

---

Subject: new ilcsoft release v01-12  
Posted by [engels](#) on Fri, 23 Sep 2011 16:38:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

a new release of ilcsoft (v01-12) is available.

This is a development release with many new features, targeted at bringing the software into shape for the DBD mass production.  
It is based on the new LCIO v2 (v02-00) that provides multiple track states and special 1D and 2D TrackerHits and a new version of Gear with extensions needed for the new C++ tracking code.

Please refer to the Release notes below for details.

Use this link to download the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-12.

Adapt the files ./releases/v01-12/release-scratch.cfg and ./releases/v01-12/release-versions.py to install the complete release from scratch at your site (more infos here).

Reference installations in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/i386_gcc34_sl4/v01-12 # SL4 32bit  
/afs/desy.de/project/ilcsoft/sw/i386_gcc41_sl5/v01-12 # SL5 32bit  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-12 # SL5 64bit
```

Please report any problems and questions regarding this release in this forum.

The iLCSoft team.

Release Notes for v01-12

```
#####  
#  
# iLCSoft release - v01-12  
#  
#####
```

-----  
afs reference installations - starting from release v01-09:

/afs/desy.de/project/ilcsoft/sw/

with subdirectories for different CPUs, operating systems and gcc versions.e.g:

i386\_gcc34\_sl4     # i386 CPU, 32 bit, gcc3.4, SL4 and compatible  
i386\_gcc41\_sl5     # i386 CPU, 32 bit, gcc4.1, SL5 and compatible  
x86\_64\_gcc41\_sl5    # i686 CPU, 64 bit, gcc4.1, SL5 and compatible

NOTE: if you haven't upgraded your system to SL5 yet, please do it as soon as possible.  
Support for SL4 systems will be deprecated after this release!

=====

New packages in iLCSoft added in v01-12

=====

MarlinTrk:

New tracking package for Marlin, using LCIO, GEAR and KalTest and KalDet.  
Defines an abstract interface for pattern recognition and track fitting.  
The current implementation of the fitter is done with KalTest and KalDet.

MarlinFastJet:

MarlinFastJet is a small wrapper around the FastJet library for Jet Finding.  
It has access to a many of the Infrared and Collinear Safe jet finding algorithms like  
kt\_algorithm, anti\_kt\_algorithm and SIScone.

MarlinKinfit (standalone):

Kinematic fitting library for Marlin.  
<https://www.desy.de/~blist/kinfit/doc/html/>

=====  
Updated versions of external tools  
=====

ROOT: 5.28.00f

GSL: 1.14

CMake: 2.8.5

-----  
Release notes of packages that have been changed  
-----

=====  
LCIO : v02-00  
=====

- added multiple trackstates for Track
  - Original functions getX() of Track will return: `trk.getTrackStates()[0].getX()`
  - TrackState::getLocation() returns one of the predefined locations:
    - AtOther = 0 ; // any location other than the ones defined below
    - AtIP = 1 ;
    - AtFirstHit = 2 ;
    - AtLastHit = 3 ;
    - AtCalorimeter = 4 ;
    - AtVertex = 5 ;
    - LastLocation = AtVertex ;
  - TrackState\* getTrackState(int location)  
returns the trackstate at the given location, where location is given by one of the predefined locations specified above
  - TrackState\* getClosestTrackState(float x, float y, float z)  
returns the trackstate closest to the given location

- getTrackState( TrackState::AtFirstHit )->getReferencePoint()
- getTrackState( TrackState::AtLastHit )->getReferencePoint()
- deprecated isReferecePointPCA()
- added TrackerHitPlane and TrackerHitZCylinder for extending the TrackerHit interface
- added BitField64::reset()
- added constructor in CellIDDecoder which takes a string argument for the encoding
- added some useful definitions that are used in the ILD software framework
  - in particular for encoding and decoding cellIDs.
- add spin and color flow information to MCParticle if available in stdhep file (LCStdHepRdr)
- clarified use of addMCParticleContribution
  - now two method signatures for standard mode and 'detailed' mode
