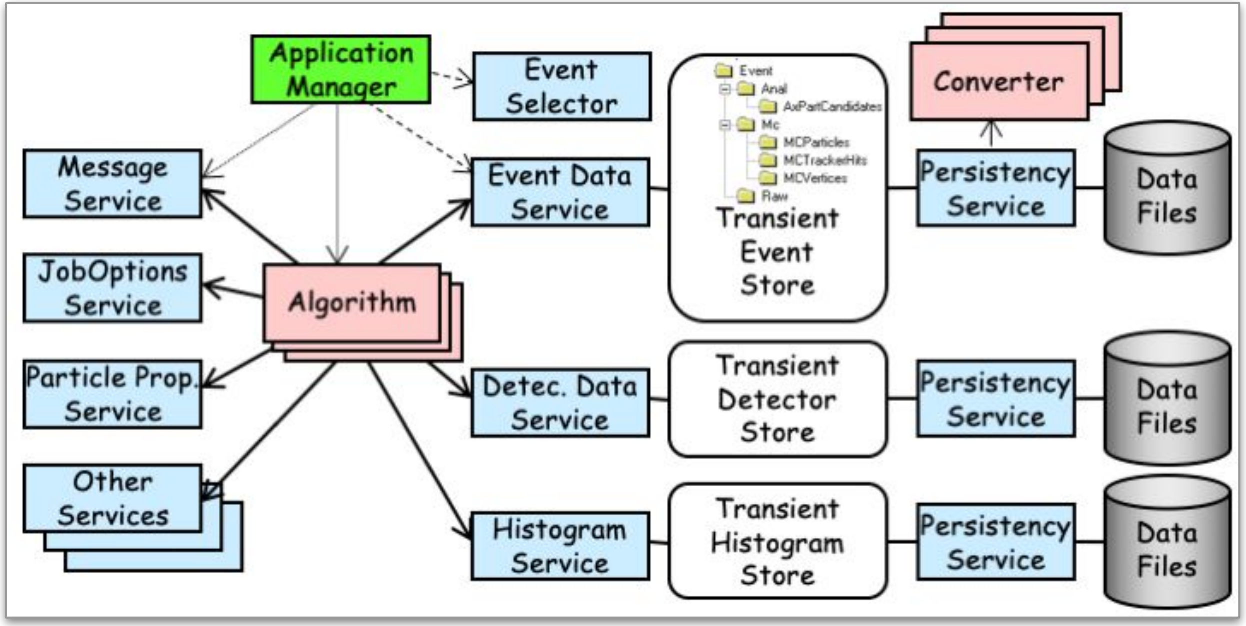
- Gaudi Tutorial
- Gaudi HowTos
  - 1. Building and Installing Gaudi
- Gaudi Help

Reference

- Gaudi User Guide
- Gaudi Developers Guide v9 (outdated)

Gaudi Architecture

- Goal: insulate physicists from software specific details such as low level libraries, I/O, graphics, ...



- Keep it simple of use

Gaudi Components

Gaudi in words

- Application Manager schedules algorithm execution
- Relevant functionality implemented as
  - Algorithms
- Algorithms are written in terms of
  - Tools
  - Interfaces
- Everything makes use of
  - Services

Algorithm

- Function
  - Take input data, manipulate it and produce new output data
- Main features
  - Configurable
    - Initialize() – called once at the start of the job
    - Execute() – called once per event
    - Finalize() – called once at the end of the job
  - Uses Services and Tools

Tools

- Function
  - Smaller pieces of code doing one particular thing (e.g. energy calibration, emulation of electronic noise)
- Main features
  - Can be called many times per event
  - Configurable
  - Private or public
    - Private
      - owned / accessible only by the component creating it
    - Public
      - owned by the framework and accessible globally

Interfaces

- Description
  - Abstract class where all the methods are pure virtual
- Used for
  - Different tools can implement the same functionality (e.g. vertex creation)
  - Factor out common properties or methods of different tools
    - Implement common base class as **interface**
  - The choice of which tool to use can be done via the job options at run time

