

# Packages

Last update on Jan 22nd, 2024

- [aidaTT](#)
- [acts](#)
  - [ACTSTracking](#)
- [bbq](#)
- [CED](#)
  - [CEDViewer](#)
- [ClickPerformance](#)
- [Clupatra](#)
- [CondDBMySQL](#)
- [ConformalTracking](#)
- [DD4hep](#)
  - [DD4hepExamples](#)
  - [MarlinDD4hep](#)
- [detector-simulation](#)
- [FastJet](#)
  - [FastJetContrib](#)
  - [MarlinFastJet](#)
- [FCalClusterer](#)
- [ForwardTracking](#)
- [Garlic](#)
- [GBL](#)
- [gear](#)
- [ilcutil](#)
- [iLCInstall](#)
- [ILDPerformance](#)
- [KalTest](#)
  - [KalDet](#)
  - [DDKalTest](#)
- [KiTrack](#)
  - [KiTrackMarlin](#)
- [lccd](#)
- [LCFIPlus](#)
- [LCFIVertex](#)
- [lcgeo](#)
- [lcio](#)
- [LCTuple](#)
- [LICH](#)
- [Marlin](#)
- [MarlinKinfitter](#)
  - [MarlinKinfitterProcessors](#)
- [MarlinReco](#)
- [MarlinTrk](#)
  - [MarlinTrkProcessors](#)
- [MarlinUtil](#)
- [MuonCutil](#)
- [MuonCVXDDigitiser](#)
- [Overlay](#)
- [PandoraPFANew](#)
  - [PandoraAnalysis](#)
  - [DDMarlinPandora](#)
- [pathfinder](#)
- [Physsim](#)
- [RAIDA](#)
- [sio](#)

NOTE: In **green** the packages managed in the [muoncollider repository](#)

In **red** the packages removed from the muoncollider releases

## ***aidaTT***

Tracking toolkit developed in the [AIDA project](#)

Last version is [v00-10](#) released the Dec 11, 2019

[Github repository](#)



**Acts** Common Tracking Software for (charged) particle track reconstruction in (high energy) physics experiments implemented in modern C++.

Last version is **v32.0.0** released the Jan, 19 2024

[Github repository](#)

## ACTSTracking

MuonC package for running track reconstructions using the ACTS library. For more documentations see [here](#).

It provides the library: **libACTSTracking.so** with these processors:

1. ACTSDuplicateRemoval
  - If tracks share more than 50% of hits, then remove the best one.
2. ACTSSeededCKFTrackingProc
  - Fits tracks using the Combinatorial Kalman Filter algorithm with triplet seeding.
3. ACTSTruthCKFTrackingProc
  - Fits tracks using the Combinatorial Kalman Filter algorithm with MC particles as seeds.
4. ACTSTruthTrackingProc
  - Builds and fits tracks out of all hits associated to an MC particle.
5. TrackTruthProc
  - Associates MCParticle to a reconstructed Track.

Last version is **v1.1.0** released the Dec 7, 2022

[Github repository](#)

## bbq

*Removed from version v02-08-MC*

A TPC event display, see [these slides](#)

Last version is v00-01-04 released in the 2010

[Svn repository](#)

## CED

CED (C Event Display) is a server client application for OpenGL drawing

Last version is **v01-09-04** released the Jun 29, 2022

[User manual](#)

[Github repository](#)

## CEDViewer

Marlin processor for the CED event display

It provides the library: **libCEDViewer.so** with these processors:

1. CEDViewer
  - Event display of LCIO objects - based on CED by A.Zhelezov.
2. DDCEDViewer
  - Event display of LCIO objects - based on CED by A.Zhelezov. Drawing detector geometry and LCIO data; removed dependence on GEAR.
3. DSTViewer
  - CED based event display for DST files.
4. DrawCLIC
  - Draws the CLIC detector in CED.
5. GenericViewer
  - Drawing Utility.
6. HEP\_RecordProcessor
  - HEP record balance and drawing.
7. VertexViewer
  - Vertex Drawing Utility.

Last version is **v01-19-01** released the Jun 29, 2022

[Github repository](#)

## ClcPerformance

Package to evaluate the performance of the CLIC detector simulation

It provides the library: **libClcPerformance.so** with these processors:

1. CLICRecoConfig
  - Allows one to configure the processors to run via command line.
2. ClcEfficiencyCalculator
  - Calculates the tracking efficiency and makes some performance plots.
3. ClcTrackingValidate
  - Makes use of MC information at particle and hit level and compares the reconstructed hits and tracks to validate the tracking.
4. HitResiduals
  - Plots te residual between the track fit and the hit in the local coordinate system u,v,w.
5. JetAnalyzer
  - Calculates properties of calorimeter showers.
6. ShowerStudy
  - Calculates properties of calorimeter showers.
7. TrackChecker
  - Plots pull distributions, track resolution parameters and reconstruction efficiencies.
8. TrueMCintoRecoForJets
  - Description not given.

