
Subject: new ilcsoft release v01-16-01
Posted by [engels](#) on Tue, 28 Aug 2012 12:19:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new release of ilcsoft (v01-16-01) is available.

This patch release is based on the previous major release (v01-16) used for running the DBD reconstruction.

This patch release includes fixes in the LCIO package (see Changelog below) and an updated version of Geant4 (9.5.p02).

Please refer to the Release notes below for more details.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-16-01.

Reference installations in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/i386_gcc41_sl5/v01-16-01 # SL5 32bit  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-16-01 # SL5 64bit
```

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

Cheers,

Jan, for the iLCSoft team.

Release Notes for v01-16-01 and v01-16

```
=====  
  
Changes in v01-16-01 w.r.t. iLCSoft v01-16  
  
=====
```

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

Geant4: 9.5.p02

=====
LCIO: v02-03-03 / v02-03-02
=====

- fixed bug in TrackerHitPlaneImpl and TrackerHitZCylinderImpl with size of covariance matrix being 0 (no set to 6) that caused corrupted LCIO files when any hits of these types were written as "TrackerHit" types, e.g. in a new subset collection that is made persistent ...
this is no problem for the DBD standard reconstruction !
- changelog from v02-03-02:
 - added lcio_merge_files tool
 - fixed cmake 2.6 incompatibility introduced in v02-00-02

=====
Release notes from iLCSoft v01-16
=====

iLCSoft release for running the DBD reconstruction.

=====
Changes in v01-16 w.r.t. iLCSoft v01-15-03

=====
=====
MarlinReco: v01-05
=====

- BCalReco (A.Rosca)
 - in BCalReco.h reduce memory use: #define MAXNENTR 80000
 - introduced new variable EdepErr.
 - small changes to read new variable, EdepErr, from the background map.
 - fixed bug in SearchTowers().
 - write also empty BCal collections
 - fixed several programming flaws/reimplemented parts of code
- BCalTagEfficiency (J.List)
 - removed local copy of TDR background map, updated README and bcal_ild_05_v05.xml
 - some minor clean-up of print statements, comments etc

=====
MarlinTrk: v01-10-01
=====

- Added some more diagnostics, no change to algorithms.

=====
MarlinTrkProcessors: v01-09
=====

- FullLDCTracking_MarlinTrk
 - Use the Chi2 probability to remove badly fitted silicon tracks so that they don't pollute the TPC Si track merging.
 - Reject combinations of TPC and Silicon Tracks if more than a certain number of Silicon Hits, default 2, get rejected by the fit of the combined track.

=====
LCFIPlus: v00-05-02

=====

- Fixed type conversion error in track hits
- Fixed MC vertex information
- Added option to choose track hit ordering
- Modifications to track selection criteria involving subdetector hits
- Vertex-jet reassociation no longer throws exception to cope with cases where tracks are dropped such as when running kt algorithm before LCFIPlus

=====

KiTrack: v01-04

=====

- SegmentBuilder: added output of failed criterion
- SubsetHopfieldNN: set the activation threshold down to 0.5

=====

KiTrackMarlin: v01-04

=====

- KiTrackMarlinTools: added convenience methods for output of track-hit information

=====

ForwardTracking: v01-07

=====

- TrueTrackCritAnalyser: added pdg and theta angle to root output
- TrackingFeedbackProcessor:
 - + added possibility to use only hits from different layers for the nHit cut
 - + added cut for theta
 - + added chi2, Ndf and chi2prob to the reco tracks in the root output
 - + added a cut for failed tracks
- ForwardTracking:
 - + added a new quality indicator, especially for the simple subset and 3-hit tracks
 - + changed lastLayerToIP to 5
 - + changed the > in the chi2prob cut to >=

Subject: new iLCSoft release v01-15-02
Posted by [gaede](#) on Tue, 25 Sep 2012 17:51:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new release of iLCSoft (v01-15-02) is available.

The reconstruction software in this release
should be close to the final DBD reconstruction version.

Please refer to the Release notes below for more details.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-15-02.

Reference installations in afs are available at:

`/afs/desy.de/project/ilcsoft/sw/i386_gcc41_sl5/v01-15-02 # SL5 32bit`
`/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-15-02 # SL5 64bit`

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

The iLCSoft team.

=====
Changes in v01-15-02 w.r.t. iLCSoft v01-15-01
=====

=====
LCIO: v02-03
=====

- added new flag bit to simulator status word:
MCParticle::BITOverlay=23 and methods
MCParticle::isOverlay()
MCParticle::setOverlay(bool v)
- print as 'o' in LCTOOLS/Operators
- added new methods (in C++)
LCReader::getRuns(EVENT::IntVec & runs) and
LCReader::getEvents(EVENT::IntVec & events)
that return the run and event numbers in a given file
- to be used in randomly reading events for bg overlay
- implemented in SIOReader
- minimal testing in test_randomaccess.cc
- in Operators.cc:
 - print covariance matrix as lower triangle matrix for Track and TrackState
 - added GoodnessOfPID to dumpe of ReconstructedParticle
- changed the gnu jel dependency in pom.xml to 2.0.1 from 0.9
plus small change in SIODump.java
- added example cpp/draw_event: displays one event in
ROOT-UGL and writes ascii file
- made UTIL::LCIterator<T> compatibel with gcc 4.3.2
(#include <typeinfo>, reported by M.Killenber)

=====
Marlin: v01-04
=====

- added EventSelector processor
 - use Marlin -x and uncomment lines with EventSelector and <if>
and specify run and event number in MyEventSelector
- bug fix: call Global::EVENTSEEDER->refreshSeeds(evt) ;
also for modifyEvent() which is needed for BG Overlay
- prevent implicit copy c'tors and assignment operators for Processors

=====
MarlinReco: v01-04-01
=====

- fixed TPCDigiProcessor :
 - added bfield correction factor to point resolution ($4.0 / bField$)
- BCalRec (A.Rosca)
 - cleaned up code (removed unnecessary code)
 - introduced new default bg map file
 - enclosed code with `if(nHits > 0) {...}` in order to prevent warning about missing CellIDEncoder string
- fixed indentation of code (fg, using emacs defaults) (r4088)
No code change, i.e. same code as r4087 (use this for diffs)
- BCalTagEfficiency (J.List)
 - verified unchanged performance on Lol samples (modulo a bug fix); steering flag for map format added

=====
Clupatra: v00-09-01
=====

- in TrackCircleDistance::operator():
adapted to fix in KalTest where curler segments now have correct Z0 (typically close to the IP)
- do not merge segments if the z-positions of both their first hits is within 20 mm of the IP
- changed LCIOTrackConverter::operator() to use the first constrained fit (\approx 3rd hit) smoothed to extrapolate back to the calorimeter
- fixed issue in ClupatraProcessor::check() when no TPCTrackerHits are present (using LCIterator)

- moved local LCIterator to LCIO as UTIL::LCIterator
use this in all processors
- updated to use new PlanarDigiProcessor
(was: SimplePlanarTestDigiprocessor)
- fixed coverity issues: mostly uninitialized c'tor, some unchecked dynamic_casts and division by zero (nhit++)
- deactivate pick'n save feature
(#define WRITE_PICKED_DEBUG_TRACKS false)

```
=====
Overlay: v00-13
=====
```

- changes in Overlay processor:
 - updated to use MCParticle::setOverlay()
 - added method Overlay::readNextEvent() to properly read the next event - either with direct access (only one input file given) or with skipNEvents
 - fixed logic to ensure same event gets same overlay events - regardless of SkipNEvents parameter
- changes to OverlayBX processor:
 - switched to use Global::EVENTSEEDER mechanism for reproducible random numbers
- introduced namespace overlay in all classes
- removed using namespace from header files
- added propset Id

=====
MarlinTrk v01-10
=====

- General
 - Made Debug output more consistent
- MarlinKalTestTrack
 - Changed smooth() to do _kaltrack->SmoothAll(), previously _hitIndexAtPositiveNDF + 1
 - Corrected orientation regarding transporting inwards or outwards in propagate.
 - Fixed problem where initial covariance term kappa,tanL wrongly set as kappa,z0. Minimal impact expected. (Tino Calancha)
 - Ensure that hits which are rejected for reasons other than Chi2 cut are added to the list of outliers.

=====
KalTest v01-05-01
=====

- Fixed orientation for in/out transportation in TKalDetCradle, previously only inward transportation was correct.
- Ensure that material effects are treated correctly by only including material effects from present surface to destination surface.
- THelicalTrack:MoveTo modified to make sure the helix really moves to the new reference point and is not out by more than 2PI.

=====
KalDet v01-10
=====

- ild
 - Reduced the tolerance for the is on surface to 1micron in ILDCylinderMeasLayer.
 - Corrected sign of phi in CalcXingPointWith. ILDDiscMeasLayer, ILDSegmentedDiscMeasLayer, ILDParallelPlaneMeasLayer.
 - Added necessary mode check for determining intersection correctly based on the direction of travel, i.e. fwd or bwd mode =+1 or -1
- ILDConeMeasLayer, ILDDiscMeasLayer, ILDPolygonBarrelMeasLayer,

ILDSegmentedDiscMeasLayer, ILDParallelPlan
arMeasLayer.

=====
Release notes from iLCSoft v01-15-01
=====

iLCSoft release for running final tests before starting the DBD reconstruction.

=====
Changes in v01-15-01 w.r.t. iLCSoft v01-15
=====

=====
LCIO: v02-02
=====

- added new class LCIterator<T> for convenient loops over collections, e.g.:

```
for( LCIterator<Track> it( evt, "Tracks" ); Track* trk = it.next() ; ) {  
    std::cout << trk->getTrackState( TrackState::AtIP ) << std::endl ;  
}
```
- added small test/example to ./src/cpp/src/TESTS/test_tracks.cc
- Operators.cc: fixed scientific format for Track/TrackState
- changed TrackState (convenience) copy constructor argument type to reference instead of pointer

- fixed potential memory leak by adding Track's assignment operator (c++ rule of three)
- added LCIO_JAVA_USE_MAVEN option to build Icio.jar library using maven

=====
MarlinTrk: v01-09
=====

- General
 - Made Debug output more consistent
- IMarlinTrack
 - Moved constant definitions of return codes outside of the header file.
 - Added additional error return, for the case where no site are filtered when calling fit.
- MarlinKalTestTrack
 - Prefer translation over rotation of the trackstate early in the fit, when using simple helix initialisation
- MarlinTrkUtils
 - Use EVENT::TrackState in place of IMPL::TrackStateImpl where appropriate.
 - The TrackState for the initialisation can be at any reference point.

=====
MarlinTrkProcessors: v01-07
=====

General:

- Debug output has been made more consistent throughout.

FullLDCTracking_MarlinTrk

- Use existing tracks parameters when refitting.
- Updated for new Icio TrackState copy constructor taking const reference.

TruthTracker

- Updated for new Icio TrackState copy constructor taking const reference.
- Corrected Helix orientation for pre-fit.

RefitProcessor

- Corrected Helix orientation for pre-fit.
- Protect against missing truth relations.

=====
 Clupatra: v00-09
 =====

- apply a refit with larger maximum chi2 increment to tracks that have only a small number of hits used (IMarlinTrkFitter::operator())
- drop poor seed clusters with no hits added except in the very forward (inner) region
- clupatra_new: added optional argument TrkSystem* to addHitsAndFilter(); if given, the smoothed TrackState is used for backward extrapolation; => not currently used, i.e. the last filtered TrackState is used (this should give better results than the previous version that used the filtered TrackState at the 4th hit)
- don't set quality bit on tracker hit: ILDTrkHitQualityBit::USED_IN_FIT anymore, as this won't work on pre-digitized files (and is not unique wrt other pat rec code)
- added DebugDigiProcessor :
 takes TrackerHits from a collection of Tracks (e.g. previously written as ClupatraPoorTrackCollection) and writes them into new Collections (according to subdetID)
 -> see ./example/clupa_debug.xml
- added debug method printAndSaveTrack() to Clupatra:
 allows to store a collection of Tracks selected through picking in CED
- added doxygen documentation of parameters in ClupatraProcessor.h
- Updated for new Icio TrackState copy constructor taking const reference.

=====
CEDViewer: v01-06
=====

- removed TrackerRawViewer
 - lives now in MarlinTPC
 - CEDViewer no longer optionally depends on MarlinTPC nor LCCD

=====
MarlinTPC: v00-10
=====

- + added namespace pathfinder to package pathfinder, changed interface processor accordingly
- + removed deprecated files
- + added ced viewer extension for raw data as its own directory: resolves double dependency problem
 - ! breaks current ilcsoft installation with ced viewer -> use head version of CEDViewer
- + changed and cleaned CMakeLists.txt file accordingly

=====
PathFinder: v00-02
=====

- + introduced namespace pathfinder -- breaks backwards compatability (!)
- + removed obsolete files
- + minor cosmetic changes

Subject: new ilcsoft developers release v01-15-03
Posted by [gaede](#) on Sat, 29 Sep 2012 09:15:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new release of iLCSoft (v01-15-03) is available.

This is the final test release before the DBD production is started.
There are some important updates to the material description for the tracking - see release notes below for details.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-15-03.

Reference installations in afs are available at:

`/afs/desy.de/project/ilcsoft/sw/i386_gcc41_sl5/v01-15-03 # SL5 32bit`
`/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-15-03 # SL5 64bit`

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

The iLCSoft team.

=====
Release notes from iLCSoft v01-15-03
=====

iLCSoft release for running final tests before starting the DBD reconstruction.

=====

Changes in v01-15-03 w.r.t. iLCSoft v01-15-02

=====

=====

MarlinReco: v01-04-02

=====

- BCalTagEfficiency(J.List):
 - threw out interpolation between BeamCal cells, introduced optional write-out of background map
 - uses LCRelationNavigator now, cleaned up example steering
 - write out also empty collections
 - added 3.5T background map and up to date example steering for ILD00
- BCalReco (A.Rosca):
 - improved calculation of Cartesian coordinates from cell position given in terms of ring/pad number.
- RecoMCTruthLink:
 - exchanged ttricol and ctrlcol in makeSkim()
(will have no effect except code sanity)
- LDCCaloDigi/SimDigital (G.Grenier):
 - Bug fix to get output hits stored in output collection+increase speed of integration
- FPCCDDigi:
 - t.mori enable to difine pixel size for each layer.

=====

KalDet v01-11

=====

- ild (S.Aplin)

- final change to FTD and Beam-pipe material

Subject: new ilcsoft release v01-16-01
Posted by [engels](#) on Mon, 12 Nov 2012 11:46:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new release of ilcsoft (v01-16-01) is available.

This patch release is based on the previous major release (v01-16) used for running the DBD reconstruction.

This patch release includes fixes in the LCIO package (see Changelog below) and an updated version of Geant4 (9.5.p02).

Please refer to the Release notes below for more details.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-16-01.

Reference installations in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/i386_gcc41_sl5/v01-16-01 # SL5 32bit  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-16-01 # SL5 64bit
```

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

Cheers,

Jan, for the iLCSoft team.

Release Notes for v01-16-01 and v01-16

=====
Changes in v01-16-01 w.r.t. iLCSoft v01-16
=====

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

Geant4: 9.5.p02

=====
LCIO: v02-03-03 / v02-03-02
=====

- fixed bug in TrackerHitPlaneImpl and TrackerHitZCylinderImpl with size of covariance matrix being 0 (no set to 6) that caused corrupted LCIO files when any hits of these types were written as "TrackerHit" types, e.g. in a new subset collection that is made persistent ...
this is no problem for the DBD standard reconstruction !
- changelog from v02-03-02:
 - added lcio_merge_files tool
 - fixed cmake 2.6 incompatibility introduced in v02-00-02

=====
Release notes from iLCSoft v01-16
=====

iLCSoft release for running the DBD reconstruction.

=====

Changes in v01-16 w.r.t. iLCSoft v01-15-03

=====

=====

MarlinReco: v01-05

=====

- BCalReco (A.Rosca)
 - in BCalReco.h reduce memory use: #define MAXNENTR 80000
 - introduced new variable EdepErr.
 - small changes to read new variable, EdepErr, from the background map.
 - fixed bug in SearchTowers().
 - write also empty BCal collections
 - fixed several programming flaws/reimplemented parts of code

- BCalTagEfficiency (J.List)
 - removed local copy of TDR background map, updated README and bcal_ild_05_v05.xml
 - some minor clean-up of print statements, comments etc

=====

MarlinTrk: v01-10-01

=====

- Added some more diagnostics, no change to algorithms.

=====

MarlinTrkProcessors: v01-09

=====

- FullLDCTracking_MarlinTrk

- Use the Chi2 probability to remove badly fitted silicon tracks so that they don't pollute the TPC Si track merging.
- Reject combinations of TPC and Silicon Tracks if more than a certain number of Silicon Hits, default 2, get rejected by the fit of the combined track.

=====
LCFIPlus: v00-05-02
=====

- Fixed type conversion error in track hits
- Fixed MC vertex information
- Added option to choose track hit ordering
- Modifications to track selection criteria involving subdetector hits
- Vertex-jet reassociation no longer throws exception to cope with cases where tracks are dropped such as when running kt algorithm before LCFIPlus

=====
KiTrack: v01-04
=====

- SegmentBuilder: added output of failed criterion
- SubsetHopfieldNN: set the activation threshold down to 0.5

=====
KiTrackMarlin: v01-04
=====

- KiTrackMarlinTools: added convenience methods for output of track-hit information

=====
ForwardTracking: v01-07
=====

- TrueTrackCritAnalyser: added pdg and theta angle to root output
- TrackingFeedbackProcessor:
 - + added possibility to use only hits from different layers for the nHit cut
 - + added cut for theta
 - + added chi2, Ndf and chi2prob to the reco tracks in the root output

- + added a cut for failed tracks
- ForwardTracking:
 - + added a new quality indicator, especially for the simple subset and 3-hit tracks
 - + changed lastLayerToIP to 5
 - + changed the > in the chi2prob cut to >=

Subject: new ilcsoft production patch release v01-16-02
Posted by [rosemann](#) on Thu, 20 Dec 2012 16:17:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new patch production release of ilcsoft (v01-16-02) is available.

This release is based on the previous patch release (v01-16-01).

This patch release includes bug fixes in MarlinReco, a fix to KalTest and minor version upgrades of the packages MarlinTPC and PathFinder (see Changelog below).

Please refer to the Release notes below for more details.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-16-02.