  - no excpetion is thrown if stepPos==Null in detailed mode - use (0.,0.,0.) instead
- new tests added:
  - added some tests for the canonical ILD CellIDEncoding (under test\_trackerhit)
  - added tests for TrackerHitPlane and TrackerHitZCylinder classes
  - added tests for the TrackState class
  - added a test for LCRTRelations (extensions)
- bug fixes:
  - added printout of ndf for tracks
  - added printout of Cluster EnergyError
  - added printout of TrackerHit and SimTrackerHit cellid's in LCIO\_LONG operators
  - fixed strict aliasing warning for RTRelations in gcc when compiled with -O3 (Release mode)
  - fixed constness of member functions (LCTime)
  - applied fortran patches for gcc4.4 on ubuntu 32bit (Phillip Klenze, MPI)
  - removed generation of cpp headers in build.xml (from ant aid.generate target)
  - removed duplicate FindJAVA.cmake module
  - fixed path to MacroCheckPackageLibs.cmake and MacroCheckPackageVersion.cmake in LCIOConfig.cmake

- changes (already pre-released in v01-60 )

-----  
- implemented new methods:

LCReader::readRunHeader(int runNumber )

- C++ only:

LCReader::getNumberOfEvents()

LCReader::getNumberOfRuns()

(anajob now prints number of runs and events in the files it is going to read)

- added access mode to

LCReader::readRunHeader(int runNumber, int accessMode )

LCReader::readEvent(int runNumber, int evtNumber, int accessMode )

- added example for creating a simple NTuple from MCParticle collection

( ./examples/cpp/rootDict/writeNTuple.C )

- added spin and color flow information to MCParticle:

float[3] getSpin()

int[2] getColorFlow()

void setSpin( float[3] )

void setColorFlow( int[2] )

- added cellID0 and cellID1 to TrackerHit and SimTrackerHit:

int getCellID0()

int getCellID1()

void setCellID0( int )

void setCellID1( int )

\* in SimTrackerHit the methods get/setCellID() were deprecated  
these methods now call get/setCellID0()

- added optional storing of the position where the energy deposition (step) occurred in

## SimCalorimeterHit

only if bit flag LCIO.CHBIT\_STEP is set

float[3] getStepPosition( int i )

\* LCIO.CHBIT\_PDG was deprecated  
should now use LCIO.CHBIT\_STEP

- added get/setEnergyError to Cluster (c++/java/fortran)
- added ostream operators (H. Hoelbe)
- added example for creating a simple NTuple from MCParticle collection (examples/cpp/rootDict/writeNTuple.C)
- added new methods (C++ only):  
LCReader::getNumberOfEvents()  
LCReader::readRunHeader(int runNumber )
- added tool lcio\_event\_counter for counting events in slcio / stdhep files
- anajob now prints number of runs and events in the files it is going to read
- removed old makefiles used by ant.cpp
- improved CMakeLists.txt (compatible with ilcsoft v01-11)
  - reduced build time of lcioDict library to 50%
- bug fixes:
  - fix printout of TrackerHit (last element of cov matrix was missing(wrong))
  - fixed typo in CalorimeterHit documentation (CHBIT should be RCHBIT)
  - some protections against NULL pointers (A.Sailer)
  - added missing fortran wrappers from cfortran.h for CalorimeterHit ( CPPFORT/lccah.h )
  - fixed several issues found by coverity (<https://coverity.cern.ch>)
  - fixed bug with unsorted indices for direct access (java) - to be reviewed

=====  
GEAR : v01-00  
=====

- Added and FTDLayerLayout by J. Duarte Campderros, IFCA
- renamed VXDParameters and VXDLayerLayout to ZPlanarParameters and ZPlanarLayerLayout which can also be used for SIT and SET
  - changes should be backward compatible through typedefs
- added get/setSITParameters() and get/setSETParameters() to GearMgr
- added simple material description to Gear
  - allowd to have materials of the form  
SimpleMaterial( name , A , Z, density, radLength, intLength )  
stored in the GearMgr
  - materials are referenced via their names (stored in user parameters)
- added constructors for double[3] and float[3] to Vector3D
  - could be used with LCIO classes, e.g.:  
gear::Vector3D pos( hit->getPosition() ) ;
- added src/test/testXMLCopy.cc (simple copy of xml files through GearXML)