Last version is **v02-05** released the Nov 24, 2023

[Github repository](#)

## Clupatra

*Removed from version v02-08-MC*

Topological pattern recognition for the TPC

It provides the library: **libClupatra.so** with these processors:

1. ClupatraProcessor
2. DebugDigiProcessor
3. TrackCheckMCTruth
4. TrackEfficiency

Last version is **v01-03** released the Mar 28, 2018

[Github repository](#)

## CondDBMySQL

*Removed from version v02-08-MC*

Interface to MySQL Conditions Database

Last version is **CondDBMySQL\_ILC-0-9-7** released the Nov 25, 2019

[Github repository](#)

## ConformalTracking

Package for running pattern recognition based on conformal mapping and cellular automaton

It provides the library: **libConformalTracking.so** with these processors:

1. ConformalTracking
  - Constructs tracks using a combined conformal mapping and cellular automaton approach.
2. ConformalTrackingV2
  - Constructs tracks using a combined conformal mapping and cellular automaton approach with a predefined steps configuration.

Last MuonC version is **v01-12-MC** released the Nov 2, 2021

Last official version is **v01-12** released the Nov 8, 2023 

[Conformal Tracking for all-silicon trackers at future electron-positron colliders](#)

[MuonC github repository](#)

## **DD4hep**

DD4Hep Detector Description Toolkit for High Energy Physics

Last version is [v01-27-02](#) released the Dec 15, 2023

DD4hep: A Detector Description Toolkit for High Energy Physics Experiments, AIDA-CONFERENCE 18 October 2013

[Github repository](#)

## **DD4hepExamples**

Several examples on how to use DD4hep

[Github repository](#)

## **MarlinDD4hep**

Provides one processor to initialize a DD4hep detector geometry from a compact file for a Marlin job.

It provides the library: [libMarlinDD4hep.so](#) with this processor:

1. InitializeDD4hep
  - Reads a compact xml file and initializes the dd4hep::Detector object.

Last version is [v00-06-02](#) released the Dec 2, 2022

[Github repository](#)

## **detector-simulation**

**Moved inside lcggeo from version v00-18-MC**

Code related to the Muon Collider detector simulation: geometries, utility scripts, analysis configurations

Last version is [v01-04-MC](#) released the Feb 27, 2023

[Github repository](#)

## **FastJet**

A [software package](#) for jet finding in pp and e<sup>+</sup>e collisions.

Last version is [3.4.2](#) released the July 28, 2023

[User manual](#)

## **FastJetContrib**

The [fastjet-contrib](#) space is intended to provide a common location for access to 3rd party extensions of FastJet.

Last version is [1.053](#) released the September 29, 2023

[SVN repository](#)

## **MarlinFastJet**

MarlinFastJet is the interface between Marlin and FastJet

It provides the library: [libMarlinFastJet.so](#) with these processors:

1. FastJetClustering
  - FastJet Clustering.
2. FastJetProcessor
  - Using the FastJet library to identify jets.
3. FastJetTopTagger
  - Using the FastJet tool JHTagger to identify top jets.

Last version is [v00-05-03](#) released the Jun 29, 2022

[Github repository](#)

## FCalClusterer

Removed from version [v02-08-MC](#)

Reconstruction for the Forward Calorimeters of Future e+e- colliders developed by the [FCAL Collaboration](#)

It provides the library: [libFCalClusterer.so](#) with these processors:

1. BeamCalClusterReco
2. DrawBeamCalFromDD4hep
3. MarlinLumiCalClusterer
4. ReadBeamCal

Last version is [v01-00-06](#) released the Oct 24, 2023

[Github repository](#)

## ForwardTracking

Track Reconstruction for the Forward Direction (for the FTD)

It provides the library: [libForwardTracking.so](#) with these processors:

1. FTDBackgroundProcessor
  - Creates salt and pepper background hits on the FTD.
2. ForwardTracking
  - Reconstructs tracks through the FTD.
3. OverlapChecker
  - Checks if tracks are overlapping.
4. SiliconEndcapTracking
  - Reconstructs tracks through the FTD.
5. StepAnalyser
  - Stores information about the path of a particle.
6. TrackPickProcessor
  - Makes a collection of only one / a few true tracks.
7. TrackingFeedbackProcessor
  - Gives feedback about the Track Search.
8. TrueTrackCritAnalyser
  - Analysis of criteria for the Cellular Automaton.

Last version is [v01-14-02](#) released the Jul 24, 2023

[Github repository](#)

## Garlic

Garlic is a Marlin Processor to identify photons and electrons.