Reference installation in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-16-02 # SL5 64bit  
/afs/desy.de/project/ilcsoft/sw/i386_gcc41_sl5/v01-16-02 # SL5 32bit (deprecated)
```

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

Cheers & Merry Christmas,

Christoph, for the iLCSoft team.

Release Notes for v01-16-02 and v01-16-01 (for v01-16 see previous posts)

=====
Changes in v01-16-02 w.r.t. iLCSoft v01-16-01

=====
MarlinReco: to v01-06 from v01-05
=====

- updated IsolatedLeptonfinder (T.Tanabe)
 - LAL Lepton Finder included in IsolatedLeptonFinderProcessor

- updated BCalTagEfficiency (J.List)
 - new option in BCalTagEfficiency: allows topick correct MCParticles from BCALMCTruthLink collection written by SGV

- updated FPCCDClustering (D.Kamai)
 - t.mori enable to difine pixel size for each layer.

- fixed memory leak in BCalReco - fix provided by D.Jeans
 - (loop variable in BCalReconstruction::Free3DArray)

Interface changed.

=====
KalTest: to v01-05-02 from v01-05-01
=====

An infinite loop is now avoided by catching a division by zero due to unreasonably small values of tanLambda.
No interface changes.

=====
PathFinder: to v00-03 from v00-02
=====

- + added errors on track parameters
- + added option to use vertex constraint for circles

+ for allowed distance between hit and track different values can be used before and after the fit
+ for circles use inverse of distance between origin and center of circle instead of the distance
Interfaces changed.

=====
MarlinTPC: to v00-11 from v00-10
=====

- + Updated the interface processor to PathFinder v00-03
- + Added example for CED Viewer
- + Updated magnetic field example
- !+ Fixed unnoticed collection modification in ChannelMappingProcessor: change in steering files!
- + introduced a single versions number file for simpler release management
- + lots of additions for Pixel reconstruction
- several smaller bug fixes

=====
Changes in v01-16-01 w.r.t. iLCSoft v01-16
=====

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

Geant4: 9.5.p02

=====
LCIO: v02-03-03 / v02-03-02
=====

- fixed bug in TrackerHitPlaneImpl and TrackerHitZCylinderImpl

with size of covariance matrix being 0 (no set to 6)
that caused corrupted LCIO files when any hits of these types
were written as "TrackerHit" types, e.g. in a new subset collection
that is made persistent ...
this is no problem for the DBD standard reconstruction !

- changelog from v02-03-02:
 - added lcio_merge_files tool
 - fixed cmake 2.6 incompatibility introduced in v02-00-02

Subject: ILDConfig bbudsc_3evt_stdreco.xml unable to access BgOverlay file
Posted by [gwilliam](#) on Wed, 06 Mar 2013 17:24:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I've recently started getting involved with the ILC and have been trying to get to grips with the software. I've installed it locally and am trying to simulate and reconstruct some test events following the trunk of ILDConfig at:

<http://svnsrv.desy.de/viewvc/marlinreco/ILDConfig/trunk/StandardConfig/current/README?view=markup>

The simulation proceeded fine and I am able to inspect the events with anajob. However, when I go to reconstruct the events with:

Marlin bbudsc_3evt_stdreco.xml

I get the error:

```
*****
```

A runtime error occurred - (uncaught exception):

```
lcio::IOException: [SIORReader::open()] File(s) not found:  
/afs/desy.de/project/ilcsoft/www/html/data/ilctest/sv01-14-0  
1-p00.mILD_o1_v05.E1000-B1b_ws.Paa_lowpt.I106472.eR80.pL20-0 0001.slcio
```

Marlin will have to be terminated, sorry.

Looking in bbudsc_3evt_stdreco.xml I see it is trying to access that file for BG overaly but I don't have permissions to access it:

ls: cannot access /afs/desy.de/project/ilcsoft/www/html/data/ilctest/sv01-14-01-p00.mILD_o1_v05.E1000-B1b_ws.Paa_lowpt.l106472.eR80.pL20-0 0001.slcio: Permission denied

Is this a mistake in the steering or can someone tell me how I can get hold of this file.

Thanks a lot in advance,

Carl

Subject: Re: ILDConfig bbudsc_3evt_stdreco.xml unable to access BgOverlay file
Posted by [gaede](#) on Sun, 10 Mar 2013 12:08:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Carl,

this is just an example background file for gamma gamma -> hadrons.
You can either comment out the BgOverlay processor in your steering file or - if you want to include bg overlay - you need to get a larger number of bg files from the grid in order to have reasonable statistics.

(see: https://ilcproddb.desy.de/admin/ild/gen/?process_type=aa_lowpt)

-Frank.

Subject: New ilcsoft developers release version v01-17
Posted by [rosemann](#) on Mon, 25 Mar 2013 16:26:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new developers release of ilcsoft (v01-17) is available.

The biggest changes are with the external tools Geant4.9.6p01 and root v5_35_05. In addition there is a new package available, namely the General Broken Lines (re-)fit package by Claus Kleinwort.

Please note: on SL5 64bit, (in most cases) there will be segfault caused by ROOT after the end of the Marlin job. Every file (including ROOT files) will be properly closed. Please ignore the error message.

This bug/feature is/was not seen in the SL6 release. A prescription how to circumvent this (if possible) will be attached as an answer to this thread.

Please refer to the Release notes below for more details on most changes.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-17.

Reference installation in afs are available at:

/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17 # SL5 64bit

/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17 # SL6

No SL5/32bit version is provided; it will be created only on specific request.

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

Cheers

Christoph, for the iLCSoft team.

Release Notes for v01-17:

=====

Release notes from iLCSoft v01-17

=====

New developers release of ilcsoft: v01-17

Please note: there is a seemingly unfixable bug in RooT; it causes a segfault at the end of a Marlin job. From the ilcsoft side everything is fine and properly finished.

Despite great effort no clear cause nor fix could be found; please report if you have a solution.

=====
Changes in v01-17 w.r.t. iLCSoft v01-16-02
=====

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

=====
Geant4: 9.6.p01
=====

=====
Root: v5-34-05
=====

=====
New package
=====

The General Broken Lines packages by Claus Kleinwort is now installable by ilcsoft.

=====
GBL: v01-15
=====

=====
Changed packages with interface changes
=====

=====
GEAR v01-03:
=====

- F.Gaede: added test/testInteractionLengths

- new example program to print interaction lengths as a function of theta for different cylinders between the calorimeters
- updated testMaterialBudgetNew.cc: incl. 1st layer of Ecal
- added example macros for DBD material plots:
 - ./example/drawintlen_DBD.C
 - ./example/drawmat_DBD.C
 - ./example/treeFromAscii.C
- C.Rosemann: fixed many issues found by coverity mostly uninitialized member variables
 - GearMgrImpl.cc:make assignment operator and copy c'tor private
 - > see Changelog for details
- M.Killenberg: wrote XML documentation for the VersatileDiskRowLayout
 - doc/VersatileDiskRowLayoutXML.*
 - improved the documentation of localToGlobal and globalToLocal in TPCModule
 - speedup of TPCParametersImpl::getModule(), removed duplicate search in map
 - getDistanceToPad called distanceToBox with wrong order of arguments
- bug fixes:
 - fixed error handling in stepping in ./src/geartgeo/TGeoGearDistanceProperties.cc
 - two bug fixes for the RectangularPadRowLayout (M.Killenberg)
 - fixed endless loop in case of rounding errors TPCParametersImpl/RectangularPadRowLayout
 - two pi wrap around for coordinates in TPCModuleImpl (Bo Li)

=====
 CED v01-08:
 =====

New features (A. Miyamoto):

- Add "Auto shot" flag in Tools menu of glced.
 When set, glced write a screen image to a file when new event is received from CED client. The file is written in a directory where glced is executed with a name, glced-N.tga, where N is a sequence number. If an environment variable, CED_IMAGEFILE is set, output file name is \${CED_IMAGEFILE}-N.tga. Note that a file could be written at any place by including a directory in \${CED_IMAGEFILE}.

A menu, Tools->Auto scale, can be used to select the scale of output image.

- A CMake variable, CED_NOT_INCLUDE_OPENGL_LINKER_PATH. If set, OpenGL linker path is not included in rpath of glced. Inclusion of the OpenGL path caused a problem at run time when gcc used for OpenGL is older than those used to build glced.
- Add 2 more CED_HIT_TYPE, CED_HIT_BOX and CED_HIT_VXD. Then will be used to display pixel hits of FPCCD.

New feature (O. Volynets):

- Added shortcut ctrl+s for fast saving of screenshots in original size (i.e. without scaling)

=====
MarlinReco v01-07:
=====

- Manqi Ruan: added new clas G2D:
Include G2CD, a general digitizer for gaseous hadron calorimeter. It takes 1mm simulated Hcal hits as input, and output digitized hits with tunable sell size. Efficiency and Multiplicity effects are also taken into account, which can be adjusted in the steering file
- J.List: fixed BCalTagEfficiency:
adapted efficiency parametrisation for ECM=1TeV
- Katsushige Coterra: updated NewLDCCaloDigi.cc for the Scintillator Strip Ecal reconstruction

=====
MarlinUtil v01-06:
=====

- Akiya Miyamoto:
implemented flags to turn On and Off geometry of EcalBarrel, EcalEndcap, HcalBarrel, HcalRing, HcalEndcap individually.

=====
Mokka mokka-08-01:
=====

- A. Lucacci-Timoce: added new drivers for the test beam
Add drivers for TBCern2011WAHCAL model

S. Lu: added new physics list from GEANT4.9.6 to PhysicsListFactor

=====
PathFinder v00-04:
=====

- development by I. Heinze
- coverity bug fixes by Ch. Rosemann and corrections to this by O. Volynets

=====
MarlinTPC v00-12:
=====

- Bug fixes by K. Fujii, A. Muennich, I. Heinze, M. Killenberg, T. Krautscheid, Ch. Rosemann
- Added functionality by I. Heinze, A. Muennich, M. Killenberg, F. Gaede, Ch. Rosemann
- Severely updated versions of the RowBasedHitFinder and PulseFinder
- confer the logs for detailed information

=====
LCTuple v01-02:
=====

J. List: added some variables

=====
Druid 2.2:
=====

by Manqi Ruan

=====
Changed packages with fixes without interface changes
=====

=====
KalTest v01-05-03
=====

- bug fix provided by K.Fujii
that makes adjusting of the z0 for curlers in ThelicalTrack::MoveTo unnecessary

=====
Coverity fixes
=====

Fixes from the static code checker coverity caused a minor version update in the following packages; no interface changes.
Please refer to the specific commit logs for more information:

=====
KalDet v01-11-01

=====
LCCD v01-02-01

=====
Marlin v01-04-01

=====
MarlinTrk_version v01-10-02
=====

Subject: root bug/feature in sl5 v01-17
Posted by [rosemann](#) on Mon, 25 Mar 2013 16:35:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

rosemann wrote on Mon, 25 March 2013 09:26

[...] Please note: on SL5 64bit, (in most cases) there will be segfault caused by ROOT after the end of the Marlin job. Every file (including ROOT files) will be properly closed. Please ignore the error message.

This bug/feature is/was not seen in the SL6 release. A prescription how to circumvent this (if possible) will be attached as an answer to this thread.[...]

Large effort has been put into finding the cause of this bug. Unfortunately no handle could be found.

To prevent the error, you can try to omit libLCFIPlus.so and libMarlinPandora.so from you MARLIN_DLL environment variable -- if possible/applicable. No spurious code was found in these libraries, and the behaviour changes with different root versions.

For identification purposes; after the summary section of Marlin at the end of the execution, the error will look something like this:

*** Break *** segmentation violation

=====
There was a crash.

This is the entire stack trace of all threads:

```
=====
#0 0x0000003e79099df5 in waitpid () from /lib64/libc.so.6
#1 0x0000003e7903c7a1 in do_system () from /lib64/libc.so.6
#2 0x00002b748807b402 in TUnixSystem::StackTrace() ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCore.so.5.34
#3 0x00002b748807817a in TUnixSystem::DispatchSignals(ESignals) ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCore.so.5.34
#4 <signal handler called>
#5 0x00002b748899af10 in G__resetglobalvar ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCint.so.5.34
#6 0x00002b7488039ac4 in TCint::ResetGlobalVar(void*) ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCore.so.5.34
#7 0x00002b748920822a in TFile::~TFile() ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libRIO.so.5.34
#8 0x00002b748801f899 in TCollection::GarbageCollect(TObject*) ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCore.so.5.34
#9 0x00002b74880257ea in TList::Delete(char const*) ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCore.so.5.34
#10 0x00002b7487fd579c in TROOT::~TROOT() ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCore.so.5.34
#11 0x0000003e7903383e in __cxa_finalize () from /lib64/libc.so.6
#12 0x00002b7487f880c6 in __do_global_dtors_aux ()
   from /afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17/root/5.34.05/lib/libCore.so.5.34
#13 0x0000000000000000 in ?? ()
```

Subject: Re: New ilcsoft developers release version v01-17

Posted by [sailer](#) on Wed, 03 Apr 2013 16:40:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

You can try

```
export MARLIN_DEBUG=1
```

This keeps Marlin from unloading the libraries from MARLIN_DLL.
Then the error disappears for me.

Besides not printing a humongous error message, only the output

```
[ VERBOSE "UNKOWN"]
```

```
[ VERBOSE "UNKOWN"] <!-- MARLIN_DEBUG=1 set in your environment - skip unloading  
processors -->
```

appears at the end.

Or are there any adverse effects to not unloading the libraries I am not aware of?

Cheers,
Andre

PS: The main reason would be for automatic systems to not be thrown off by a non-zero exit status of Marlin.

Subject: Re: New ilcsoft developers release version v01-17
Posted by [gaede](#) on Thu, 04 Apr 2013 09:11:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Andre,

thanks for providing this Workaround !

In fact the dlclose() call is not strictly needed - it just seemed to be the clean way of closing opened resources at the end of the program.

Anyways, the next Marlin release will not do it anymore and MARLIN_DEBUG will no longer be needed.

Cheers, Frank.

Subject: New iLCSoft developers release v01-17-01
Posted by [gaede](#) on Fri, 24 May 2013 12:43:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new developers release of ilcsoft (v01-17-01) is available.

It provides many smaller fixes, a number of new features and some new packages:

DD4hep: Geometry description (M.Frank, P.Mato)

SLIC, XercesC, HepPDT, GDML, LCDD - Slic geant4 simulation (J.McCormick)

Please refer to the Release notes below for details.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-17-01.

Reference installation in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17-01 # SL5 64bit  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-01 # SL6
```

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

Frank, for the iLCSoft team.

```
#####  
#  
# iLCSoft release - v01-17-01  
#  
#####
```

Developers release that provides many smaller fixes and a number of new features as well as a number of new packages.

=====

Changes in v01-17-01 w.r.t. iLCSoft v01-17

=====

New packages added to iLCSoft:

GBL: General Broken Lines (K.Kleinworth, Ch.Rosemann)
DD4hep: Geometry description (M.Frank, P.Mato)
SLIC, XercesC, HepPDT, GDML, LCDD - Slic geant4 simulation (J.McCormick)

=====
LCIO: v02-04
=====

- Ch.Grefe, CERN
added a new python binding: pyLCIO based on the optional ROOT dictionary
see: \$LCIO/examples/python/README for examples on how to use it
- added new method to limit the collections that are going to be read
from a file:
LCReader::setReadCollectionNames(const std::vector<std::string>& cn)
- C++ only so far
- anajob and dumpevent now read variable LCIO_READ_COL_NAMES with
space separated list of collection names to read only these
- pointers to LCObjects that are not in the event are now set to NULL
- an Exception is thrown if a null pointer is found in a subset collection
(e.g. if LCIO_READ_COL_NAMES is set inconsistently such that
the collection(s) holding the elements is not read)
- this behavior can be deactivated with setting the environment variable
LCIO_IGNORE_NULL_IN_SUBSET_COLLECTIONS=1
if really needed, e.g. to read inconsistent or corrupt files
- added build option for C++ only builds
using pre-generated header files:
cmake -DLCIO_GENERATE_HEADERS=OFF ..
- note: header files will have to be updated manually by developers
after changing the class layout
- some minor fixes:
 - fix missing include issue in UTIL/LCIterator.h for latest ubuntu
 - update freehep-sio dep to 2.3
 - enforce cmake version >= 2.8

=====
Marlin: v01-05

=====

- new optional global parameter: "LCIORReadCollectionNames" allows to restrict the collection that are read from the LCIO file(s) to the ones given (needs LCIO v02-04)
- new command line option (H.Perrey, DESY):
 - u : consistency check and updated version of xml file
- fix problem with ROOT seg fault at program end: do not call dlclose on shared libraries anymore (A.Sailer)

=====

CED: v01-09

=====

- turned off C++ name mangling in client library libCED.so/dylib to facilitate calling ced from python

=====

CEDViewer: v01-07

=====

- changed calling attributes of ced_hit_ID to newer version
 - > needed for CED v01-09 which has no function overloading in library anymore (c-style)
 - changed the default conventions for the track state to be drawn. Now the following processor options are valid
 - 1: do not draw helix
 - 0: default track state (the 0-th element in the track state vector) (the rest options as before)
 - 1: AtIP
 - 2: AtFirstHit
 - 3: AtLastHit
 - 4: AtCalorimeter
- If no track state were found, exit with an error

- avoid wrong values for the helix parameters for the case of small curvature or zero magnetic field by assigning large value of Pt to the track
- update for the multi-module support via generic GEAR interface

=====
 Clupatra: v00-10
 =====

- bug fix: remove sqrt from computation of $r_{\phi_res}^2$ for finding best hit
- included multi-module TPC support via GEAR interface
- ignore hits that are outside of the TPC range (possible in data)
- added new processor FixCellIDs_Error_TestBeam (based on the original FixCellIDs) that sets the correct hit CellIDs and hit uncertainties for test beam data
- added examples for test beam data. Look at examples/README for more details

=====
 LCTuple: v01-03
 =====

- added support for TrackerHits

=====
 KalDet: v01-12
 =====

- added detector geometry for lctpc which is now based on ild classes. Notes:
 - it is compatible with Clupatra/MarlinTrk
 - the correspondence between row number and module is represented in a different way compared to the original class GearTPCKalDetector. In the current implementation, row numbers are global, i.e. rows of different modules but with the same center are now the same rows! This is needed to unify the measurement layers that are combinations of several modules.