```
=====
Marlin : v01-01
=====
```

- Added dynamic command line options (see example by calling Marlin -h)
- Added checking of duplicate processor entries in the steering file

Bug fixes:

- skip linking of lcio libraries through lccd (cause warning on mac osx)
- examples/mymarlin/src/MyProcessor.cc: protected against non-existing collections
- made gcc 4.4.3 compliant

- removed compiler warnings

=====  
KalTest : v01-02  
=====

- introduced bounded planes (D.Kamai)
- examples provided in hybrid example to demonstrate use of bounded planes in VXD and FTD
- added extra Transport method to transport directly to a measurement layer and modified the previous Transport to site method to use this method. No change in previous functionality, only additional functionality added

=====  
KalDet : v01-02  
=====

- Updated for ILD detector reconstruction with KalTest and MarlinTrk
- ILD detectors added in new ild directory.
  - VXD, TPC, FTD(both simple disk and petal based)
  - SIT is currently only a placeholder using simple planar design
- common hit and plane classes added to ild/common subdirectory
- uses UTIL/ILDConf.h from Icio for detector element encoding
- Added ILDParallelPlanarMeasLayer for VXD and SIT, which has closed solution for calculating the crossing point of a helix. Constructor ensures that it is parallel by supplying only r and phi, which are used to calculate the normal and centre for planar surface.

=====  
Mokka : mokka-07-07  
=====

## What is new in this Mokka release

=====

- I. Improvements of the FTD driver
- II. The step position saving for the SDHcal
- III. Improvements of the AHCAL Barrel
- IV. Improvements of the TPC implementation
- V. Definition of three new detector models
- VI. Further improvements of Mokka implementation

=====

Please note that:

This Mokka release co-works with Geant4 9.4 (and 9.4.patch01) and CLHEP 2.1.0.1

It was tested against GEAR v01-00 and LCIO v02-00, gcc 3.4.6 and 4.1.2, SL4 and SL5.

=====

### I. Improvements of the FTD driver

Thanks to Jordi Duarte new improvements are available in the FTD driver:

\* Included the z-staggered design of the petals.

- The driver is compatible with both designs (staggered and turbine-blade).

The database is containing a new field in the "disks" table:

petal\_support\_zoffset which define the stagger distance in z for the petals.

Database ftd08 : z\_offset=0.0,

petal\_inclination\_angle\_support = 4.0 ---> turbine-blade design

Database ftd09 : z\_offset=1.5,

petal\_inclination\_angle\_support = 0.0----> staggered design,

The FTD version for the new models (in the "ingredients" table) is the SFtd10, but for optimization studies the SFtd09 subdetector (SFtd06.cc and ftd08, turbine-design) can be used.

- \* Included the GEAR persistency using the new FTDPParameter and FTDLayerLayout classes
- \* Fixed some geometric bugs (z-position of the sensor was not compatible with the mechanical design)
- \* Changed the FTD codification in the CGA/EncoderSi class and propagated the change in the Tesla/TRKSD\_FTD00 sensitive class.

## II. The step position saving for the SDHcal

In ' Detailed Shower Mode ' (while using the command /Mokka/init/lcioDetailedShowerMode true ), the step position in the SDHcal sensitive volume is written to the LCIO file, together with the secondary PDG.

## III. Improvements of the AHCAL Barrel

Thanks to Shaojun Lu, several improvements are available for the HCAL Barrel:

1. Put absorber and cassette into one: 20mm
2. Remove cassette box from driver for Barrel layer: 0.5mm
3. Added Hcal\_fiber\_gap back into Barrel layer: 1.7mm
4. Remove layer\_support\_length from driver for Barrel layer: 0mm
5. Update driver for lateral\_structure\_thickness, it only in the middle between the two modules: 15mm
6. Remove lateral\_structure\_thickness between Barrel and electronics service. There could not be steel wall between readout electronics board and HBU electronics board.
7. BarrelChamberSupportTrap has been removed from both header and source code.
8. Barrel module has been placed with rotation, to keep the "lateral structure" in the middle for both modules.

- 9. Update chamber\_z\_offset from "0" to "-Hcal\_lateral\_plate\_thickness/2.0".
- 10. All the relative x/y/z length and shift has been updated with the updated modification and parameters.
- 11. The I-coordinate calculation has been updated inside "SHcalBarrel.cc".  
Due to the rotation of Module inside ScHcalSc03, rotation for I-coordinate has been updated too inside "SHcalBarrel.cc".

#### IV. Improvements of the TPC implementation

Thanks to Steve Aplin several improvements are available for the TPC driver:

- the material of the TPC cathode grip ring was changed to carbon, and
- its dimension in z was reduced from 30 mm to 15 mm.

#### V. Definition of three new detector models

Three new ILD models were created, which contain the FTD, TPC, AHcal above mentioned improvements and - one of the models - the SDHcal.

The three models are:

- ILD\_01\_pre02 - AHCal and Si-Ecal
- ILD\_01\_SDH\_pre00 - SDHCal and Si-Ecal
- ILD\_01\_SciW\_pre00 - AHCal and Scintillator-Ecal

#### VI. Further improvements of Mokka implementation

Thanks to Frank Gaede, we have changes in the Coil drivers' Gear parameters and the rename of CellID to CellID1 for SimTrackerHit.