It provides the library: [libGarlic.so](#) with these processors:

1. ECALPreClustering
  - Preclusters ECAL hits to define ROIs for Garlic.
2. GarlicProcessor
  - Clustering and photon recognition.
3. garlicCutCalculator
  - Makes histograms of EM shower properties.

Last version is [v03-01](#) released the Mar 28, 2018

[Github repository](#)

## GBL

General Broken Lines: track refitting with broken lines in 3D

Last version is [V02-02-01](#) released the Aug 4, 2020

[Github repository](#)

## gear

GEAR: GEometry Api for Reconstruction

Last version is [v01-09-01](#) released the Jun 29, 2022

[Github repository](#)

## ***ilcutil***

A utility package for the iLCSoft software framework.

Last version is [v01-07-01](#) released the Jun 9, 2023

[Github repository](#)

## ***iLCInstall***

Installation script that enable a fully automated installation of iLCSoft with minimal user intervention

Last version is [v02-05-MC](#) released the Apr 26, 2023

[Github repository](#)

## **ILDPerformance**

Removed from version [v02-08-MC](#)

Package to evaluate the Performance of the ILD detector simulation

It provides the library: [libILDPerformance.so](#) with these processors:

1. DDDiagnostics
2. PIDTree
3. Pi0Tree
4. Residuals
5. Skim
6. TrackerHitCounter
7. WWZZProcessor
8. dEdxAnalyser
9. validateAutoProcessor
10. validateDST\_SinglePhotonProcessor
11. validationPreProcessor

Last version is [v01-12](#) released the Jun 26, 2023

[Github repository](#)

## ***KalTest***

Kalman Filter tool (K. Fujii et al)

Last version is [v02-05-01](#) released the Jun 29, 2022

[Kalman-filter-based track fitting in non-uniform magnetic field with segment-wise helical track model](#) submitted on 21 Oct 2013

[Github repository](#)

## ***KalDet***

Define detector geometry

Last version is [v01-14-01](#) released the Mar 28, 2018

[Github repository](#)

## ***DDKalTest***

Interface between KalTest fitter and DD4hep based geometry

Last version is [v01-07](#) release the Jun 29, 2022

[Github repository](#)

## ***KiTrack***

Toolkit for Tracking. Consists of KiTrack (Cellular Automaton, a Hopfield Neural Network, the hit and track classes) and Criteria (the criteria classes).

Last version is [v01-10](#) released the Dec 11, 2019

[Github repository](#)

## **KiTrackMarlin**

Implementation of classes for the use of KiTrack by Marlin

Last version is [v01-13-02](#) released the Dec 2, 2022

[Github repository](#)

## **lccd**

*Removed from version [v02-08-MC](#)*

Linear Collider Conditions Data framework

Last version is [v01-05-01](#) released the Jun 29, 2022

[Github repository](#)

## **LCFIPlus**

Flavor tagging code for ILC detectors, see [documentation](#)

It provides the library: [libLCFIPlus.so](#) with these processors:

1. LcfiplusProcessor
  - Lcfiplus general processor.
2. TrackToPFOConverterProcessor
  - Processor to convert track collection into ReconstructedParticle collection.

Last version is [v00-10-01](#) released the Nov 19, 2021

[LCFIPlus: A Framework for Jet Analysis in Linear Collider Studies](#) submitted on 28 Jun 2015

[Github repository](#)

## **LCFIVertex**

Package for vertex finding as well as vertex charge determination in b- and c-jets.

It provides the library: [libLCFIVertexProcessors.so](#) with these processors:

1. ConversionTagger
  - Does conversion and V0 tagging.
2. DSTAIDAPlotProcessor
  - Plots various outputs from the flavour tag.
3. DSTCollectionProcessor
  - Takes the flavour tag info, mc truth info, and some NN input info, and adds it as Particle ID info to the jets, for the DSTs.
4. DSTPlotProcessor
  - Plots various outputs from the flavour tag.
5. FlavourTag
  - Performs a flavour tag using previously trained neural nets.
6. FlavourTagInputsProcessor
  - Takes a set of vertices as a decay chain with its associated jet and calculates flavour tag inputs storing them in the Jet RP's pid.
7. KnowYourInputs
  - Evaluates hits+clusters+tracks.
8. LCFIAIDAPlotProcessor
  - Creates an AIDA plot of the LCFIVertex tagging efficiency-purity values and various other things.
9. NeuralNetTrainer
  - Trains a neural net from the lcio file.
10. PerEventIPFitterProcessor
  - Per Event IP fitter - trims tracks to reach probability threshold.
11. Plot
  - Plots various outputs from the flavour tag.
12. RPCutProcessor
  - Cuts RPs based on several criteria removing those that have no tracks.
13. SignificanceFit
  - The aim of the processor is to calculate the resolution parameters used for the joint probability flavour tag. This is done by considering the tracks with negative impact parameters. The curve is modeled as a gaussian + two exponentials.
14. TrueAngularJetFlavourProcessor
  - Determine MC Jet Flavour by angular matching of heavy quarks to jets, also determine hadronic and partonic charge of jet.