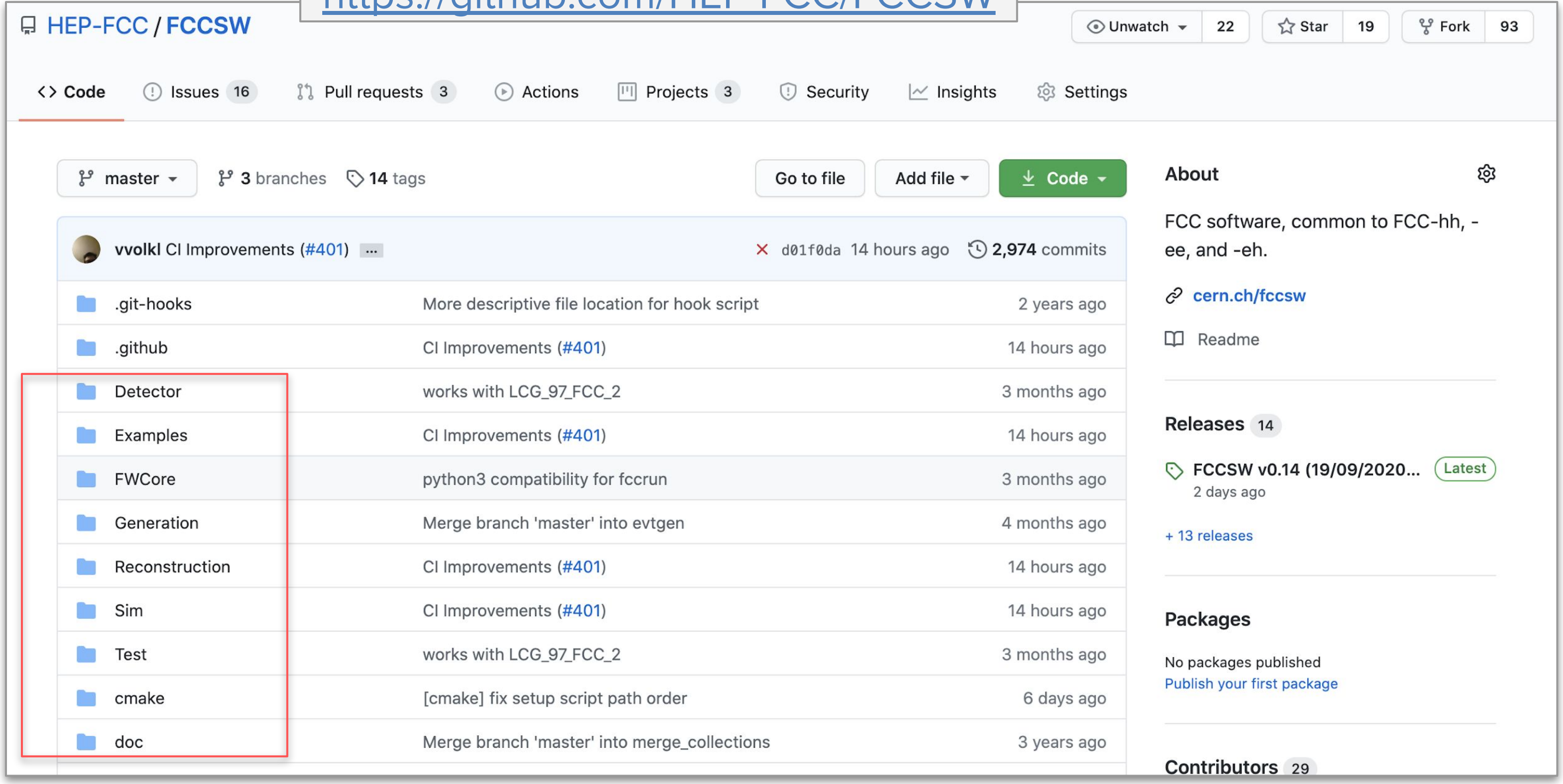
Services

- Description
  - Support a given functionality of general interest
    - E.g. Job Options Svc, Message Svc, Event Data Svc, Histogram Svc, Ntuple Svc, Detector Data Svc
- Created by the framework to provide global functionality
  - Users do not need to care

Code structure of FCCSW

FCCSW github landing page

https://github.com/HEP-FCC/FCCSW


HEP-FCC / FCCSW

Unwatch 22 | Star 19 | Fork 93

Code | Issues 16 | Pull requests 3 | Actions | Projects 3 | Security | Insights | Settings

master | 3 branches | 14 tags | Go to file | Add file | Code

vvolkl CI Improvements (#401) ... d01f0da 14 hours ago 2,974 commits

.git-hooks	More descriptive file location for hook script	2 years ago
.github	CI Improvements (#401)	14 hours ago
Detector	works with LCG_97_FCC_2	3 months ago
Examples	CI Improvements (#401)	14 hours ago
FWCore	python3 compatibility for fccrun	3 months ago
Generation	Merge branch 'master' into evtgen	4 months ago
Reconstruction	CI Improvements (#401)	14 hours ago
Sim	CI Improvements (#401)	14 hours ago
Test	works with LCG_97_FCC_2	3 months ago
cmake	[cmake] fix setup script path order	6 days ago
doc	Merge branch 'master' into merge_collections	3 years ago

About

FCC software, common to FCC-hh, -ee, and -eh.

cern.ch/fccsw

Readme

Releases 14

FCCSW v0.14 (19/09/2020... Latest
2 days ago

+ 13 releases

Packages

No packages published
Publish your first package

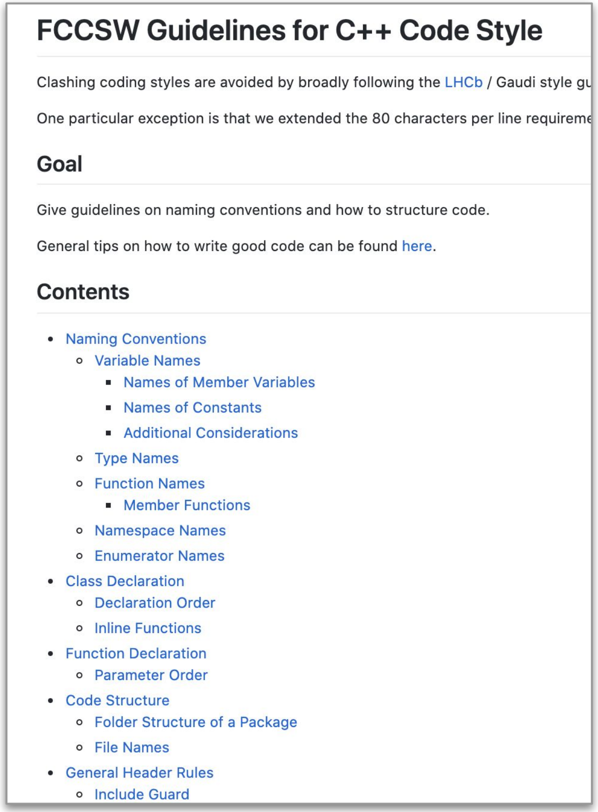
Contributors 29

FCCSW in words

- Core component: FWCore
  - Provides connection with data stores
- Functionality by category
  - Detector, Generation, Sim, Reconstruction
- Each category contains packages
  - E.g. Sim/SimDelphesInterface
  - And category specific documentation (markdown files in folder *doc*)
- Each package has the follow minimal components
  - *src/components*, folder with code files {.h, .cpp}
  - *options*, folder with job options files {.py}
  - *CMakeLists.txt*, file with {build, install, test} instructions

Code style guidelines for FCCSW

- Code style guidelines
- Variable names
  - Meaningful names
  - Members variables start with m_, constant with k
- Function names
  - Functions names begin with lower case; capital letter for each new word
- Rules for class, functions declaration, header files
- Automatic check



FCCSW Guidelines for C++ Code Style

Clashing coding styles are avoided by broadly following the LHCb / Gaudi style gu

One particular exception is that we extended the 80 characters per line requireme