```
=====
MokkaDBConfig : v03-02
=====
```

includes latest dump of the Mokka database  
mokka-07-07-dbdump.sql.tgz (taken on 2011.09.23-15:00)

added mysql-admin.sh script to mysql-local-db-setup.sh which allows to shutdown and startup database.

=====  
CED : v01-03  
=====

- New features (H. Hoelbe, DESY):
  - \* Save/Load settings option (saves zoom, background color, view position, etc in ~/.glced)
  - \* Detector components have now layers and layer description
  - \* Increased the number of available layers
  - \* Add side view projection (press 'S' to toggle)
  - \* Add front view projection (press 'F' to toggle)
  - \* Add perspectival option (objects far away appears smaller)
  - \* Its possible to toggle the visible of the axes (select it from the menu)
  - \* Show frames per seconds (select it from the menu)
  - \* Enhanced fisheye view by adding zoom levels when view mode toggles, or when switches to side, front or reset view
  
- \* Add a new optional draw style for the detector, called "New view" includes:
  - \* Detector cuts: longitudinal (press and hold 'z' or 'Z') and transversal cuts (select it from the menu)
  - \* Its possible to change the value of the transparence (menu)
  - \* in 3d-view also added a line model for better identification
  - \* Inner edges can be different from outeredges, and the inner cylinder can be rotated with an other angle as the outer cylinder shape
  - \* Its possible to set a flag in GEO\_Tube objects, so that they only get drawn in new view, see marlinCED.c for an example
  
- \* Fixes/Intern stuff (H. Hoelbe, DESY):
  - \* Fixed mouse-wheel functionality for Ubuntu
  - \* Changed CED from a c to a c++ project
  - \* Change order of drawing items
  - \* Removed zoom level reset from side and front view
  - \* Update deepfilter and blend function
  
- added OpenGL library paths to the rpath list for preventing LD\_LIBRARY\_PATH (set by geant4 env init script) to overwrite the opengl system libraries used

in the linking of glced with the ones from Mesa (installed in afs), which might cause a dramatic performance penalty

=====  
CEDViewer : v01-03  
=====

- added new TrackerRawViewer
  - displays raw data from LCTPC large prototype
  - needs CEDViewer to be build with MarlinTPC and LCCD
  - see example/viewTrackerRawData.xml and example/reconstructLCTPCRawData.xml
  
- CEDViewer
  - added new steering parameters "DrawHelixForTrack" and "DrawDetectorID"
  
  - fixed display of ReconstructParticles (did not respect the marker flags - patch provided by T. Tanabe)

=====  
MarlinUtil : v01-04  
=====

New features:

- MarlinCED:
  - added more subdetectors: FTD, LHCal, LCal, Coil, Yoke, SIT
  - changed detector colors to match the default colors used in Mokka
  - detector components have now layers and layer description
  - Cylinder: inner edges can be different from outeredges, and the inner cylinder can be rotated with an other angle as the outer cylinder shape

- made compatible with CLIC (no LHCa1 and 11 FTD layers):  
LHCa1 is now optional and the FTD is drawn dynamically from gear....

To use the new CED features use CED\_GeoTube instead of CED\_GeoCylinder.

- header files now installed into subdirectory (marlinutil) to avoid name clashes  
still backwards compatible
- moved classes FPCCDData and FPCCDPixelHit from MarlinReco  
to MarlinUtil to avoid dependency of Overlay on MarlinReco

#### Fixes:

- MarlinCED:
  - old ced\_hit\_id is now deprecated (the old ced\_hit\_id used one int to store type and layer number).  
The new one have one argument more, (layer and type are now 2 fields)
  - Added try/catch blocks to prevent gear::UnknownParameterException, while accessing undefined detector components.
  - fixed drawGEARDetector
  - fixed length of VXD detector ( had used half size only !)
  - allow to toggle visibility of VXD
  - made compatible with new gear (used for ILD\_01):
  - use optionally FTDPParameters
  - use size of SIT arrays (will have to be changed to ZPlanarLayout)
  - optionally use new coil parameter names

=====  
MarlinReco : v00-30  
=====

- General
  - Enabled build without cernlib (option MARLINRECO\_CERNLIB=off) - e.g. for macos-64bit sub packages (BrahmsTracking, Satoru, MarlinKinfitt,...) that need cernlib will not be built
  - Removed cmake configuration files to avoid linking/compiling against MarlinReco

- Analysis
  - Removed MarlinKinfitt processor
  - RecoMCTruthLinker
    - Modification to allow Bremsstrahlung photons to be written to Skimmed list (off by default)
- Tracking
  - Use ILDCellID0 from Icio Util
  - Removed deprecated setIlsReferencePointPCA
  - New processor FPCCDClustering.cc was added ( Daisuke Kamai )  
(FPCCDData and FPCCDPixelHit classes are in MarlinUtil )
  - TPCDigiProcessor
    - Changed exception to warning for the case of points with the same x-y coordinates being passed to getPadTheta and getPadPhi.  
Corrected the check function which was previously using float == float, to use fabs(float-float) < tolerance.  
In this case tolerance has been set to a tenth of a micron