15. V0Performance
  - Does performance analysis of conversion and V0 tagging.
16. VertexChargeProcessor
  - Takes a set of vertices as a decay chain with its associated jet and calculates the vertex charge.
17. ZVTOP\_ZVKINProcessor
  - Kinematic Vertex Reconstruction Algorithm.
18. ZVTOP\_ZVRESProcessor
  - ZVTOP\_ZVRES - Vertex Reconstruction Algorithm.


Last version is [v00-08](#) released the Dec 11, 2019

[The LCFIVertex package: vertexing, flavour tagging and vertex charge reconstruction with an ILC vertex detector](#) submitted on 20 Aug 2009

[Github repository](#)

## lcgeo

DD4hep based geometry models for lepton collider detectors

Last version is [v00-19-00](#) released the Nov 22, 2023 

MuonColl last version is [v00-18-01-MC](#) released the Apr, 26 2023

NB From version v00-18-MC it contains also the MuonCollider geometry (also known as detector-simulation).

[Github repository](#) Moved to [k4geo](#)

## lcio

LCIO ( Linear Collider I/O ) is a persistency framework and event data model for linear collider detector studies.

Last version is [v02-20-02](#) released the Dec 6, 2023

[LCIO - A persistency framework for linear collider simulation studies](#), CHEP03 March 24-28, 2003

[Github repository](#)

## LCTuple

Marlin package to creates a ROOT TTree with a column wise ntuple from LCIO collections. Some useful notes can be found [here](#).

It provides the library: [libLCTuple.so](#) with these processors:

1. CreateRecoRelations
  - Creates transient collections with realtions from ReconstructedParticles to Tracks and Clusters.
2. LCTuple
  - Creates a ROOT TTree with a column wise ntuple from LCIO collections.
3. MergeCollections
  - a. Creates a transient subset collection that merges all input collections.

Last version is [v01-15-MC](#) released the Feb 27, 2023

Official last version is [v01-14](#) released the Jun 29, 2022

[Github repository](#)

## LICH

*Removed from version [v02-08-MC](#)*

The processor can be used for producing sample to train the TMVA, or use the TMVA weight files for charged particle identification

It provides the library: [libLICH.so](#) with this processor:

1. LICH

Last version is [v00-01](#) released the Jan 10, 2017

[Github repository](#)

## Marlin

Modular Analysis and Reconstruction for the LINear Collider



It provides the following processors:

1. AIDAProcessor
  - Handles AIDA files. Creates on directory per processor. Processors only need to create and fill the histograms, clouds and tuples.
2. ConditionsProcessor – Available only with LCCD --
  - Provides access to conditions data transparently from LCIO files or a databases, using LCCD.
3. EventSelector
  - Returns true if given event was specified in EventList
4. LCIOOutputProcessor
  - Writes the current event to the specified LCIO outputfile. Needs to be the last ActiveProcessor.
5. MemoryMonitor
  - Prints out the memory consumption at defined interval.
6. PatchCollections
  - Adds missing collections to the event - either parsed from inputfiles or as given in PatchCollections.
7. SimpleFastMCPProcessor
  - Creates ReconstructedParticles from MCParticles according to the resolution given in the steering file.
8. Statusmonitor
  - Prints out information on running Marlin Job: prints number of runs run and current number of the event. Counting is sequential and not the run or event ID.
9. StdHepReader
  - Reads StdHep files as input and creates LCIO events with MCParticle collections. Make sure to not specify any LCIOInputFiles in the steering in order to read StdHep files.
10. TestProcessor
  - Simple processor to test the marlin application. Prints run and event number.

Last version is [v01-19](#) released the Dec 9, 2022

[Marlin 1.17.1 documentation](#)

[Github repository](#)

## ***MarlinKinfIt***

Kinematic fitting library for Marlin

It provides the library: [libMarlinKinfIt.so](#)

Last version is [v00-06-01](#) released the Jun 29, 2022

[Documentation](#)

[Github repository](#)

## ***MarlinKinfItProcessors***

Marlin Processors to use MarlinKinfIt

It provides the library: [libMarlinKinfItProcessors.so](#) with these processors:

1. DijetTester
  - Does a 4C fit on 2 jet events (Px, Py, Pz, E).
2. MCParticleFilter
  - Chooses which MCParticles to keep.
3. MassConstraintFitter
  - Description not given..
4. PhotonResponseAdjuster
  - Adjusts Photons from filtered pandora pfos collection.
5. SimpleVertexProcessor
  - Fits 2 charged tracks to a single vertex.
6. TTBarExample
  - Does a 6C fit on 6jet ttbar events (Px, Py, Pz, E, M12 = M34 = M\_W (for all permutations assuming the b-jets are tagged)).
7. TopTester
  - Does a 7C fit on 6 jet events (Px, Py, Pz, E, M34 = M56 = 80.4 GeV, M134 = M256).
8. TrackResponseAdjuster
  - Smears Charged Particles from filtered Monte Carlo Particle collection.
9. WW5CFit
  - Does a 5C fit on 4 jet events (Px, Py, Pz, E, M12 = M34 (for all three permutations)).
10. ZH5CFit
  - Does a 5C fit on 4 jet events (Px, Py, Pz, E, M12 = MZ (for all six permutations)).
11. ZHllqq5CFit
  - Does a 5C fit on ZH->mumubb events.

Last version is [v00-05](#) released the Jun 29, 2022

[Github repository](#)

Assembly of various Marlin processor for reconstruction

It provides the library: [libMarlinReco.so](#) with these processors:

1. AbsCalibr
  - Calorimeter Absolute Energy Calibration.
2. Add4MomCovMatrixCharged
  - Set the covariance matrix in (P,E) for all charged pfos in PandoraPFOs Collection.
3. AddClusterProperties
  - AddClusterProperties does whatever it does.
4. AnalyseSidEdxProcessor
  - Reads Si tracker dEdx data from the .slcio file and stores them in a root file for analysis.
5. anaPix
  - It tests TTree from FPCCDPixelHits.
6. AngularCorrection\_dEdxProcessor
  - Correct\_AngularCorrection\_dEdxProcessor: Makes a hard angular-based correction of dEdx for all the Tracks in the event.
7. BCalTagEfficiency
  - only if FORTRAN is available --
  - ForwardVeto propagates electrons or photons from IP to the BeamCal, and it gives the Energy and Momentum of the detected ones to the output ReconstructedParticles collection.
8. BruteForceEcalGapFiller
  - Makes a collection of ECAL gap hits.
9. CCDDigitizer
  - Should create VTX TrackerHits from SimTrackerHits.
10. CLICDSTChecker
  - Prints out information in DST in a useful format.
11. CLICPfoSelector
  - Selects Pfos from full PFO list using timing cuts.
12. CLICPfoSelectorAnalysis
  - Produces a tree and scatter plots using PFO variables defined in the CLICPfoSelector.
13. CheckPlots — NOT COMPILED by DEFAULT ---
  - Produces Check Plots.
14. ClusterCheater5\_3
  - Creates true clusters.
15. ComputeShowerShapesProcessor
  - Cluster Shower Profile extraction using Fitting.
16. Compute\_dEdxProcessor
  - dE/dx calculation using truncation method.
17. Compute\_dEdxProcessor2021
  - dE/dx calculation using truncation method, angular correction applied after smearing.
18. CreatePDFProcessor
  - Processor for PDF templates creation.
19. CreatePDFs
  - Creates from a gauge sample pdf files for a likelihood based particle identification.
20. DDStripSplitter
  - Applies SSA to a strip-based calorimeter.
21. DistilledPFOcreator
  - Merge PandoraPFOs and GammaGammaParticles into DistilledPFOs taking care to avoid double counting.
22. EMShowerFinder
  - A photon finder processor based on the KIT and KITutil 'classes and functions.
23. ErrorFlow
  - Processor computes total error of a jet based on the information from each sub-detector.
24. EvaluateTauFinder
  - Checks performance of TauFinder and writes output to root file.
25. FPCCDClustering
  - It tests TrackerHits from FPCCDPixelHits.
26. FPCCDDigitizer
  - Should create FPCCD's VTXPixelHits from SimTrackerHits.
27. Fox
  - Calculates Fox-Wolfram moments.
28. G2CD
  - Gaseous Calorimeter Digitizer: take 1mm Simulated Hits as input.
29. GammaGammaCandidateFinder
  - Take initial BestPi0 implementation and extend to all combinations consistent with pi0 mass.
30. GammaGammaCandidateTruthFilter
  - Description not given.
31. GammaGammaSolutionFinder
  - Description not given.
32. hybridRecoProcessor
  - Description not given.
33. ILDCaloDigi
  - Performs simple digitization of sim calo hits.
34. IsoLepTrainingProcessor
  - Description not given.
35. IsolatedLeptonFinderProcessor
  - Isolated Lepton Finder Processor.
36. IsolatedLeptonTaggingProcessor
  - Description not given.

