
Subject: iLCSoft release v01-17-06
Posted by [gaede](#) on Tue, 19 Aug 2014 18:43:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new developers release of ilcsoft is available: v01-17-06.

A lot of fixes, changes and updates are contained - please refer to the release notes below.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-17-06
[<https://svnsrv.desy.de/websvn/wsvn/General.ilctools/ilcinstall/tags/v01-17-06/?op=dl&isdir=1&rev=3051&peg=3051>]

As usual the reference installations in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17-06      # SL5 64bit  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-06      # SL6  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc46_ub1204/v01-17-06  # Ubuntu 12.04 64bit
```

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

The ilcsoft team:
Yorgos, Christoph and Frank.

```
#####  
#  
# iLCSoft release - v01-17-06  
#  
#####
```

```
=====
```

Changes in v01-17-06 w.r.t. iLCSoft v01-17-05

```
=====
```

Introduces 3 new packages:

=====
DDSim v00-02
=====

by F.Gaede et al.

- Linear collider simulation models based on DD4hep and Geant4

=====
Physsim v00-01
=====

by J.Tian

=====
GDML v03-01-01

LCDD v04-00-00

SLIC v04-00-00
=====

-SLIC MCParticle handling code was almost completely rewritten.

The output MCParticle collection is now in the proper order, with generator primaries listed first and simulation particles listed after these.

These changes have been extensively tested over the last several weeks.

Overall event processing runtime performance should now be about 10-15% faster, too.

=====
LCIO v02-05
=====

- Create Maven dev branch (J. McCormick)

- Update pom with new deployment information (J. McCormick)

- Turned LCIO_GENERATE_HEADERS off (generation of headers with java is broken)
(F.Gaede)

=====
Gear v01-04-01

=====

(I.Rubinsky)

- Merge with telPlanes branch: introducing new TrackePlanes geometry classes
 - Add set methods to SiPlanes and TrackerPlanes classes
 - Adding an TrackerPlanes example geometry
 - Introduce a redundant method for backward compatibility to old code using SiPlanes interface
 - Patching an interface function for backward compatibility of the SiPlanes geoemtry layout
- Fixed bug that extents were calculated before the test for identical coordinate systems was performed (M.Killenber)

=====

MarlinTrkProcessor v01-11

=====

- Added processor for VXD tracking using a cellular automaton algorithm based on mini - vectors (Y.Voutsinas)
- FPCCDTracking code: etIntersectionEasy and getIntersectionEasyTest are modified (call by value -> call by reference.), code became more readable (T. Mori)

=====

KiTrack v01-05

=====

- extending IHit class to cope with mini-vectors (Y.Voutsinas)
- Importing a first set of criteria for a mini-vector based CA (Y.Voutsinas)
- fix for ROOT 5.34.18, calling of TVector3 Angle fcn (Y.Voutsinas)

=====

KiTrackMarlin v01-05

=====

- Implementation for a Cellular Automaton VXD tracking based on mini - vectors (Y.Voutsinas)

=====
MarlinReco v01-10
=====

- Utility for studying pixel occupancy (T.Mori)
- Some modifications for pair-BG study are done (T.Mori)

- initial import of a quark/jet matching algorithm into RecoMCTruthLink (B. Hermberg)

- SDHCAL digitizer : replace all vectors of steps by vectors of StepAndCharge (G.Grenier)
- SDHCAL digitizer : add use of induced charge to select kept steps (G.Grenier)

=====
MarlinUtil v01-08-01
=====

- draw all layers for SIT and SET (F.Gaede)
- draw two layers per double layer (F.Gaede)

=====
DD4Hep v00-10
=====

Frank Gaede

- fixed version numbers and release notes for release v00-08 and v00-09
(both have the wrong version number for the libraries 0.7 !)
- added ./doc/prepare_a_release.howto
- added new method to BitField64: setValue(unsigned lowWord, unsigned highWord)
- added some simple unit tests for BitField64

2014/08/15 Markus Frank

- DDG4 Add possibility for user hook to the particle handler + Data member to store additional particle data for the output stage.
- Allow to display MC particles in DDEve
Add small help file describing how to start DDEve.
doc/DDEveManual.pdf
- Update CLICSiD Example.

v00-09

2014/08/07 Markus Frank

-
- DDG4: First version to support MC truth in DDG4 including particle filtering to optimize the size of the MC record.
 - DDG4 fix SimpleCalorimeter sensitive action and properly support hit aggregations.
 - DDEve smaller modifications to support DDG4IO if DD4hep was built with the Geant4 option ON.

| v00-08 | eighth beta release ...

- Some minor fixes:
 - made compatible with older geant4 versions (9.5)
 - add Bitflag to store CellID1 in SimCalorimeterHit collections
 - fix position conversion from Geant4 to ROOT
 - add cellID determination to SensitiveAction
 - ...