Goal

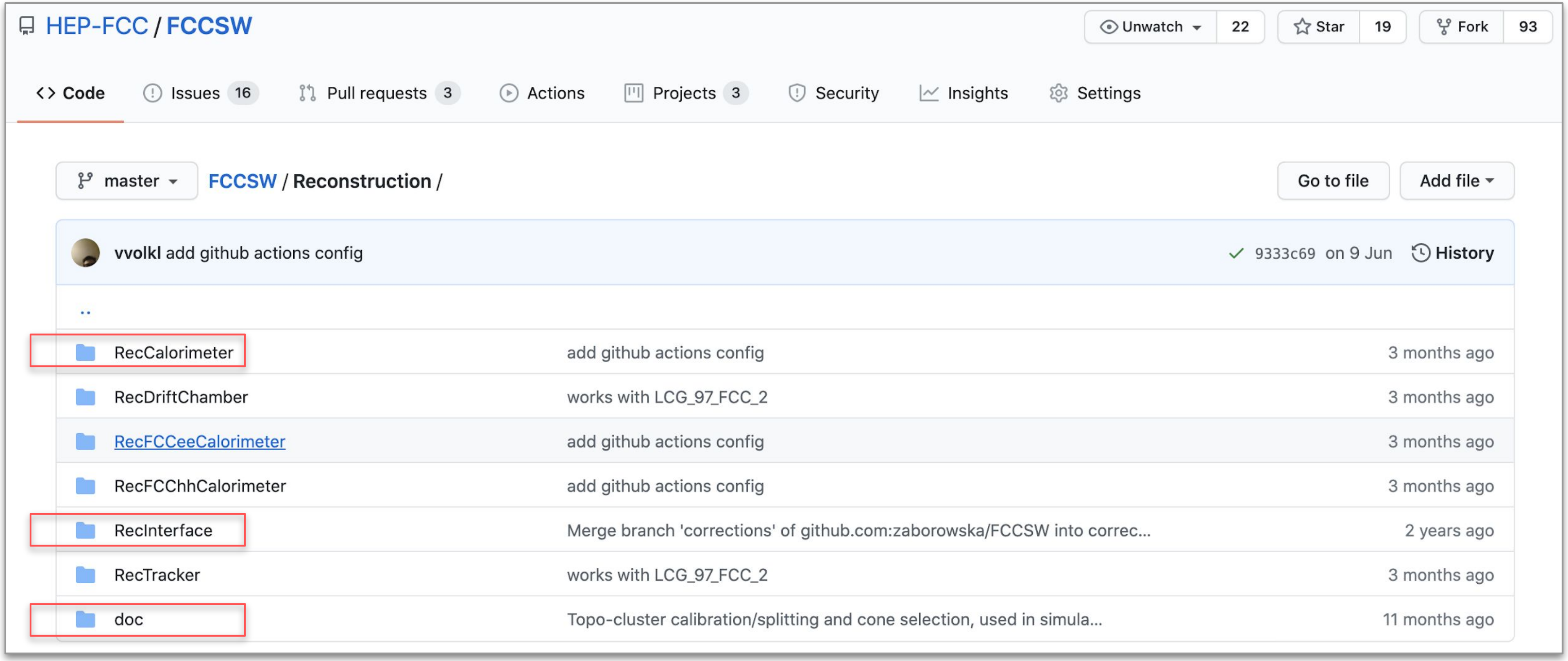
Give guidelines on naming conventions and how to structure code.

General tips on how to write good code can be found here.

Contents

Example: Reconstruction

Example: Reconstruction


HEP-FCC / FCCSW

Unwatch 22 | Star 19 | Fork 93

Code | Issues 16 | Pull requests 3 | Actions | Projects 3 | Security | Insights | Settings

master | FCCSW / Reconstruction / | Go to file | Add file

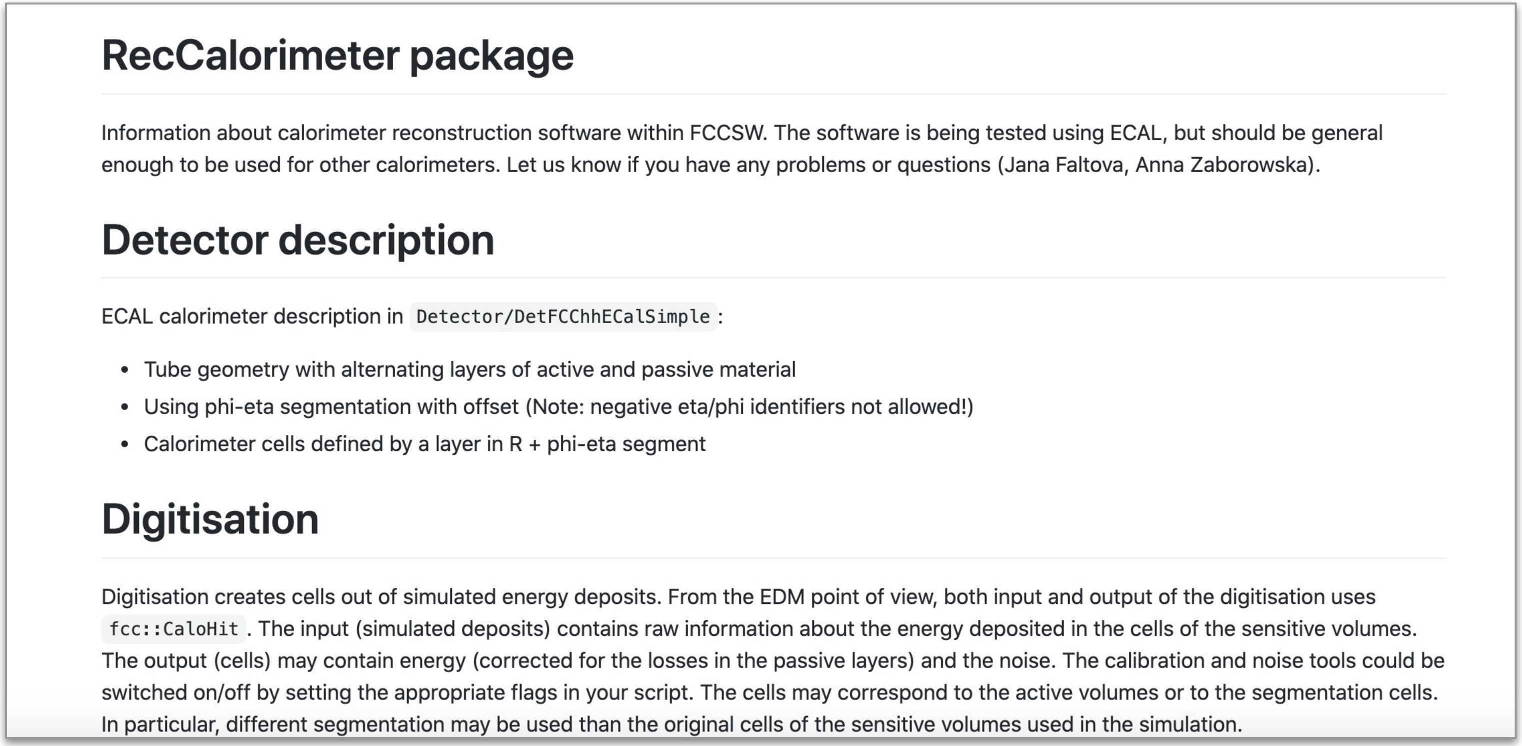
vvolkl add github actions config ✓ 9333c69 on 9 Jun History

..