37. IsolatedPhotonTaggingProcessor
  - Description not given.
38. JetPFOsProcessor
  - Description not given.
39. KIT
  - Description not given.
40. KinkFinder
  - Finds kinks, prongs and splits.
41. LDCCaloDigi
  - Performs simple digitization of sim calo hits.
42. LikelihoodPIDProcessor
  - Particle ID code using Bayesian Classifier.
43. MCTruthJetEnergy
  - Calculates the MC truth jet energy of the MCParticles associated to the reconstructed particles in the jet.
44. MokkaCaloDigi
  - Mokka digitizer.
45. MomentumCov
  - Calculate the covariance matrix in the momenta space.
46. NNClusterProcessor
  - Simple nearest neighbour clustering.
47. NewLDCCaloDigi
  - Performs simple digitization of sim calo hits.
48. PFOID
  - Performs particle identification.
49. photonCorrectionProcessor
  - Applies an energy correction to photon-like PFOs.
50. PrepareRECParticles
  - Converts input to ReconstructedParticles and puts them into a new collection making sure all the information which is needed to run the TauFinder is provided.
51. QuarkJetPairing
  - Description not given.
52. RealisticCaloDigiScinPpd
  - Performs digitization of sim calo hits.
53. RealisticCaloDigiSilicon
  - Digitization of silicon simcalo hits.
54. RealisticCaloRecoScinPpd
  - Performs fist reconstruction of scintillator calo hits.
55. RealisticCaloRecoSilicon
  - Performs fist reconstruction of silicon ECAL hits.
56. RecoMCTruthLinker
  - Links ReconstructedParticles to the MCParticle based on number of hits used.
57. ReconstructedParticleImpl\_CopyProcessor
  - ReconstructedParticleImpl\_CopyProcessor: Copies each entry of a LCIO collection of ReconstructedParticleImpl. Select in the optional parameters which members should not be copied.
58. SatoruJetFinderProcessor — NOT COMPILED by DEFAULT ---
  - A multi algorithm jet finder.
59. SelectReconstructedParticle
  - Selects particles from all reconstructed particles to be used for the thrust finder.
60. SiStripClus — NOT COMPILED by DEFAULT ---
  - Marlin processor intended for cluster finding - using digitized data worked out with SiStripDigi.
61. SiStripDigi — NOT COMPILED by DEFAULT ---
  - Marlin processor intended for detailed digitization of silicon strip sensors.
62. SiTracker\_dEdxProcessor
  - Calculates dE/dx for planar silicon trackers.
63. SimDigital
  - Creates SDHCAL digitized CalorimeterHits from SDHCAL SimCalorimeterHits.
64. SimDigitalLinkToParticles
  - Links calorimeter hits to MC particles.
65. SimDigitalToEnergy
  - Transforms the threshold value of digitized SDHCAL CalorimeterHits into energy.
66. SimpleCaloDigi
  - Performs simple digitization of sim calo hit.
67. SimpleFCalDigi
  - Performs simple digitization of SimCalorimeter hits in forward calorimeters.
68. SimpleMuonDigi
  - Performs simple digitization of sim muon hits.
69. Sphere
  - Calculates eigenvalues of sphericity tensor.
70. TJetsPFOAnalysisProcessor
  - Basically a copy-pasted version of the PFOAnalysis processor working on individual TrueJet jets.
71. TOFEstimators
  - Computes time-of-flight of the chosen ReconstructedParticle to the specified end point (SET hit or Ecal surface). To be used for a further particle ID.
72. TPCDigiProcessor
  - Produces TPC TrackerHit collection from SimTrackerHit collection, smeared in RPhi and Z.
73. TaJetClustering
  - Tau Jet Finding.
74. TauFinder
  - Writes tau candidates as ReconstructedParticles into collection. It runs on a collection of ReconstructedParticels.

75. ThrustReconstruction
  - Calculates thrust axis and thrust value of event using different algorithms.
76. TrackToRecoParticleConverter
  - Processor to convert track collection into ReconstructedParticle collection
77. TrackZVertexGrouping.
  - Groups tracks based on their Z0 significance.
78. TrueJet
  - Description not given.
79. TruthVertexFinder
  - Extracts a secondary vertex from generator collection.
80. V0Finder
  - V0 Finder Processor.
81. VTXBgClusters
  - Should create VTX TrackerHits from SimTrackerHits.
82. VTXDigiProcessor
  - Should create VTX TrackerHits from SimTrackerHits.
83. VTXDigitizer
  - Should create VTX TrackerHits from SimTrackerHits.
84. VTXNoiseClusters
  - Adds SimTrackerHits with salt'n pepper noise clusters.
85. VTXNoiseHits
  - Should create VTX TrackerHits from SimTrackerHit.
86. VertexRestorer — NOT COMPILED by DEFAULT ---
  - VertexRestorer.
87. YThresh — only if FORTRAN is abilitate --
  - Finds the crossover value of the yCut variable from NMin to NMin+1 jets found using durhamycut.
88. ZFinder
  - Description not given.