```
=====
MarlinPandora : v00-06
=====
```

- Fixed problem where minimum number of ftd hits obtained from xml steering file was assigned to wrong variable.
- Updated to account for new monitoring location and added ability to link against PandoraMonitoring library.
- Updated to use new pandora API for pfo extraction.
- Updated to reflect recent changes to pandora APIs, including typedef changes and use of CaloHitLists, rather than OrderedCaloHitLists.
- Removed PathLengthCalculator class, now unused (no longer need to specify path lengths from interaction point).
- Added new PandoraSettings files following addition of new PhotonReconstruction algorithm to Pandora fine granularity content library.
- Added xml files containing photon likelihood PDFs, for both one and nine bins of reconstructed cluster energy.
- Updated PandoraSettings files to use new functionality of soft cluster merging and isolated hit merging algorithms.
- Updated to handle "Ã la Videau" geometry more elegantly.

=====  
PandoraPFANew : v00-07  
=====

Moved PandoraMonitoring to PandoraPFANew repository and updated CMake and standalone makefiles accordingly.

Addressed warnings produced when compiling without monitoring.

Restructured pandora manager classes, separating common functionality into Manager base class.

Separated common functionality of CaloHit and Track managers into InputObjectManager class.

Separated common functionality of Cluster and Pfo managers into AlgorithmObjectManager class.

Exposed CreateTemporaryListAndSetCurrent APIs to enable Cluster or Pfo creation without need for nested algorithms.

Made GetPfoList API available to client applications.

Added new PfoPreparation and PfoCreationParent algorithms.

Updated VisualMonitoring algorithm to deal with multiple pfo lists.

Modified CaloHit manager to allow for division/fragmentation and re-merging of CaloHits, including during reclustering operations.

CaloHit availability monitoring updated and moved from CaloHit helper to CaloHit manager.

CaloHit manager now saves CaloHitLists, rather than OrderedCaloHitLists; APIs now work with CaloHitLists, resulting in algorithm changes.

Added new types of CaloHit: Rectangular CaloHit (existing type) and new Pointing CaloHit, with dimensions specified in delta(eta) and delta(phi).

Implemented GetCellCorners and GetCellLengthScale functions for Rectangular and Pointing CaloHits.

Updated file reader/writer classes following changes to CaloHits and interfaces.

Added new PhotonReconstruction algorithm, for clustering and identification of photons before standard pandora reconstruction.

Added implementation of multivariate/pid analysis for photon identification, supporting separate PDFs for bins of reconstructed cluster energy.

Added ability to read/write pandora histograms from xml.

Added new interface for use when calculating transverse shower profile. Function can return list of identified shower peaks.

Renamed functions for calculation of longitudinal and transverse shower profiles. Calculations are accessible via ParticleIdHelper.

Updated isolated hit merging algorithm and soft cluster merging algorithm to handle (neutral) clusters from multiple specified cluster lists.

Added new algorithms: MergeSplitPhotons and BeamHaloMuonRemoval.

Replaced efficiency monitoring algorithm (no longer maintained) with simple algorithm for