Andre Sailer, 2014-07-17

Unify cmake option variables, small cmake corrections
Change options `_WITH_` to `_USE_`
Print Warning that variables with `_WITH_` are deprecated
Change Definitions to `_USE_` as well
Add `REQUIRED` to `find_package geant4` and `xercesc` if they are turned on

Updated documentation
Updated ILDExDet example

Christian.Grefe, 2014-07-15

made DDSegmentation optionally a stand-alone package
create DDSgementationConfig.cmake when build as part of DD4hep

Markus Frank, 2014-07-02

- add LCIO conversions from DDSim - sensitive detectors
can now simply instantiate LCIO Sim hits

2014/06/30 Markus Frank

- DDG4: Support user (context-) extensions for Geant4.
User contexts are available for events and runs. The contexts
are accessible from the Geant4Context member of each Geant4Action,
the base class for all processing entities.
-- Adding a context:
context()->event().addExtension(pointer_to_object, typeid(TYPE), 0);
or
context()->event().addExtension<TYPE>(pointer_to_object);
In the first case the object is NOT DELETED at the end of the event.
In the second case the object is deleted.
-- Accessing the context:
TYPE* ctx = context()->event().extension<TYPE>(false);
The boolean argument indicates if an exception should be thrown
if the extension is not present ie. here no exception.
- DDCore: adjust a few print statements.

2014/06/27 Markus Frank

- Separate XML loading from LCDD implementation.
- New package: DDEve: a bit more sophisticated TEve specialization
To start use examples/CLICSiD/compact/DDEve.xml
DDEve can interface currently to ROOT files created by DDG4.
An example data file is provided as well in:
examples/CLICSiD/compact/CLICSiD_events.root
Documentation is lacking a bit....
start the display with:

\$> root.exe \${DD4hepINSTALL}/examples/DDEve/DDEve.C

2014/06/19 Frank Gaede

- replaced TGeoUnits.h by DD4hepUnits.h
- renamed namespace tgeo to dd4hep
- > users need to replace tgeo::mm by dd4hep::mm

2014/06/12 Frank Gaede

- changed default unit for angles to be radians (tgeo::rad = 1.)
- enforce usage of units from TGeoUnits.h in xml expression evaluators
- convert to degrees for angles in TGeoShapes constructors
- NB: there is one inconsistency left here: angles returned from TGeoShapes are already in degrees, this is the one case where a quantity returned from DD4hep/TGeo does not have the default units - thus one should not write
double phi = coneSeg.Phi1() / tgeo::rad ;
but rather
double phi = coneSeg.Phi1() * tgeo::deg ;

2014/06/03 Markus Frank

- Provision for ROOT persistency for DD4hep detector descriptions:
Create Cint dictionary for DD4hepCore by default when building the library. For the time being the area of DDSegmentation is left out, since these objects require changes due to the handling of references and template specializations. Besides these, DD4hep detector descriptions can be saved and read-back directly to/from ROOT files - which may improve a lot the startup time of processes.
- The DDG4 dictionary is created by default when building the library. When importing DDG4 from python only the DD4hepCore DD4hepG4 libraries must be present. It should no longer be necessary to compile the necessary AClick on the fly.
- Investigations on wirframe drawing:
Wireframes are a property of the scene graph. You can toggle to wireframe using the 'w' key on the keyboard. Other representations are

'r': Solid view

't': Solid + wireframe view.

This information is accessible from the help menu of the ROOT OGL widget.

----- 2014/05/28 Markus Frank

| v00-07 | seventh beta release ...

2014/05/28 Markus Frank

- Several fixes and improvements in DDG4
- Cache output levels in Geant4Kernel object
- Add shell handler to measure energy escape e.g. in calorimeters
- Fix bug in Polycone shapes concerning angles.
IMPORTANT: If you forked the repository examples/CLICSiD your PolyconeSupport does not work anymore, because the angles were given in degrees. They should be given in radians!

2014/05/21 Markus Frank

- Fix material creation from XML
JIRA bug: DD4hep - DDFORHEP-4
<https://sft.its.cern.ch/jira/browse/DDFORHEP-4>
using <composite> in material xml files results in wrong material properties
- Fix JIRA bug: DD4hep - DDFORHEP-3
<https://sft.its.cern.ch/jira/browse/DDFORHEP-3>
Recursive assemblies result in error in TGeo geometry
Bug results in errors when closing the geometry like:
"Error in <TGeoVoxelFinder::SortAll>: Wrong bounding box for volume SIT_assembly"
- Provision to store survey constants to the DetElement as an extension for Alignment calculations (priv. comm. C.Parkes)
- Fix bug in expression evaluation of angles.
All detector constructors receive angles in "rad". No correction from degree to rad should be necessary.