RecCalorimeter	add github actions config	3 months ago
RecDriftChamber	works with LCG_97_FCC_2	3 months ago
RecFCCeeCalorimeter	add github actions config	3 months ago
RecFCChhCalorimeter	add github actions config	3 months ago
RecInterface	Merge branch 'corrections' of github.com:zaborowska/FCCSW into correc...	2 years ago
RecTracker	works with LCG_97_FCC_2	3 months ago
doc	Topo-cluster calibration/splitting and cone selection, used in simula...	11 months ago

Example: Reconstruction: doc

File: Reconstruction/doc/RecCalorimeter.md



RecCalorimeter package

Information about calorimeter reconstruction software within FCCSW. The software is being tested using ECAL, but should be general enough to be used for other calorimeters. Let us know if you have any problems or questions (Jana Faltova, Anna Zaborowska).

Detector description

ECAL calorimeter description in `Detector/DetFCChhECalSimple` :

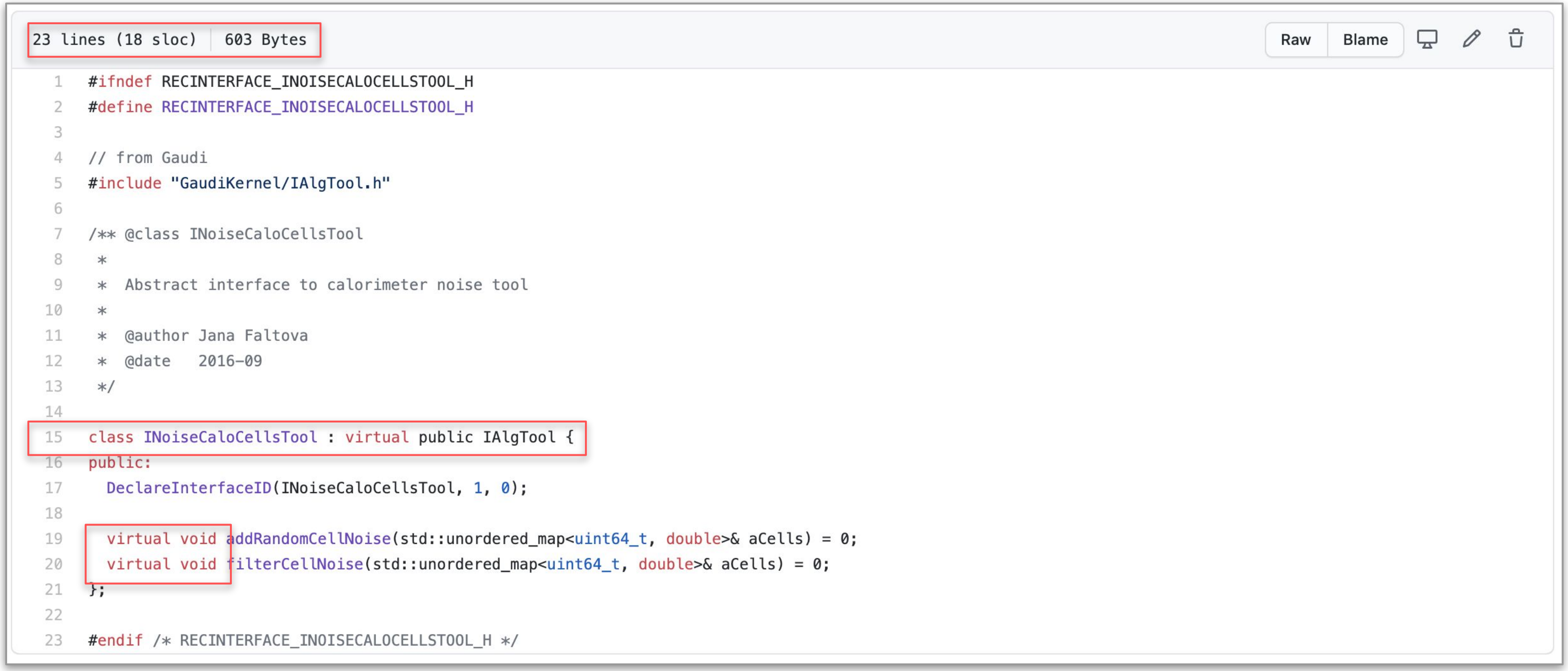
- Tube geometry with alternating layers of active and passive material
- Using phi-eta segmentation with offset (Note: negative eta/phi identifiers not allowed!)
- Calorimeter cells defined by a layer in R + phi-eta segment

Digitisation

Digitisation creates cells out of simulated energy deposits. From the EDM point of view, both input and output of the digitisation uses `fcc::CaloHit` . The input (simulated deposits) contains raw information about the energy deposited in the cells of the sensitive volumes. The output (cells) may contain energy (corrected for the losses in the passive layers) and the noise. The calibration and noise tools could be switched on/off by setting the appropriate flags in your script. The cells may correspond to the active volumes or to the segmentation cells. In particular, different segmentation may be used than the original cells of the sensitive volumes used in the simulation.

Example: Reconstruction: RecInterface

File: Reconstruction/RecInterface/RecInterface/INoiseCaloCellsTool.h

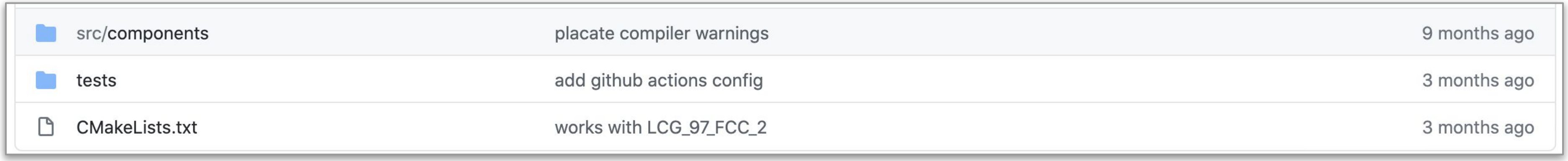

23 lines (18 sloc) | 603 Bytes

Raw Blame

```cpp
#ifndef RECINTERFACE_INOISECALOCELLSTOOL_H
#define RECINTERFACE_INOISECALOCELLSTOOL_H

// from Gaudi
#include "GaudiKernel/IAlgTool.h"

/** @class INoiseCaloCellsTool
 *
 *  Abstract interface to calorimeter noise tool
 *
 *  @author Jana Faltova
 *  @date   2016-09
 */

class INoiseCaloCellsTool : virtual public IAlgTool {
public:
  DeclareInterfaceID(INoiseCaloCellsTool, 1, 0);

  virtual void addRandomCellNoise(std::unordered_map<uint64_t, double>& aCells) = 0;
  virtual void filterCellNoise(std::unordered_map<uint64_t, double>& aCells) = 0;
};

#endif /* RECINTERFACE_INOISECALOCELLSTOOL_H */
```