evaluating efficiency of particle id for single particle samples.

Added ECAL-only option to PerfectClustering algorithm.

Changed behaviour of V0PfoCreation algorithm in case where predicted mass squared is negative.

Added option to visual monitoring algorithm allowing use of new hit colour scheme, which uses continuous palette to reflect energy of hits.

Altered behaviour of MManager::GetMCParticleList function, returning success for case where no mc particles have been created.

Removed unnecessary const qualifier from pointer returned by value in MCParticleHelper.

Corrected update of inner and outer pseudolayers when calo hit is removed from a cluster.

Removed possibility of deleting a cluster that is currently part of a pfo.

=====

PandoraAnalysis : v00-03

=====

Added new PandoraPFACalibrator processor, courtesy of A. Lucaci-Timoce.

Added example steering file for PandoraPFACalibrator.

Updated PandoraAnalysis example steering file.

Updated to deal with missing trees more gracefully.

Updated to deal with zombie root files more gracefully.

Included <cstdlib> header to allow use of atoi and atof functions under gcc4.3 onwards.

=====

Overlay : v00-11

=====

- New processor (OverlayEvents) to merge a number of events in a LCIO file into one event.
- First version of background overlay processor for FPCCD (FPCCDOverlayBX) (D.Kamai)
- Added dependency to MarlinUtil (needed by FPCCOverlay )

=====

CondDBMySQL : ILC-0-9-5

=====

- bug fixes in MySqlTypes.h: (P. Klenze)

- activated mysql reconnect option
- fixed uninitialized variable theLog

=====  
ilcinstall : v01-12  
=====

new modules added:

- marlintrk.py

changes in modules:

- root.py: fixed bug when configuring with gsl (LD\_RUN\_PATH needs to be set in order to link against the correct gsl libraries)
- kaltest.py: added MarlinUtil dependency to KalDet
- marlinreco.py : made cernlib optional
- cedviewer.py: added optional dependency to LCCD and MarlinTPC
- marlinpkg.py: added cmake find support for MarlinTPC
- marlin.py: added tests (ctest)
- lcio.py: removed ant support
- overlay.py: added dependency to MarlinReco

general changes and bug fixes:

- added ILCSOft.cmake.env.sh to initialize environment for developers
  - any package can be configured by simply typing: cmake ..
  - the following order is used to find packages:
    1. CMAKE\_PREFIX\_PATH environment variable defined in ILCSOft.cmake.env.sh or by setting/exporting into current shell
    2. CMAKE\_PREFIX\_PATH cmake variable defined in ILCSOft.cmake or using cmake's -D command line argument

3. manually defined <PKG>\_DIR variables using cmake's -D command line argument

1. has the lowest precedence and 3. the highest, i.e. <PKG>\_DIR variables override any previous CMAKE\_PREFIX\_PATH variables.

NOTE: This order of precedence only applies if variables are not already stored in CMakeCache.txt !! In order to clear the cache erase CMakeCache.txt or the build directory.

- removed backwards compatibility for older CMakeModules

- baseilc.py: added logfile info in case of error (by P. Klenze)

- baseilc.py: added -dev and -exp suffixes to lookup versions in branches instead of tags

- marlin.py: need to unset MARLIN\_DLL for running tests (MARLIN\_DLL could otherwise include a plugin which has not yet been built)