=====
MarlinTPC: v00-13
=====

- A. Muennich: added preparation processors for Clupatra test beam data usage
 - added a processor ADC2PrimaryElectronFactorCalculator (needs conditions object transformation)
 - added a helper tool to calculate intersection between circle and track in global coordinates
 - Reimplementation of the splitting in the PulseFinder:
 - bug fix to HitCopier
- F. Mueller: added two more methods to calculate pulse time correction of hit error calculation
- S. Caiazza: added documentation
- F. Gaede: added extension to CED
- I. Heinze: Extended CutApplicationProcessor so that tracks with more than one hit per row can be cut out.
- O. Volynets introduced new track description with MarlinTrk
- M. Killenberg: Wrote documentation for the IntersectionCalculator
- Ch. Rosemann: added first version of straight line fitting in row based TPC with general broken lines
 - HitCopier added
- + Bug fixes by I. Heinze, A. Muennich, M. Killenberg, T. Krautscheid, O. Volynets, F. Mueller, F. Gaede, Ch. Rosemann

=====
MarlinTrk: v01-11
=====

- Improved readability of the MarlinTrkUtils functions
- Added LCTPC-specific modifications:
 - geometry building is now based on the detector name from GEAR
 - Magnetic field may be zero -> avoid the value of infinity for kappa
 - Interaction Point (IP) meas. layer may not exist for LCTPC
 - Do not throw exception if the layer is not found, return "no_intersection" code instead. This avoids the problem if some of the state is not or cannot be available,

e.g. the IP and CaloFace states are not (yet?) defined for the LCTPC geometry

=====
MarlinTrkProcessors: v01-09-01
=====

- updated calling attributes of ced_hit_ID to new CED v01-09

=====
MarlinUtil: v01-06-01
=====

- adopted for multi-module support in MarlinCED for the TPC detector

Subject: New iLCSoft developers release v01-17-02
Posted by [rosemann](#) on Mon, 10 Jun 2013 15:37:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new patch release for the v01-17 cycle is available: v01-17-02

It consists mainly of bug fixes/updates in the CondDBMySQL, LCCD and LCIO packages. Additionally features in LCIO and MarlinUtil were added.

Please refer to the Release notes below for more details on most changes.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-17-02.

Reference installation in afs are available at:

/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17-02 # SL5 64bit
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-02 # SL6

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

Cheers
Christoph, for the iLCSoft team.

=====
Changes in v01-17-02 w.r.t. iLCSoft v01-17-01
=====

Quick patch release that fixes bugs in LCCD and CondDBMySql and adds
functionality in MarlinUtil

=====
LCIO v02-04-01
=====

- LCReader/SIOReader sets the RunHeader parameter "LCIOFileName" to the current file name (usefull if a list of files is processed)
- bug fix in SIOReader: used to hold pointer to StringVec argument of SIOReader::open(StringVec& files) - now the filenames are copied to local class member variable
- added missing const keywords for getValue() and highestBit() in UTIL::BitField64 (Ch.Grefe)

=====
MarlinUtil v01-07
=====

- split up library into libMarlinUtil and libMarlinUtil_ann

=====
LCCD v01-03:
=====

- updated all database test binaries
 - source/test/conddbtests/multiplefolders.cc
 - source/test/conddbtests/calfiletoddb.cc
 - source/test/conddbtests/calfilefromdb.cc
 - source/test/conddbtests/createdbfile.ccto take the db folder as argument on the command line as part of the DBcol name
example:

./bin/calfiletodb condDB_channelmapping_HEAD_20130604_155014.slcio channelmapping
/lctpc/test/channelmapping 0 999999

- bug fix: wrong declaration logic (made obvious by coverity bugfix):
move declaration and initialization outside loop in LCGenericObjectStreamer

=====
CondDBMySQL_ILC-0-9-6

=====
- updated boot-sql.h to newer MySQL syntax

Subject: [ilcinstall - SLIC] SLIC compilation errors [SOLVED]
Posted by [aconway](#) on Fri, 05 Jul 2013 22:22:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I was running ilcsoft-install on both my own laptop (Arch Linux) and a remote machine (SL 6.2) and both ran into an error when compiling SLIC. The error is in the log file attached.

I don't know how to work around this so I can't install until it's fixed.

Thanks,
Alex

EDIT
Was a bug in the code which has now been fixed.

File Attachments

1) [release-scratch-2013-7-5_15-47.log](#), downloaded 2071 times

Subject: [ilcinstall] Minor problems with ilcsoft-install and related scripts.
Posted by [aconway](#) on Fri, 05 Jul 2013 22:44:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I ran into two very minor issues when running ilcsoft-install on my Arch Linux laptop.

The first issue that came up was the following error message:

```
[aconway@tp ilcinstall]$ python2 ilcsoft-install -i ./examples/slic/release-scratch.cfg
+ Running ilcsoft-install [ v01-17-02 ]
+ Read configuration file [ /home/aconway/fermi/ilcinstall/examples/slic/release-scratch.cfg ]
Traceback (most recent call last):
  File "ilcsoft-install", line 72, in <module>
    execfile( config_file )
  File "/home/aconway/fermi/ilcinstall/examples/slic/release-scratch.cfg", line 28, in <module>
    ilcsoft = ILCSoft( ilcsoft_install_dir )
  File "/home/aconway/fermi/ilcinstall/ilcsoft/ilcsoft.py", line 70, in __init__
    self.release_number = release_string[re.search('\d', release_string).start()]
AttributeError: 'NoneType' object has no attribute 'start'
```

It turns out that `release_string` is defined on line 65 of `ilcsoft.py` as the output of ``lsb_release -d 2>/dev/null``. On my computer, that command outputs "Description:\tArch Linux", which doesn't have any numbers. Thus the regex on line 70 returns a null value. I got around it by deleting line 70, which didn't seem to mess anything up.

The second issue was in finding my Java installation. My Java binaries are in `/opt/java/bin` so I couldn't add the path to `ilcsoft/java.py` because the strings in `self.reqfiles[]` are automatically prepended with `'usr/'`. I fixed this by symlinking `java`, `javac`, `jar` and `javadoc` to `/usr/bin`, but a cleaner solution would have been nicer. In my case the correct paths could have been acquired simply through ``which java`` etc.

Thanks,
Alex Conway
aconway at fnal dot gov

Subject: [ilcinstall - slicPandora] "NameError: name 'SlicPandora' is not defined"
Posted by [aconway](#) on Tue, 09 Jul 2013 21:12:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
Hi,

I'm running into trouble attempting to install slicPandora with ilcinstall. I got slic installed but when I try to install slicPandora with

./ilcsoft-install examples/slicPandora/release-scratch.cfg
I get the error below.

Anyone know how to fix this?

Thanks,
Alex

```
-bash-4.1$ ./ilcsoft-install examples/slicPandora/release-scratch.cfg
+ Running ilcsoft-install [ v01-17-02 ]
+ Read configuration file [
/cloud/login/aconway/ilcinstall/examples/slicPandora/release-scratch.cfg ]
+ PLATFORM Linux - Scientific Linux Fermi release 6.2 (Ramsey)
+ GCC_VERSION gcc (GCC) 4.4.6 20110731 (Red Hat 4.4.6-3)
+ LSB_RELEASE LSB
Version: :core-4.0-amd64:core-4.0-noarch:graphics-4.0-amd64:graphics-4.0-noarch:printing-4.0-a
md64:printing-4.0-noarch
Distributor ID: ScientificFermi
Description: Scientific Linux Fermi release 6.2 (Ramsey)
Release: 6.2
Codename: Ramsey
+ UNAME Linux mcdrd.fnal.gov 2.6.32-279.22.1.el6.x86_64 #1 SMP Tue Feb 5 14:33:39 CST
2013 x86_64 x86_64 x86_64 GNU/Linux
```

Traceback (most recent call last):

```
File "./ilcsoft-install", line 72, in <module>
  execfile( config_file )
File "/cloud/login/aconway/ilcinstall/examples/slicPandora/release-scratch.cfg", line 106, in
<module>
  ilcsoft.install( SlicPandora( SlicPandora_version ))
NameError: name 'SlicPandora' is not defined
-bash-4.1$
```

Subject: Re: [ilcinstall - slicPandora] "NameError: name 'SlicPandora' is not defined"
Posted by [sailer](#) on Wed, 10 Jul 2013 04:18:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Alex,

I think you have to add a line of:

from slicpandora import SlicPandora
For all the other modules the corresponding line is in
ilcinstall/ilcsoft/___init___py
but for SlicPandora this has not happened yet.

Cheers,
AndrÃ©

Subject: Re: [ilcinstall - slicPandora] "NameError: name 'SlicPandora' is not defined"
Posted by [aconway](#) on Tue, 16 Jul 2013 16:09:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, that fixed it, thank you.

Subject: [ilcinstall - slicPandora] Gear installation identifies ROOT improperly for
some users
Posted by [aconway](#) on Tue, 16 Jul 2013 17:12:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

According to the ROOT folks, packages with ROOT dependencies should not rely on the
environment variable \$ROOTSYS, since this is incompatible with most distribution-provided
ROOT packages, which installs ROOT in the usual places (/usr/bin, /usr/lib, /usr/lib64) etc.

Quote: please let us know which packages depend explicitly on \$ROOTSYS because they need
fixing as they should not rely on ROOTSYS but on root-config, or when in code on the global
gRootDir or on include/RConfigure.h.

- <http://root.cern.ch/phpBB3/viewtopic.php?f=3&t=14545>

Gear depends on the \$ROOTSYS variable and the effect of this is that Gear fails to compile when
installing slicPandora and doesn't say why. I'm working around the issue by installing ROOT
according to the 'Location Independent Installation' instructions at this page:
<http://root.cern.ch/drupal/content/installing-root-source>.

Thanks,
Alex Conway

File Attachments

1) [release-scratch-2013-7-16_11-5.log](#), downloaded 329 times

Subject: [ilcinstall - slicPandora] Gear installation fails, cannot find CLHEP?
Posted by [aconway](#) on Tue, 16 Jul 2013 17:44:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I'm still trying to install slicPandora and I am running into an error while trying to compile Gear.

-- Configuring incomplete, errors occurred!

make: *** No targets specified and no makefile found. Stop.

make: *** No rule to make target `install'. Stop.