2014/05/21 Frank Gaede

- add MaterialManager class providing
 - access to materials at any point or on straight

- line between two points
- creation of material with averaged properties (A,Z,rho,x0,Lambda)
- added utility print_materials to print material properties along a straight line between two points including integrated radiation and interaction lengths (useful for debugging geometries and materials)
- use averaged material for Surfaces where the thickness extends beyond the volume boundaries
- introduced new simple data class MaterialData for this

*** known issues

- ** materials don't work for detectors with assemblies in assemblies as the TGeo navigation dose not seem to work:
Error in <TGeoVoxelFinder::SortAll>: Wrong bounding box for volume SIT_assembly
-> ROOT bug or feature ?
- ** using <composite/> in compound materials results in incorrect material properties see ILDExDet/compact/materials Polysterene as example

2014/05/06 Frank Gaede

- DDSurfaces/DDRec:
 - added ICylinder and CylinderSurface to provide access to cylinder radius for the tracking
- teveDisplay:
 - draw sensitive surfaces in red and helper surfaces in gray
 - protect against missing Icio input file
- reduced printout from VolumeManager (changed INFO to DEBUG)

2014/04/05-2014/04/29 Frank Gaede

- Implementation of Surface classes for track reconstruction
 - abstract interfaces in DDSurface: ISurface and IMaterial
 - implementation in DDRec: Surface, SurfaceManager, DetectorSurfaces
 - basic functionality exists:
 - measurement surface vectors u,v, normal and origin
 - check if point is inside the bounds of the corresponding volume
 - currently the material is taken to be the one from the volume
 - > to do: average material if surface thicknesses extend beyond the

volume for comound materials

- added drawing of surfaces and (Sim)TrackerHits and (Sim)CalorimeterHits to teveDisplay (link lcio file to teve_infile.slcio)
- added some surfaces to ILDExDet: VXD, SIT, TPC
- added test_surfaces to ILDExSimu (fails currently for some hits that are not created on the surface ...)
- added some assemblies to ILDExDet to better structure the geometry hierarchy

2014/04/28 Markus Frank

-
- First attempt to understand fully the handling of recursive assemblies from the 2014/04/25 was apparently with limited success. Now they should work!

2014/04/25 Markus Frank

-
- Moved boost::spirit parsers to DDCore, since these parsers are also used in other packages such as DDCond and DDAlign - whenever strings have to be converted to objects. Consequently, the boost spirit code fragments were removed from DDG4.
 - Improved and more correct type checking when assigning handles
 - Explicit implementation of the object extension mechanism to be used through inheritance or aggregation rather than having seperate and nearly identical implementations for various objects.
 - Improved definition of basic objects describing Conditions and Alignment.
 - Improved implementation of DDAlign, the alignment support. Not to be used currently - experimental package.
 - Initial implementation of DDCond the package supporting the handling of conditions. Not to be used currently - experimental package.
 - Geant4Converter: I convinced myself, that the bug in handling assemblies of assemblies is resolved. To be tested though.

2014/02/07 Frank Gaede

-
- added DDGear, support for interfacing to gear for backward compatibility
 - lives currently in ILDExDet (should become (optional) part of core DD4hep)
 - users need to GearParameter objects as extensions to the DetElement
 - > see ILDExTPC_geo.cpp, ILDExVXD_geo.cpp, ILDExTPCSIT_geo.cpp as examples
 - program convertToGear creates gear xml file from compact file (only if GearParameter objects are avialble as extensions)
 - added test_convert_ild_gear

- fixed geometry in ILDExVXD_geo.cpp, ILDExTPCSIT_geo.cpp wrt. to position of support and sensitive ladders

=====
MOKKA mokka-08-04
=====

(Shaojun Lu)

- Updated Ahcal Barrel Sensitive detector driver && endcap scalable driver for ILD_o1_v06.
- Added two optional parameters to allow the low granularity layers configuration for optimisation studies. Both Ahcal Barrel and Endcap SD granularity can be modified at Mokka runtime by setting the two parameters in Mokka steering file.
- Updated the AHCAL driver for writing out the layer cell granularity information into gear file.
- Fixed a wrong definition: $I = 0$ should be calculated by using fraction cell size, too. Not only cell center.
- Updated Ahcal geometry driver which used in ILD_o1_v06 to allow user to set a low granularity region.

The layer count from 1 to 48. The low granularity layer use the same number rules.