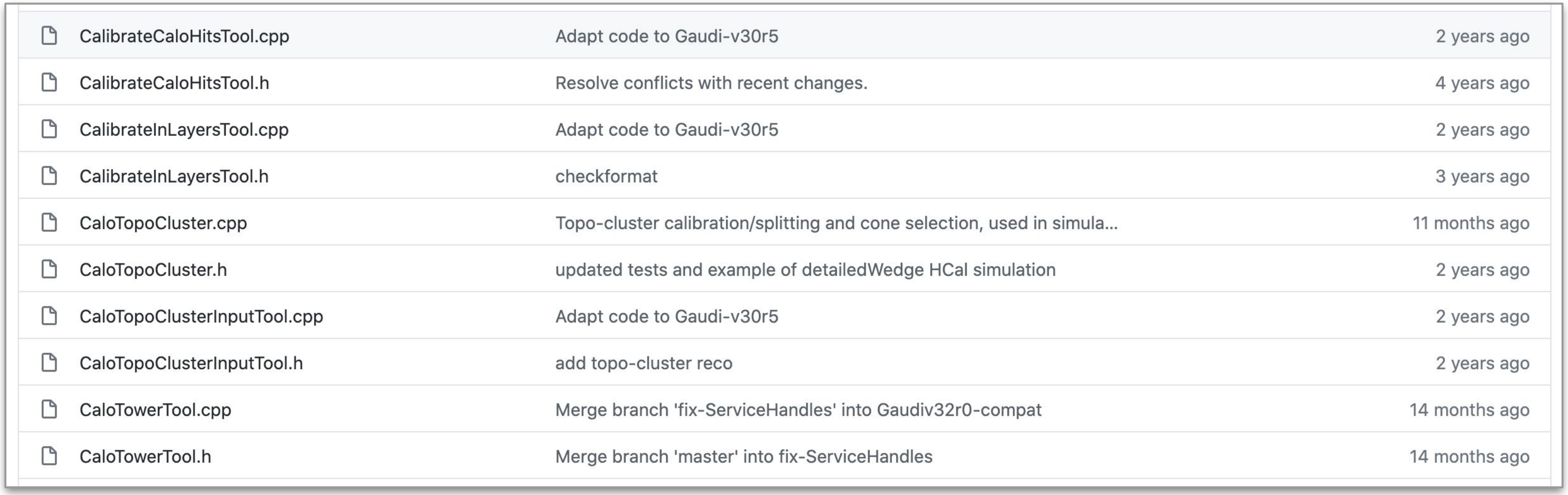
Example: RecCalorimeter

Reconstruction/RecCalorimeter



src/components	placate compiler warnings	9 months ago
tests	add github actions config	3 months ago
CMakeLists.txt	works with LCG_97_FCC_2	3 months ago

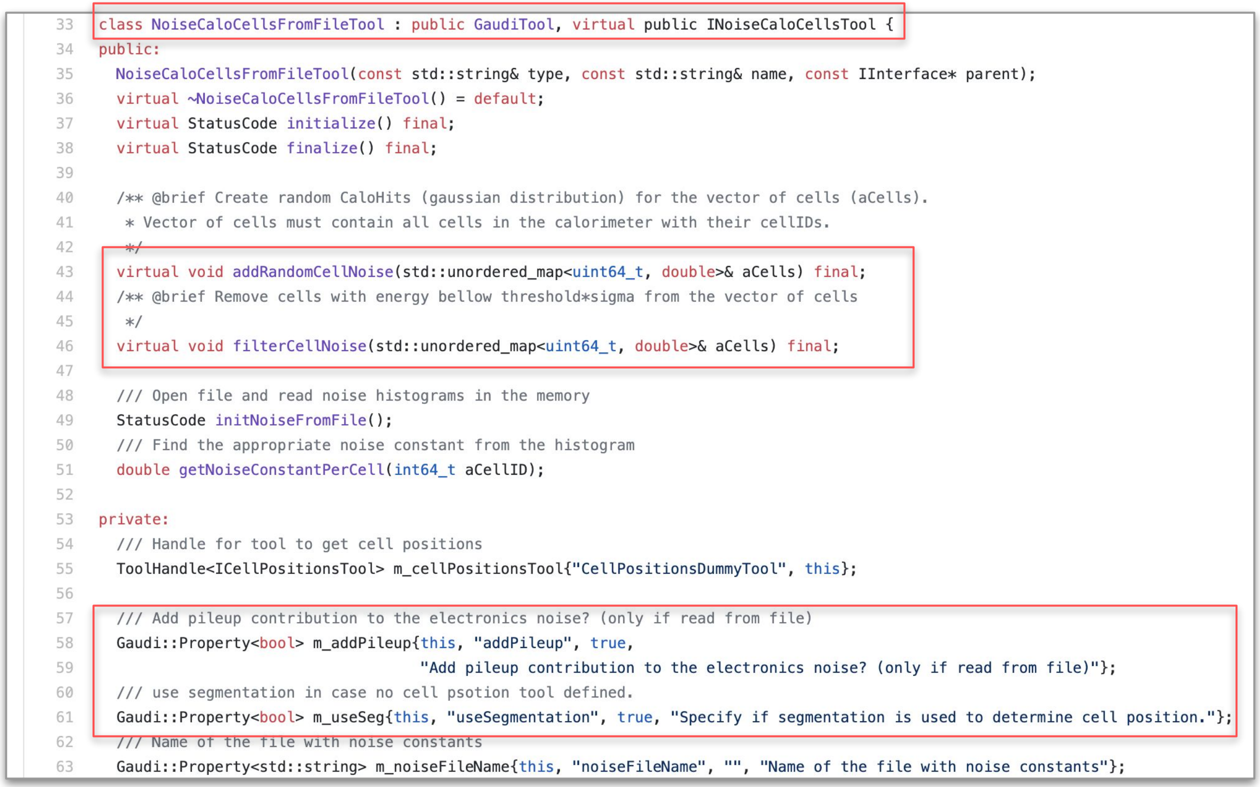
Reconstruction/RecCalorimeter/src/components