*** ERROR in module [GEAR]: DEBUG INFO: {'PLATFORM': 'Linux - Arch Linux',
'GCC_VERSION': 'gcc (GCC) 4.8.1', 'LSB_RELEASE': 'LSB Version:\t1.4\nDistributor
ID:\tarch\nDescription:\tArch Linux\nRelease:\trolling\nCodename:\tn/a', 'UNAME': 'Linux tp
3.9.2-1-ARCH #1 SMP PREEMPT Sat May 11 20:31:08 CEST 2013 x86_64 GNU/Linux'}

*** ERROR in module [GEAR]: At least one of these files:

['/home/aconway/fermi/slicPandora/v00-00/gear/v01-03/lib/libgear.a',
'/home/aconway/fermi/slicPandora/v00-00/gear/v01-03/lib/libgear.so',
'/home/aconway/fermi/slicPandora/v00-00/gear/v01-03/lib/libgear.dylib']
is required for using this installation of GEAR

Full logfile is attached.

If I try to build Gear myself, I get the following:

```
[aconway@tp build]$ cmake
```

```
-DILCUTIL_DIR=/home/aconway/fermi/slicPandora/v00-00/ilcutil/v01-00/ ..
```

```
-- Check for ILCUTIL (1.0.0)
```

```
-- Found ILCsoft_CMAKE_MODULES:
```

```
/home/aconway/fermi/slicPandora/v00-00/ilcutil/v01-00/cmakemodules
```

```
-- Found ILCUTIL: /home/aconway/fermi/slicPandora/v00-00/ilcutil/v01-00
```

```
-- Check for CLHEP_LIBRARIES: CLHEP
```

```
-- Check for CLHEP_CLHEP_LIBRARY: CLHEP -- failed
```

```
CMake Error at /usr/share/cmake-2.8/Modules/FindPackageHandleStandardArgs.cmake:108  
(message):
```

```
Could NOT find CLHEP (missing: CLHEP_INCLUDE_DIRS CLHEP_LIBRARIES
```

PACKAGE_VERSION_COMPATIBLE)

Call Stack (most recent call first):

/usr/share/cmake-2.8/Modules/FindPackageHandleStandardArgs.cmake:315

(_FPHSA_FAILURE_MESSAGE)

/home/aconway/fermi/slicPandora/v00-00/ilcutil/v01-00/cmakemodules/FindCLHEP.cmake:85

(FIND_PACKAGE_HANDLE_STANDARD_ARGS)

CMakeLists.txt:43 (FIND_PACKAGE)

-- Configuring incomplete, errors occurred!

Not sure how to continue.

Thanks,
Alex Conway

File Attachments

1) [release-scratch-2013-7-16_12-36.log](#), downloaded 536 times

Subject: Re: [ilcinstall - slicPandora] Gear installation fails, cannot find CLHEP?

Posted by [sailer](#) on Tue, 16 Jul 2013 19:11:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Alex,

This error is in the log file

Please install doxygen or set INSTALL_DOC to OFF

So try setting INSTALL_DOC=OFF

in the release-scratch.cfg

When running cmake by hand you probably also have to specify

-DCLHEP_DIR=/path/to/clhep

I tried the slicpandora example myself now.

There was a problem with the Marlin head version, maybe use the tagged versions for Marlin v01-05 and lcio v02-04-01.

Cheers,
Andr ©

Subject: Re: [ilcinstall - slicPandora] Gear installation identifies ROOT improperly for some users

Posted by [sailer](#) on Tue, 16 Jul 2013 19:15:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Actually I think this is also related to INSTALL_DOC

Though gear should set INSTALL_DOC to OFF when doxygen is not found, I think the other packages do it that way.

The log file shows that it finds all the root libraries

```
-- Check for ROOT_CONFIG_EXECUTABLE: /bin/root-config
-- Check for ROOT (5.34.07)
-- Check for ROOT_EXECUTABLE: /usr/bin/root
-- Check for ROOT_CINT_EXECUTABLE: /usr/bin/rootcint
-- Check for ROOT_LIBRARIES:
Core;Cint;RIO;Net;Hist;Graf;Graf3d;Gpad;Tree;Rint;Postscript;Matrix;Physics;MathCore;Thread
-- Check for ROOT_COMPONENT_LIBRARIES: Geom
-- Check for ROOT_CORE_LIBRARY: /usr/lib64/root/libCore.so -- ok
-- Check for ROOT_CINT_LIBRARY: /usr/lib64/root/libCint.so -- ok
-- Check for ROOT_RIO_LIBRARY: /usr/lib64/root/libRIO.so -- ok
-- Check for ROOT_NET_LIBRARY: /usr/lib64/root/libNet.so -- ok
-- Check for ROOT_HIST_LIBRARY: /usr/lib64/root/libHist.so -- ok
-- Check for ROOT_GRAF_LIBRARY: /usr/lib64/root/libGraf.so -- ok
-- Check for ROOT_GRAF3D_LIBRARY: /usr/lib64/root/libGraf3d.so -- ok
-- Check for ROOT_GPAD_LIBRARY: /usr/lib64/root/libGpad.so -- ok
-- Check for ROOT_TREE_LIBRARY: /usr/lib64/root/libTree.so -- ok
-- Check for ROOT_RINT_LIBRARY: /usr/lib64/root/libRint.so -- ok
-- Check for ROOT_POSTSCRIPT_LIBRARY: /usr/lib64/root/libPostscript.so -- ok
-- Check for ROOT_MATRIX_LIBRARY: /usr/lib64/root/libMatrix.so -- ok
-- Check for ROOT_PHYSICS_LIBRARY: /usr/lib64/root/libPhysics.so -- ok
-- Check for ROOT_MATHCORE_LIBRARY: /usr/lib64/root/libMathCore.so -- ok
-- Check for ROOT_THREAD_LIBRARY: /usr/lib64/root/libThread.so -- ok
-- Check for ROOT_GEOM_LIBRARY: /usr/lib64/root/libGeom.so -- ok
-- Check for libdl.so: /usr/lib64/libdl.so
-- Found ROOT: /usr/include/root
--
```

Subject: New iLCSoft developers release v01-17-03
Posted by [gaede](#) on Mon, 19 Aug 2013 15:47:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new patch release for the v01-17 cycle is available: v01-17-03

It provides bug fixes and some new features in a number of packages. Please refer to the Release notes below for more details on most changes.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-17-03.

Reference installation in afs are available at:

/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17-03 # SL5 64bit
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-03 # SL6

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

Frank, for the iLCSoft team.

=====
Changes in v01-17-03 w.r.t. iLCSoft v01-17-02
=====

Developers release with some updated packages:

=====
LCIO v02-04-02
=====

- add template std::string toString(const T* obj) for easy printing w/o operators (e.g. from Python)
- cleanup of Cmake code for BUILD_ROOTDICT
- fixed (again) target dylib2so for APPLE

- Updated the StdHepReader to use the exact same interface as the LcioReader (Ch.Greffe)

=====
GEAR v01-03-01
=====

- added convenience wrapper BField::at(double x, double y, double z)

=====
Mokka mokka-08-02:
=====

- B.Pawlik:
 - bug fix for global position of LumiCal hits.
- A.Lucaci:
 - Updated Ecal driver SEcal05 (J.Marshall)
Fixed variable cell sizes, layer number. (J.Marshall)
Fixed EndCap problem in case of cells with varying cell sizes for hybrid ECAL. (J.Marshall)

- S.Lu: Added a scalable Ahcal driver SHcalSc04. (a copy of SHcalSc03)
The geomemtry of Ahcal Barrel and Ahcal Endcaps have been updated.

Added a scalable sensitive detector driver SDAHcalEndCapScalable.
Implement a scalable AhcalEndcaps driver, the Endcaps modules number and the modules length are derived from the hcal barrel maximum radius at the run time.

- S.Lu: Created Mokka model "ILD_o1_v06" on "pollin1.in2p3.fr".
using "SEcal05" as default ECAL driver, (Silicon ECAL)
and a scalable driver "SHcalSc04" as default Ahcal driver.

Created Mokka model "ILD_o2_v06" on "pollin1.in2p3.fr".
using "SEcal05" as default ECAL driver.
(SD HCAL).

Created Mokka model "ILD_o3_v06" on "pollin1.in2p3.fr".
using "SEcal05" as default ECAL driver, (Scintillator ECAL)

and a scalable driver "SHcalSc04" as default Ahcal driver.

=====
CED v01-09-01
=====

- fixed wrapper function `ced_hit(...,type,...)` to call `ced_hit_ID(..., type & 0xFF,type >> CED_LAYER_SHIFT ,...)` for backward compatibility with old code

=====
ilcutil v01-01
=====

cmakemodules v01-14 :

- Changes to FindROOT.cmake (H.Perrey)
 - modified findroot module to first check for ROOT in \$ROOTSYS, then in \$PATH directories, before looking in /usr/bin; this makes sure that the ROOT version installed with ILCsoft or local installations (if existing/set up correctly) are preferred over system-wide packages and avoids conflicts between the installations.