- Updated calculated parameter: Hcal_module_radius, backward compatible to DBD, and scalable with Hcal_nlayers for same outer shape

(F.Gaede)

- optionally create xml files
 - with all database queries from subdetector drivers
 - files are created in directory './mokka_xml_dump/' if it exists - no files are created if directory does not exist
- (un)/comment `#define DUMPXML` in MySQLWrapper.cc for (de)activating this feature

=====
LCTuple v01-03-01
=====

- add an example for creating an lctuple w/ simhits (F.Gaede)

=====
KalTest v01-05-04
=====

- Fixed a field direction bug (Bo Li)
- Slightly modified the track model class to make the straight track work; Straight track generation is the example ct was implemented. (Bo Li)

=====
KalDet v01-13-01
=====

- Fix for ROOT 5.34.18 (TVector3 c'tor) (F.Gaede)

=====
GBL v01-16-04
=====

- Added the removal of trailing whitespaces (C.Rosemann)
- cpp: some shadowing resolved (C. Kleinwort)
- cpp,py: binary files with `_double_` values implemented (C. Kleinwort)

=====
Overlay v00-14
=====

- An option to remove VTXPixelHits Collection is added. (A.Miyamoto)

=====
MarlinTPC v00-17
=====

A.Munnich, P.Hayman, C.Kleinwort, F.Mueller, C.Rosemann, M.Rogowski, A.Shirazi, A.Deisting, A.Chau, K.Zenker, Ch.Krieger, Bo Li

- Various bug fixes
- And added functionality

(M.Rogowski)

- Added: TimePixHoughTransformNormalProcessor. Track finding using a Hough transformation and normal parameterisation for the straight lines.
- Added TimePixPPHoughTransformProcessor
- Added: TimePixTrackEventDisplayProcessor
- Added TimePixRandomHoughTransformProcessor
- TimePixWindowedTrackFinderProcessor: Modified TimePixRandomHoughTransformProcessor, which was a randomized Hough Transformation, to a windowed randomized Hough Transformation. These two processors are now so different, that the windowed randomized Hough Transformation gets its own processor.

(A.Munnich)

- Analysis processor for the hit efficiency and corresponding root plot script.
- Work on simulation chain processors:

GEMProcessor: Added the possibility to use a polya function in the GEM gain instead of an exponential distribution.

TPCElectronicsProcessor: Added a base class ADCPulse and derived the Gaussian shaping and a new implementation for the ALTRO electronics from it. With a steering parameter the processor can switch between the different electronics implementations.

The new structure should make it easier to add other shaping pulse classes.

ChargeDistributionProcessor: Activated the foreseen usage of longitudinal diffusion generateElectronsForLP1: increased the beam spread. Especially in z weird behavior could be observed in one shoots almost always at the same position with respect to the time binning.

(F.Mueller)

- Analysis processor for PRF calculation and corresponding root plot script.
- Improved memory performance of analysis processors

- Implemented a new method for millepede output.

(Ch.Krieger)

- Implemented XrayEventDisplayProcessor to produce event displays of TimepixXrayObjects eg after a likelihood based discrimination
- New processor to apply different likelihood cuts for different energy bins to XrayObjects: XrayLikelihoodCutProcessor
- Added processor to remove dead and noisy pixels from rawdata collections (based on TimePixZeroSuppressionProcessor) -> TimePixRemoveMaskedPixelsProcessor

(P.Hayman)

-Micromegas-Related Update

- Added MMAdvancedCalibration Processor and MMRefitTool to better calibrate the MM PRF
- Added a calibration option and new PRF formatting to MMHitFinderProcessor, and changed the definition of the hit charge to be consistent with the GEMs
- Adjusted MMCalibrationProcessor to work with new PRF (more or less; older scripts should still work)
- Added a constant-Z option to MMHitTimeCorrectionProcessor
- Added calculation of chi2 to TrackFitterLinearRegression.cc and TrackFitterSimpleHelix.cc
- Added a processor to calculate Bias

(K.Zenker)

- Add a processor to test the TPCDistortionsMap

(C.Kleinwort)

- FHT processor added
- Added first version of RowBasedFastHoughTransformationProcessor
- Various improvements and added functionality on track fitting, triplet finder and FHT processor

=====
MarlinPandora: v00-14
=====

TAG V00-14 FOR ILC SOFT USAGE

Update to reflect changes to PandoraContentApi.

TAG V00-13 FOR ILC SOFT USAGE

Update to reflect changes to PandoraSDK.
Config overhaul; add some new PandoraSettings files and update readme.

TAG V00-12 FOR ILC SOFT USAGE

Add hadronic energy non-linearity correction function.

=====

PandoraAnalysis V00-06

=====

TAG V00-06 FOR ILC SOFT USAGE

Calibrate binary now reads TChains, rather than a single TTree.

=====

PandoraPFANew: v00-16

=====

TAG V00-16

Updated daughter package versions.

TAG V00-15

Updated daughter package versions.

TAG V00-14

Updated daughter package versions.

TAG V00-13 FOR ILC SOFT USAGE

Updated daughter package versions.

=====
LCFIVertex v06-02
=====

- Fixed duplicate namespace in forward declaration (F.Gaede)

=====
ILCUtil v01-02
=====

- Updated the numbers for the CDash server; tests running next night -- see if everything is set right (C.Rosemann)