CalibrateCaloHitsTool.cpp	Adapt code to Gaudi-v30r5	2 years ago
CalibrateCaloHitsTool.h	Resolve conflicts with recent changes.	4 years ago
CalibrateInLayersTool.cpp	Adapt code to Gaudi-v30r5	2 years ago
CalibrateInLayersTool.h	checkformat	3 years ago
CaloTopoCluster.cpp	Topo-cluster calibration/splitting and cone selection, used in simula...	11 months ago
CaloTopoCluster.h	updated tests and example of detailedWedge HCal simulation	2 years ago
CaloTopoClusterInputTool.cpp	Adapt code to Gaudi-v30r5	2 years ago
CaloTopoClusterInputTool.h	add topo-cluster reco	2 years ago
CaloTowerTool.cpp	Merge branch 'fix-ServiceHandles' into Gaudiv32r0-compat	14 months ago
CaloTowerTool.h	Merge branch 'master' into fix-ServiceHandles	14 months ago

Example: RecCalorimeter tool

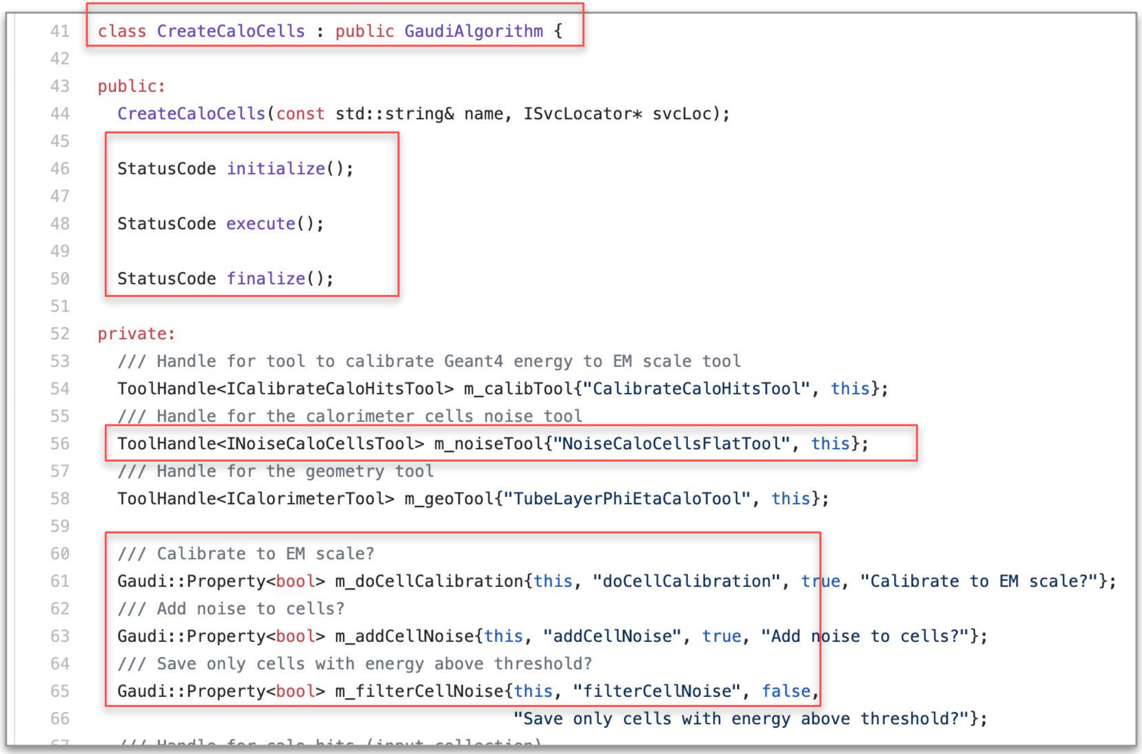
Reconstruction/RecCalorimenter/src/components/NoiseCaloCellsFromFileTool.h



```
class NoiseCaloCellsFromFileTool : public GaudiTool, virtual public INoiseCaloCellsTool {
public:
  NoiseCaloCellsFromFileTool(const std::string& type, const std::string& name, const IInterface* parent);
  virtual ~NoiseCaloCellsFromFileTool() = default;
  virtual StatusCode initialize() final;
  virtual StatusCode finalize() final;

  /** @brief Create random CaloHits (gaussian distribution) for the vector of cells (aCells).
   * Vector of cells must contain all cells in the calorimeter with their cellIDs.
   */
  virtual void addRandomCellNoise(std::unordered_map<uint64_t, double>& aCells) final;
  /** @brief Remove cells with energy bellow threshold*sigma from the vector of cells
   */
  virtual void filterCellNoise(std::unordered_map<uint64_t, double>& aCells) final;

  /// Open file and read noise histograms in the memory
  StatusCode initNoiseFromFile();
  /// Find the appropriate noise constant from the histogram
  double getNoiseConstantPerCell(int64_t aCellID);

private:
  /// Handle for tool to get cell positions
  ToolHandle<ICellPositionsTool> m_cellPositionsTool{"CellPositionsDummyTool", this};

  /// Add pileup contribution to the electronics noise? (only if read from file)
  Gaudi::Property<bool> m_addPileup{this, "addPileup", true,
                                    "Add pileup contribution to the electronics noise? (only if read from file)"};
  /// use segmentation in case no cell psotion tool defined.
  Gaudi::Property<bool> m_useSeg{this, "useSegmentation", true, "Specify if segmentation is used to determine cell position."};
  /// Name of the file with noise constants
  Gaudi::Property<std::string> m_noiseFileName{this, "noiseFileName", "", "Name of the file with noise constants"};
```