=====
MarlinUtil v01-07-01
=====

- added wrapper function (extern "C")
`void draw_helix(...)` to `MarlinCED::drawHelix(...)`

=====
MarlinTPC v00-14-00
=====

- F. Mueller:
 - + added simple fit processors for straight line and helix
 - + added listener for FieldSetting
- A. Muennich: work on Distortion processors, several namespace and doxygen fixes

- C. Kleinwort: added new GBL interface processor that also works for helices; includes the straight line case,
 - decision is made depending on magnet field data
- K. Zenker: updating photodot reconstruction
- Ch. Rosemann: removal of old GBL interface processor, completely superseded by new one
 - + new way to access the b field data from the measurement survey: via conditions data, using the gear::BField interface
 - check svn log for further info
 - + several bug fixes

=====
 PathFinder v00-05
 =====

- several bug fixes that enhances track finding efficiency:
 - ! wrong sign in some cases of the sz projection fit
 - ! straight line hough space fixed ranges of parameters
 - ! input binning used coherently

=====
 DD4hep v00-04
 =====

- placement options are now compatible with geant4
- fixed Geant4SensitiveDetector::getVolumeID()
- modified ILDEx model to not use assemblies (for now !?)
- removed obsolete geant4 example ILDEx
- Fix population procedure of volume manager
- Fix uniqueness of volume identifiers.
- Remove call with lookups by placed volume to the volume manager
- improved ILDEx toy example:
 - fixed creation of cellIDS for simhits
 - creates SimTrackerHits for VXD, SIT and TPC
 - creates SimCalorimeterHits for AHcal barrele and endcap
 - works now wigth ILDExSimu _and_ SLIC if no assemblies are used
- assemblies work with the DD4hep Geant4Converter and VolumeManager
- added a prototype example for a ROOT independent plugin mechanism for SensitiveDetectors
- added example implemetation for Calice test beam calorimeters: CaliceTbeam (Shaojun Lu, DESY)

=====
MarlinPandora: v00-11
=====

TAG V00-11 FOR ILC SOFT USAGE

Protect against missing/unavailable TrackStates.
Add modifications to allow use of hybrid ScW/SiW and full ScW ECAL models.

TAG V00-10 FOLLOWING REPOSITORY RESTRUCTURING

CMake and Makefile changes following updated treatment of the libraries that constitute the PandoraPFANew package.

GENERAL INFO:

Pandora consists of the PandoraSDK (Software Development Kit), which comprises a single library and header files with flexible APIs.

There is also an optional PandoraMonitoring library, which contains event display and tree/histogram-writing functionality.

The PandoraMonitoring library depends on the PandoraSDK and ROOT.

Pandora algorithms will necessarily depend upon the PandoraSDK (and also PandoraMonitoring, if desired).

The algorithms can be built as part of a Pandora client application, such as MarlinPandora, or the standalone TestPandora binary.

Algorithms declared to be suitably generic may be designed for use by multiple client applications.

Such re-usable algorithms are best bundled together into a Pandora "content" library.

Algorithms are always registered via the client application, using the Pandora APIs, but the client app can link against as many content libraries as required.

Recover daughter or sibling tracks if they are flagged as unable to form a pfo.

Copy track states from input Lcio tracks.

Fix subdetector hit number encoding.

Write out track subset collections to investigate different track usage in pfa.

Updated TrackCreator class to work with either ILD00 or ILD01 geometry.

=====
PandoraPFANew: v00-12

=====

TAG V00-12 FOR ILC SOFT USAGE

Updated daughter package versions.

TAG V00-11 FOLLOWING REPOSITORY RESTRUCTURING

PandoraPFANew is now a "metadata" package containing only Makefiles for downloading and building relevant Pandora libraries.

GENERAL INFO:

Pandora consists of the PandoraSDK (Software Development Kit), which comprises a single library and header files with flexible APIs.

There is also an optional PandoraMonitoring library, which contains event display and tree/histogram-writing functionality.

The PandoraMonitoring library depends on the PandoraSDK and ROOT.

Pandora algorithms will necessarily depend upon the PandoraSDK (and also PandoraMonitoring, if desired).

The algorithms can be built as part of a Pandora client application, such as MarlinPandora, or the standalone TestPandora binary.

Algorithms declared to be suitably generic may be designed for use by multiple client applications. Such re-usable algorithms are best bundled together into a Pandora "content" library.

Algorithms are always registered via the client application, using the Pandora APIs, but the client app can link against as many content libraries as required.

TAG V00-10 FOR Liquid Argon TPC reconstruction usage

Redesign of MC Manager, enabling creation of multiple MC particle lists.

Add MCParticleType API parameter.

Improve manner in which pandora content is reset between events, adding ability to register reset functions.

Move core EventWriting and EventReading algorithms into Framework library.

Add HitTypes VIEW_U, VIEW_V, VIEW_W, registered with FINE granularity.

Allow visualization of marker at a specified position.

Move FineGranularityContent library to separate repository.

Makefile updates.

=====
PandoraAnanlysis V00-05

=====
FOR ILCSoft USAGE

AnalysePerformance binary now reads TChains, rather than a single TTree.
Included new PfoTarget information in PfoAnalysis tree.
PfoAnalysis now attempts to store an estimate of energy deposited in coil.
Explicitly set ROOT directory for output tree and histograms.
Finalise steering parameters in init method, rather than in constructor.
Added Calibrate test binary.
Added run and event counters for display of number of runs and events processed.
Altered PandoraPFACalibrator histogram binning and names.
Removed unused histograms.
Cosmetic changes.
Makefile update, including basic dependency tracking.

Subject: Installation of ilcsoft v01-17-03 - Bug in DD4hep installation?!
Posted by [poeschl](#) on Wed, 16 Oct 2013 16:20:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Experts,

I am about to carry out an ilcsoft installation from *scratch* and I think I have stumbled over a bug in the installation of the dd4hep package

I got an error message saying

*** ERROR in module [DD4hep]: failed to compile!!

Further upstream it is written e.g.

/srv/ilcsoft/v01-17-03/DD4hep/v00-04/DDCore/include/DD4hep/Objects.h:25:27: error:
Math/Vector3D.h: No such file or directory

This means that the installation doesn't find some root include files.

It appears that the FindROOT.cmake in dd4hep finds my system wide root installation, which indeed doesn't contain the requested include files (why not is my problem but this doesn't matter here). When installed from *scratch* dd4hep should however always use the root version which got installed through the corresponding ilcinstall scripts.

The problem is (to my mind) in this line of code

```
find_program(ROOT_CONFIG_EXECUTABLE root-config
PATHS $ENV{ROOTSYS}/bin)
```

By this it stops searching on the first root-config it finds, which in my case is my (maybe buggy) system wide root installation.

I think one has to remedy this vulnerability, i.e. the FindROOT.cmake has somehow to be told which root version to use instead of starting to search on its own. It is very likely that on a system an additional root version is installed.

A temporary patch is:

```
find_program( ROOT_CONFIG_EXECUTABLE root-config HINTS ${ILC_HOME}/root/5.34.05/bin
NO_DEFAULT_PATH)
```

After that build tells me

```
Found ROOT: /srv/ilcsoft/v01-17-03/root/5.34.05
```

while before it told me

```
Found ROOT: /usr
```

I prefer to use however a kind of official patch because I am afraid to destroy things down the line if I start to joke around with my private patches.

Thanks in advance and cheers,

Roman

P.S.: If the bug has been reported elsewhere and is corrected in the mean time, please excuse my ignorance

Subject: new ilcsoft developers release v01-17-04
Posted by [rosemann](#) on Thu, 24 Oct 2013 14:16:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new patch release in the v01-17 cycle of developers versions is available with v01-17-04

Apart from the usual bug fixes, a number of new features are included. Also a new package "SlicPandora" is introduced.

Please refer to the release notes, which are duplicated below.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-17-04.

Reference installation in afs are available at:

`/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17-04 # SL5 64bit`
`/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-04 # SL6`

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

Christoph, for the ilcsoft team.

=====
Changes in v01-17-04 w.r.t. iLCSoft v01-17-03
=====

Introduces a new package:
=====

SlicPandora v01-00-00
=====

by J. McCormick

=====
LCIO v02-04-03
=====

- Added a statement about the minimum python version to use

- Adapted the example to the changed interface in EventLoop.

```
=====
MOKKA mokka-08-03
=====
```

! Corrected the density value of Nitrogen gas, and Changed the input density of Cherenkov gas by the pressure. (Shaojun Lu)

+ Add new driver TBcatcher09 for the CALICE TailCatcher (Angela Lucaci-Timoce)

- the previous driver (TBcatcher08) had a bug in the TBcatcher08::BuildCatcherLayers()

```
for (G4int i = 0; i < n_layers; i++)
{
    // here is the line with the bug
    // bool withCassette = (_layerPattern.data()[i-1] == '1');
    // the line below is the correct one
    bool withCassette = (_layerPattern.data()[i] == '1');
```

the boolean withCassette was always false for i=0 (i.e. for the first layer). As a result, the first fine layer was never build.

- The sensitive detector was changed to TBSD_VCell04.
With respect to the AHCAL case, for the TailCatcher a 1 is added in the layer number (otherwise we get layers from 0 to 15, instead of layers from 1 to 16)

(I checked that previous TailCatcher drivers use TBSD_VCell03 as sensitive detector, therefore they will not be affected by this change.)

! fixed naming of pointer variables: (Frank Gaede)
removed duplicate class name qualifiers (TBSD_VCell03::TBSD_VCell03)

```
=====
DD4HEP DD4hep v00-05
=====
```


Markus Frank:

- Throw exception if the expression evaluator sees a problem.
The message printed before was often not seen.
- Improve printout on exceptions and plugin abortion
- Remove the typedef RotationZYX -> Rotation and import the type RotationZYX from ROOT::Math.

- Add more explicit error messages in case a plugin is not found.
- Add file TGeoUnits.h, which allows to use the "usual SystemOfUnits" from CLHEP using TGeo with the namespace tgeo::xxx
Example: tgeo::mm -> 0.1 (Geant3: cm, sec, GeV, degree)
 ::mm -> 1.0 (Geant4: mm, nsec, MeV, radian)

I hope I got all translations right.