Last version is [v01-34](#) released the Jul 7, 2023

[Event Reconstruction with MarlinReco at the ILC](#) submitted on 20 Feb 2007

[Github repository](#)

## MarlinTrk

Tracking Package based on LCIO and GEAR, primarily aimed at providing track fitting in Marlin.

Last version is [v02-09-01](#) released the Jun 29, 2022

[Github repository](#)

## MarlinTrkProcessors

A collection of Tracking Related Processors Based on MarlinTrk

It provides the library: [libMarlinTrkProcessors.so](#) with these processors:

1. CalcTrackerHitResiduals
  - Creates Track Collection from MC Truth. Can handle composite spacepoints as long as they consist of two TrackerHits.
2. ClonesAndSplitTracksFinder
  - Takes the track collection, checks for doubles and merges them in an output track collection.
3. DDCellsAutomatonMV
  - Description not given.
4. DDPlanarDigiProcessor
  - Creates TrackerHits from SimTrackerHits, smearing their position and time according to the input parameters. The geometry of the surface is taken from the DDRec::Surface associated to the hit via the cellID.
5. DDSpacePointBuilder
  - Combines si-strip measurements into 3D spacepoints (1TrackerHitPlanar+1TrackHitPlanar = 1 TrackerHit), that can be used by reconstruction.
6. DDTPCDigiProcessor
  - Produces TPC TrackerHit collection from SimTrackerHit collection, smeared in RPhi and Z. A search is made for adjacent hits on a pad row, if they are closer in z and r-phi than the steering parameters \_doubleHitResRPhi (default value 2.0 mm) and \_doubleHitResZ (default value 5.0 mm) they are considered to overlap. Clusters of hits smaller than \_maxMerge (default value 3) are merged into a single tracker hit, with the position given as the average position of the hits in phi and in z. Clusters which have \_maxMerge hits or more are determined to be identifiable as multiple hits, and are not added to the tracker hit collection. This of course means that good hits caught up in a cluster of background hits will be lost.
7. ExtrToSIT
  - Refits an input track collection (TPC or VXD), and used IMarlinTrk tools to propagate it to SIT.
8. ExtrToTracker
  - Refits an input VXD track collection and uses IMarlinTrk tools to propagate it to the main tracker.
9. FPCCDFullLDCTracking\_MarlinTrk
  - Performs full tracking in ILD detector.
10. FPCCDSiliconTracking\_MarlinTrk
  - Pattern recognition in silicon trackers.
11. FilterConeHits
  - Selects tracker hits in a cone opened around a MC particle direction.

12. FilterDoubleLayerHits
  - Removes hits in double layers if they don't have a corresponding close-by hit in the other sublayer.
13. FilterTimeHits
  - Selects tracker hits based on their time corrected for the time of flight.
14. FilterTracks
  - Filters a collection of tracks based on NHits and MinPt and outputs a filtered collection.
15. FullLDCTracking\_MarlinTrk
  - Performs full tracking in ILD detector.
16. RefitFinal
  - Calls finaliseLCIOTrack after taking the trackstate from the existing track. No re-sorting of hits is done. Some cuts on detector's hits or track's Chi2 can be applied.
17. RefitProcessor
  - Refits an input track collection, producing a new collection of tracks.
18. SetTrackerHitExtensions
  - Creates Track Collection from MC Truth. Can handle composite spacepoints as long as they consist of two TrackerHits.
19. SiliconTracking\_MarlinTrk
  - Pattern recognition in silicon trackers.
20. SplitCollectionByLayer
  - Splits a hit collection based on the layer number of the hits.
21. SplitCollectionByPolarAngle
  - Selects tracker hits based on their polar angle.
22. TrackSubsetProcessor
  - Takes tracks from multiple sources and outputs them (or modified versions, or a subset of them) as one track collection.
23. TruthTrackFinder
  - Builds tracks out of all hits associated to an MC particle.
24. TruthTracker
  - Creates Track Collection from MC Truth. Can handle composite spacepoints as long as they consist of two TrackerHits.

Last version is [v02-15-MC](#) released the Apr 26, 2023

Official last version is [v02-12-04](#) released the Jul 24, 2023 

[Github repository](#)

## MarlinUtil

Library that contains classes and functions that are used by more than one processor. In particular it contains the geometry classes that are used until we have the geometry for reconstruction package (GEAR).

It contains the following processors:

1. ConvertToMIPScale
  - Transforms the energy amplitude of each calorimeter hit passing the cut into the MIP scale.
2. CutOnCosThetaQQbar
  - The output condition of this processor is false as long as the  $|\cos(\theta)| > \text{cutCosTheta}$  and true otherwise.
3. CutOnGEANT4Bug
  - The output condition of this processor is true as long as no track has more than a factor of k more energy deposited in the calorimeter as its energy given by momentum and mass. This should cut out events where GEANT4 produces additional energy depositions. If at least one such a track is found the return value is false. Only tracks with an energy larger than eMin are taken into account.
4. DrawMCParticles
  - Draws the MC Particle prediction w/o calorimeters in the ced display.
5. PGDBP
  - Physical and Geometrical Database (for Boojum at least)
6. SelectEvents
  - Selects certain events from input files.
7. SimpleTimer
  - Offers simple timer utilities.
8. SkipNEvents
  - The output condition of this processor is false for the first n LCEvents. Afterwards it is set to true.