Example: RecCalorimeter algorithm

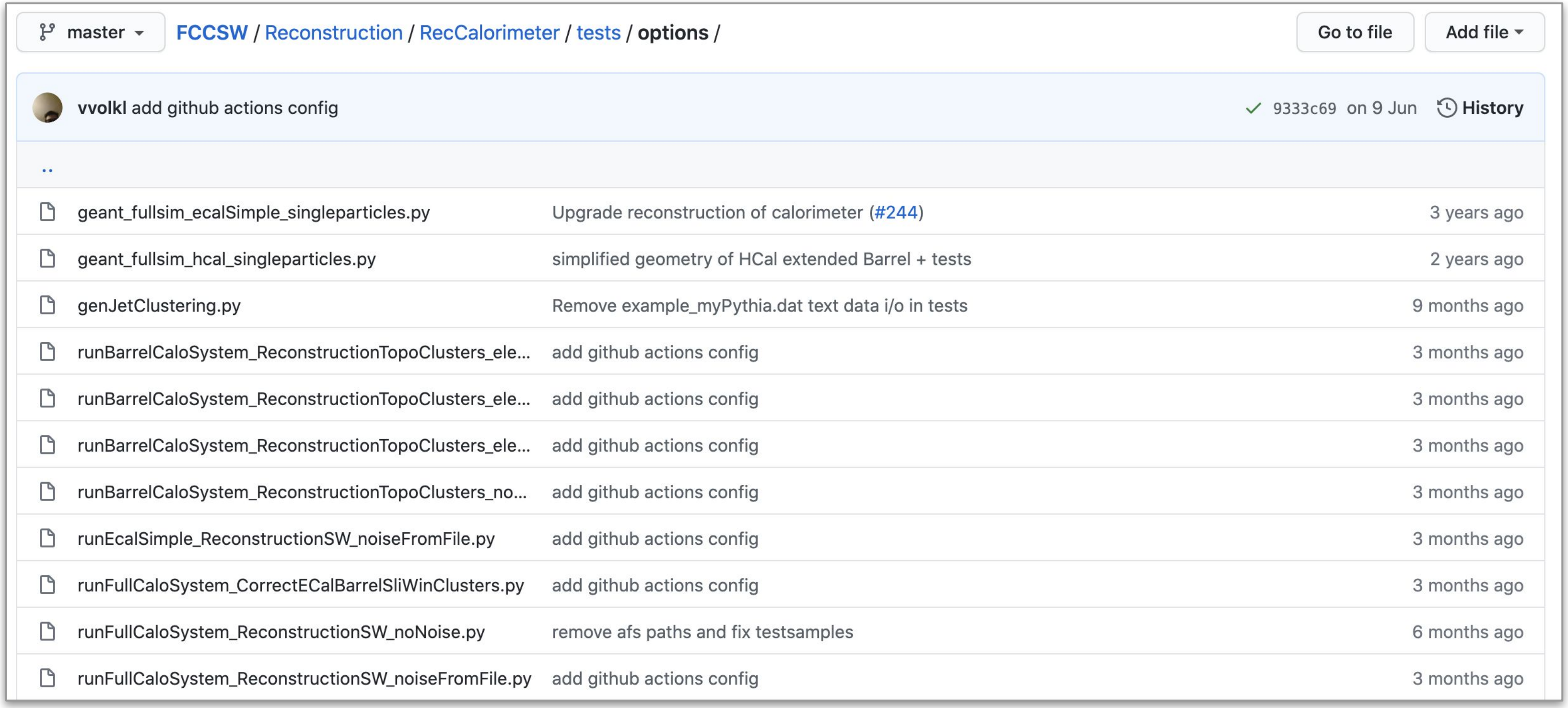
Reconstruction/RecCalorimenter/src/components/CreateCaloCells.h



```
class CreateCaloCells : public GaudiAlgorithm {

public:
  CreateCaloCells(const std::string& name, ISvcLocator* svcLoc);

  StatusCode initialize();

  StatusCode execute();

  StatusCode finalize();

private:
  /// Handle for tool to calibrate Geant4 energy to EM scale tool
  ToolHandle<ICalibrateCaloHitsTool> m_calibTool{"CalibrateCaloHitsTool", this};
  /// Handle for the calorimeter cells noise tool
  ToolHandle<INoiseCaloCellsTool> m_noiseTool{"NoiseCaloCellsFlatTool", this};
  /// Handle for the geometry tool
  ToolHandle<ICalorimeterTool> m_geoTool{"TubeLayerPhiEtaCaloTool", this};

  /// Calibrate to EM scale?
  Gaudi::Property<bool> m_doCellCalibration{this, "doCellCalibration", true, "Calibrate to EM scale?"};
  /// Add noise to cells?
  Gaudi::Property<bool> m_addCellNoise{this, "addCellNoise", true, "Add noise to cells?"};
  /// Save only cells with energy above threshold?
  Gaudi::Property<bool> m_filterCellNoise{this, "filterCellNoise", false,
                                          "Save only cells with energy above threshold?"};
  /// Handle for calo hits (input collection)
```

Example: RecCalorimeter options

Reconstruction/RecCalorimenter/tests/options: collection of jobOptions files (can be used in software tests via proper definition in CMakeLists.txt)



FCCSW / Reconstruction / RecCalorimeter / tests / options /

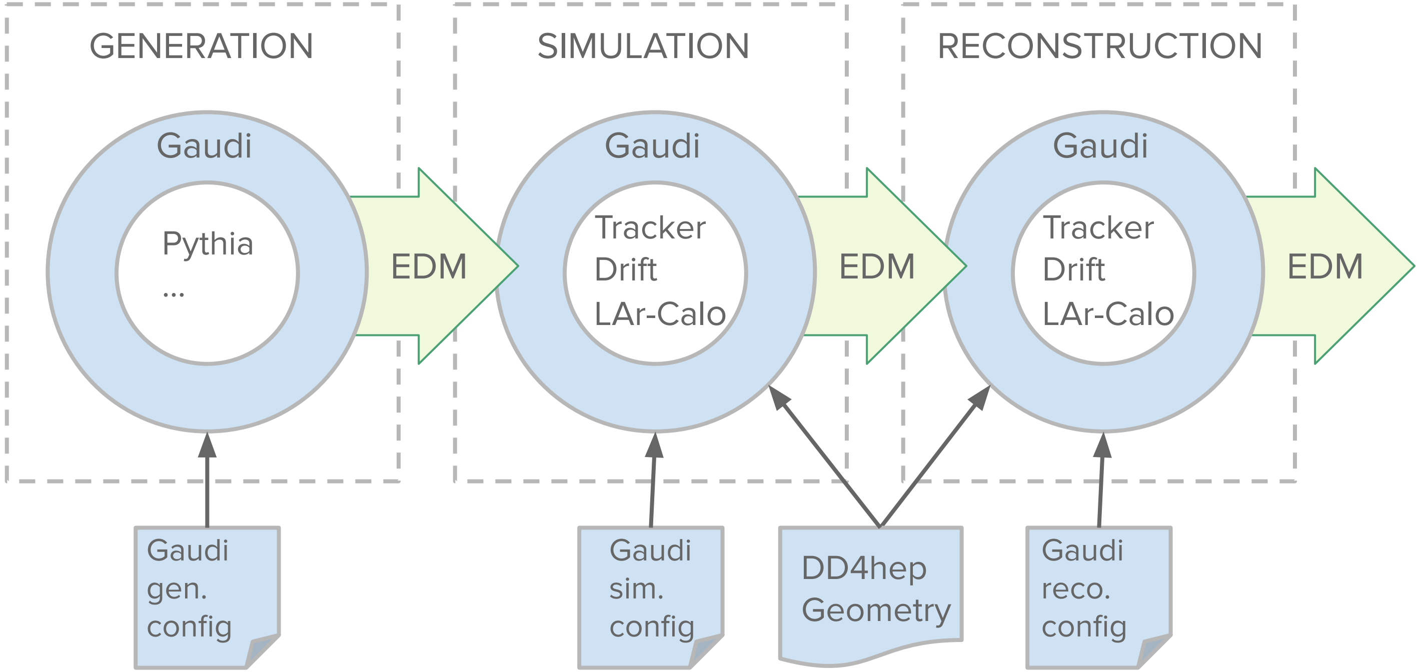
vvolkl add github actions config ✓ 9333c69 on 9 Jun