Last version is [v01-17-02](#) released the Oct 30, 2023

[Github repository](#)

## MuonCutil

MuonCutil is a utility package for the MuonCollider software framework

Last version is [v02-07-MC](#) released the Nov 2, 2021

[Github repository](#)

## MuonCVXDDigitiser

VXD digitiser processor for Muon Collider

It provides the library: [libMuonCVXDDigitiser.so](#) with this processor:

1. MuonCVXDDigitiser
  - Should create VTX TrackerHits from SimTrackerHits. Digitization follows the procedure adopted in the CMS software package using a timewindow to collect hits.

and the library [libMuonCVXDRealDigitiser.so](#) with the processor:

1. MuonCVXDRealDigitiser
  - should create VTX TrackerHits from SimTrackerHits. It contains also the simulation of the chip RD53A (experimental).

Last version is [v00-01](#) released the Apr, 26, 2023

[Github repository](#)


## Overlay

Packages for overlaying LCIO background events on simulator level

It provides the library: [libOverlay.so](#) with these processors:

1. FPCCDOOverlayBX
  - Overlays LCGeneVTXPixelHits of background lcio files for many bunch crossings.
2. JoinEvents
  - Opens an lcio file and adds all collections of the corresponding (run/event-number) events to the current event.
3. Overlay
  - Opens a second (chain of) lcio file(s) and overlays events.
4. OverlayBX
  - Overlays background lcio files for many bunch crossings.
5. OverlayEvents
  - Merges a number of events in a LCIO file into 1 event.
6. OverlayTiming
  - Processor to overlay events from the background taking the timing of the subdetectors into account.
7. OverlayTimingGeneric
  - a. Processor to overlay events from the background taking the timing of the subdetectors into account.

Last version is [v00-24-MC](#) released the Feb 28, 2023

Last official repository is [v00-23-01](#) released the Nov 21, 2023 

[Github repository](#)

## PandoraPFANew

Metadata package to bring together and build multiple [Pandora](#) libraries:

- [LCCContent](#) Algorithms and tools for LC event reconstruction
- [LArContent](#) Algorithms and tools for LAr TPC event reconstruction
- [PandoraMonitoring](#) ROOT-based Event Visualisation Environment and tree-writing functionality
- [PandoraSDK](#) Powerful Software Development Kit for pattern recognition algorithms

Last version is [v04-06-02](#) released on Jan 14, 2024

Tutorial [slides](#) (24th June 2019 - Fermilab)

[Documentation](#)

[Github repository](#)

## PandoraAnalysis

Pandora calibration and analysis tools in iLCSoft / Marlin framework

It provides the library: [libPandoraAnalysis.so](#) with these processors:

1. PandoraPFACalibrator
  - for calibration of PandoraPFA.
2. PfoAnalysis
  - Analyses output of PandoraPFANew.

Last version is [v02-00-01](#) released the Dec 11, 2019

[Github repository](#)

## DDMarlinPandora

Interface between Marlin and PandoraPFA

It provides the library: [libDDMarlinPandora.so](#) with these processors:

1. DDCaloDigi
  - Performs simple digitization of sim calo hits.
2. DDCaloDigi\_BIB
  - Performs simple digitization of sim calo hits, with BIB energy subtraction.
3. DDPandoraPFANewProcessor
  - Pandora reconstructs clusters and particle flow objects.
4. DDSimpleMuonDigi
  - Performs simple digitization of sim muon hits.

Last version is [v00-14-MC](#) released the Feb 28, 2023

Official last version is [v00-12](#) released the Nov 1, 2021

[Github repository](#)

## pathfinder

*Removed from version [v02-08-MC](#)*

**PA**ckage for Tracking with a Hough trafo **FINDER**

Last version is v01-13-02 released in the 2013

[SVN repository](#)

## Physsim

*Removed from version [v02-08-MC](#)*

Physsim is a matrix element package

It provides the library: [libPhyssim.so](#)

Last version is [v00-04-02](#) released the Jun 29, 2022

[Github repository](#)

## RAIDA

ROOT implementation of Abstract Interfaces for Data Analysis (AIDA)

Last version is [v01-11](#) released the Jun 26, 2023

[Github repository](#)

## sio

SIO is a persistency solution for reading and writing binary data in SIO structures called *record* and *block*.

Last version is [v00-02](#) released the Dec 6, 2023

[Github repository](#)