..

geant_fullsim_ecalSimple_singleparticles.py	Upgrade reconstruction of calorimeter (#244)	3 years ago
geant_fullsim_hcal_singleparticles.py	simplified geometry of HCal extended Barrel + tests	2 years ago
genJetClustering.py	Remove example_myPythia.dat text data i/o in tests	9 months ago
runBarrelCaloSystem_ReconstructionTopoClusters_ele...	add github actions config	3 months ago
runBarrelCaloSystem_ReconstructionTopoClusters_ele...	add github actions config	3 months ago
runBarrelCaloSystem_ReconstructionTopoClusters_ele...	add github actions config	3 months ago
runBarrelCaloSystem_ReconstructionTopoClusters_no...	add github actions config	3 months ago
runEcalSimple_ReconstructionSW_noiseFromFile.py	add github actions config	3 months ago
runFullCaloSystem_CorrectECalBarrelSliWinClusters.py	add github actions config	3 months ago
runFullCaloSystem_ReconstructionSW_noNoise.py	remove afs paths and fix testsamples	6 months ago
runFullCaloSystem_ReconstructionSW_noiseFromFile.py	add github actions config	3 months ago

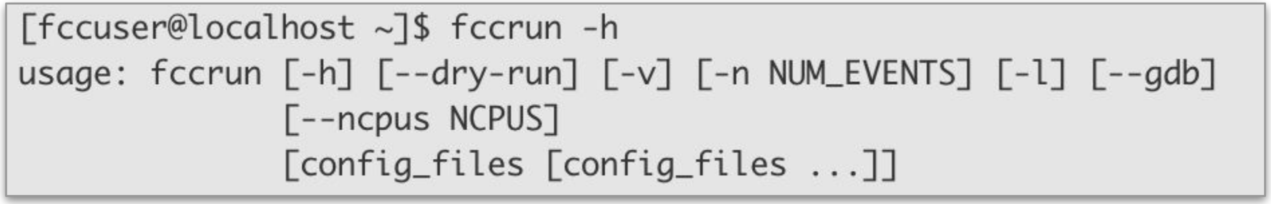
How do we run all this?

Schematic overview



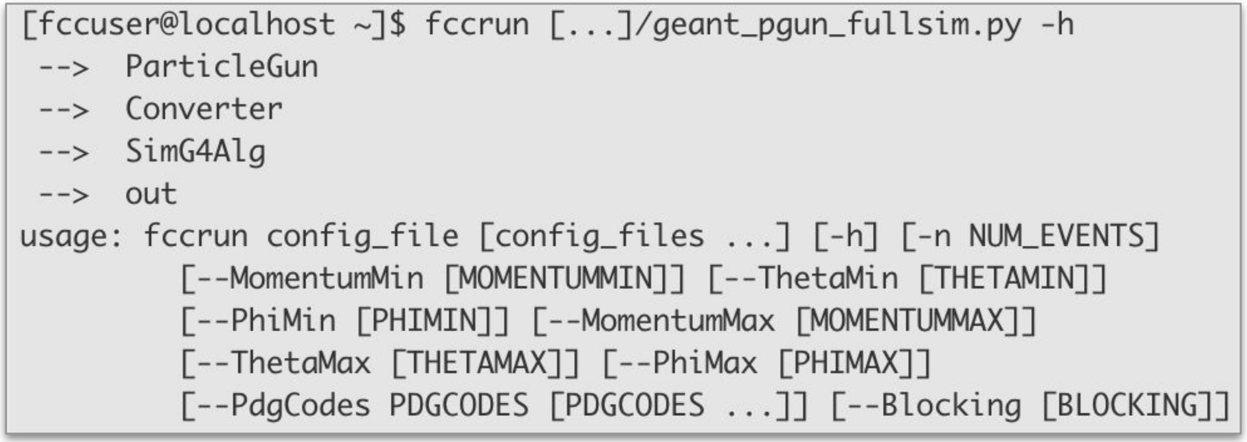
fccrun

- *fccrun* provides some overall option



```
[fccuser@localhost ~]$ fccrun -h
usage: fccrun [-h] [--dry-run] [-v] [-n NUM_EVENTS] [-l] [--gdb]
              [--ncpus NCPUS]
              [config_files [config_files ...]]
```

- Python jobOptions scripts are self-documenting (algorithms run are also shown)



```
[fccuser@localhost ~]$ fccrun [...]/geant_pgun_fullsim.py -h
 -->  ParticleGun
 -->  Converter
 -->  SimG4Alg
 -->  out
usage: fccrun config_file [config_files ...] [-h] [-n NUM_EVENTS]
         [--MomentumMin [MOMENTUMMIN]] [--ThetaMin [THETAMIN]]
         [--PhiMin [PHIMIN]] [--MomentumMax [MOMENTUMMAX]]
         [--ThetaMax [THETAMAX]] [--PhiMax [PHIMAX]]
         [--PdgCodes PDGCODES [PDGCODES ...]] [--Blocking [BLOCKING]]
```

- Processing requires the specification of a minimal full set of options

Summary

- FCCSW is a Gaudi project
  - Profits exploits all the built-in functionality of the framework
- Users
  - No need to know internals, just understand functionality
- Developers / contributors
  - Need to understand
    - How {algorithms, tools, services) work (C++)
    - How jobOptions work (Python)