Geant4		Geant3	
millimeter	mm=1,cm=10	centimeter	mm=0.1,cm=1
nanosecond	s=1e9	second	ns=1e-9,s=1
Mega electron Volt	GeV=1e3	GeV	GeV=1
radian	rad=1	degree	rad=180/pi

Unclear to me was if in Geant3 steradian is also $2*2*\pi$ or $2*2*180$ degree.

F. Gaede:

- changes to build system:
(see ReadMe.txt for details)
 - moved examples to ./example directory
 - they are not built as part of DD4hep anymore
- install thisdd4hep.sh in ./bin
(modified to have the correct pathes)
- added -DINSTALL_DOC=on/off option
to build doxygen documentation (in ./doc/html/index.html)
- create DD4hepConfig.cmake for easy building against DD4hep
(see examples CMakeLists.txt)
- fixed doxygen API documentation (C.Rosemann)
(greatly improved with many graphs)

=====
MarlinReco v01-08
=====

- John Marshall:

Update fillECALGaps function, so that it remains identical to that implemented in NewLDCCaloDigi processor (i.e. copy the NewLDCCaloDigi updates from 9th Jan 2013).
Apply HCAL "other" calibration constant to both ring and plug calorimeter hit types.

=====
MarlinTPC v00-15
=====

General remarks: Following the analysis workshop at KEK several new analysis processors were

developed and added. Each comes with a corresponding root script to make the corresponding graph/plot.

A style file has been added to be used to make plots, so they look consistent.

Authors so far: F. Mueller and A. Muennich

New in analysis/:

- ChargeQualityProcessor
- DistortionPerformanceProcessor
- HitQualityProcessor
- AnalysisCutSelectorProcessor
- TrackParametersDistributionProcessor
- ResolutionPerformanceProcessor
- TripletResolutionPerformanceProcessor
- HistogramWidthExtractionTools
- TripletCalculationTool

New in analysis/rootscripts

- LCTPCStyle.C
- TrackParametersDistribution.C
- ProcessAllAnalyses.C
- LoadAnalysisScripts.C
- ChargeQuality.C
- HitQuality.C
- TripletResolutionPerformance.C
- ResolutionPerformance.C
- DistortionPerformance.C
- AnalysisCutSelector.C

- F. Mueller: added: a processor to synchronize the hit time of data taken with several RCUs
- A. Muennich: added additional time method in PulseFinder to get time from center of gravity of region around maximum.
added usage of reference point in ModuleDistortionCalculator
- P. Hayman: added Micromegas analysis tools: MMPulseFinderProcessor
MMHitFinderProcessor
MMPadResponseFunction
MMHitTimeCorrectionProcessor
MMCalibrationProcessor
- C. Kleinwort: New RowTripletBasedTrackFinderProcessor; FixCellID,
FixErrors adjusted
New track finder (for LPTPC) based on triplets of rows.
TPCHitFixErrors overwrites covariance matrix (instead of adding).
TPCHitFixCellIDs encodes ModuleID in CellID0, need to be switched off for
Clupatra
(`<parameter name="EncodeModuleID" type="bool">>false`
`</parameter>`).
- Bo Li: update for tracking in multi-module of the branch
KalDet_GearTPC_devel
- C. Menzen: added Multichip read-in. Reads individual data files for
each chip individually
multichip capability in InGridSimpleClusterFinder and
TimepixHitAndTrackChargeprocessor
- + minor small improvements on plots and several bug fixes

=====
GBL v01-16-01
=====

- implemented diagonalization of scatters in cpp version
- major cleanup of python version

Subject: Re: Installation of ilcsoft v01-17-03 - Bug in DD4hep installation?!
Posted by [rosemann](#) on Thu, 24 Oct 2013 14:29:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Roman,

sorry, but I've overlooked your message. I have to admit that I can't follow.

I've built dd4hep many times, but I've never encountered the problem you describe.

How do you initialize the installation -- using ilcinstall or a different approach?

If you try to build DD4hep directly (or any other ilcsoft package for that matter), then it's advisable to use the ILCSoft.cmake file as a basis for configuration. It would need to be modified, e.g. the package you want to build needs to be commented out.

Cheers,
Christoph

Subject: Re: Installation of ilcsoft v01-17-03 - Bug in DD4hep installation?!
Posted by [poeschl](#) on Sun, 27 Oct 2013 16:37:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Christoph,

thanks a lot for your reply.

I use standard the ilcsoft installation, i.e. `./ilcsoft-install -i release/v01-17/release-scratch.cfg`

After having encountered my problem I have also try to build dd4hep standalone using the correct ILCsoft.cmake.

I think I did everything as it should be.

It might be that you have never encountered the problem since the system wide root installation on the system you were installing always complied with the requirements of dd4hep. In this case you will not notice (you may want to check, which root version dd4hep actually uses in your case). Anyway the problem can be solved by the user by installing a proper root version system wide (or by my patch). When you install dd4hep/ilcsoft on a multi-user system the vulnerability continues to exist.

That said I have further problems with dd4hep

In file included from /srv/ilcsoft/v01-17-03/DD4hep/v00-04/DDCore/include/DD4hep/L CDD.h:14,
from /srv/ilcsoft/v01-17-03/DD4hep/v00-04/DDCore/src/Volumes.cpp: 10:
/srv/ilcsoft/v01-17-03/DD4hep/v00-04/DDCore/include/DD4hep/Handle.h:108: error: integer
constant is too large for Å type

where the line in question in Handle.h is

```
108 inline unsigned long magic_word() { return 0xFEEDAFFEDEADFACE; }
```

Maybe this goes along with the following warning

CMake Warning:

Manually-specified variables were not used by the project:

```
BUILD_32BIT_COMPATIBLE  
INSTALL_DOC
```

... and with the cmake instructions in

```
***** Installing software *****
```

```
#####  
#####  
##### Compiling DD4hep version v00-04...  
#####  
#####
```

```
----- adding additional commands to build_env.sh :
```

```
----- adding additional command to build_env.sh test -r ${G4ENV_INIT} && { cd $(dirname  
${G4ENV_INIT}) ; . /$(basename ${G4ENV_INIT}) ; cd $OLDPWD ; }
```

+ Generated cmake build command:

```
$ cmake -C /srv/ilcsoft/v01-17-03/ILCSoft.cmake -DDD4HEP_USE_XERCESC="0"  
-DINSTALL_DOC="ON" -DBUILD_32BIT_COMPATIBLE="OFF" -DDD4HEP_WITH_GEANT4="1"  
-DCMAKE_BUILD_TYPE="RelWithDebInfo" -DDD4HEP_USE_PYROOT="0"  
/srv/ilcsoft/v01-17-03/DD4hep/v00-04
```

...

All this comes from the official ilcsoft installation.

I should say that I have indeed still a 32 bit machine. I don't know why dd4hep switches BUILD_32BIT_COMPATIBLE to OFF.

Maybe the following output is also helpful

```
*** ERROR in module [ DD4hep ]: DEBUG INFO: {'PLATFORM': 'Linux - Scientific Linux CERN
SLC release 5.10 (Boron)', 'GCC_VERSION': 'gcc (GCC) 4.1.2 20080704 (Red Hat 4.1.2-54)',
'LSB_RELEASE': 'LSB Version:\t:core-4.0-ia32:core-4.0-noarch:graphics-4.0-ia32:g
raphics-4.0-noarch:printing-4.0-ia32:printing-4.0-noarch\nDi stributor
ID:\tScientificCERN\nDescription:\tScientific Linux CERN SLC release 5.10
(Boron)\nRelease:\t5.10\nCodename:\tBoron', 'UNAME': 'Linux pc-91231.lal.in2p3.fr
2.6.18-371.1.2.el5 #1 SMP Wed Oct 23 09:42:05 CEST 2013 i686 i686 i386 GNU/Linux'}
```

Thanks a lot and cheers,

Roman

Subject: Re: Installation of ilcsoft v01-17-03 - Bug in DD4hep installation?!
Posted by [rosemann](#) on Tue, 29 Oct 2013 14:51:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Roman,

coming back to the original problem:

poeschl wrote on Wed, 16 October 2013 09:20[...]It appears that the FindROOT.cmake in dd4hep finds my system wide root installation, which indeed doesn't contain the requested include files (why not is my problem but this doesn't matter here). When installed from *scratch* dd4hep should however always use the root version which got installed through the corresponding ilcinstall scripts.

[...]

I think one has to remedy this vulnerability, i.e. the FindROOT.cmake has somehow to be told which root version to use instead of starting to search on its own. It is very likely that on a system an additional root version is installed.

...

This /shouldn't/ be any problem, so I still don't completely understand how this is happening.

Citing from the cmake documentation: Quote: CMake searches for a file called "Find<package>.cmake" in the CMAKE_MODULE_PATH followed by the CMake installation

So if you've set/extended the CMAKE_MODULE_PATH prior to calling the individual installation, then the ROOT version specified there is used. If you use the ilcinstall method, this is done automatically.

The following explanations come to my mind:

- you didn't install ROOT with ilcinstall
- there is a bug with your specific CMake version (which is not unheard of)

If you could provide me please (directly or pointers):

- the install config file (release-scratch.py)
- release-versions.py
- the OS

Cheers,
Christoph

Subject: Re: Installation of ilcsoft v01-17-03 - Bug in DD4hep installation?!
Posted by [poeschl](#) on Sat, 14 Dec 2013 10:51:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christoph,

I just want to inform you that at least the 32-64 bit issue is solved by.

```
inline unsigned long magic_word() { return 0xFEEDAFFEADFACE; }  
->  
inline unsigned long magic_word() { return 0xFEEDAFFEADFACE; }
```

I have found the hint on the web

bytes.com/topic/c/answers/137121-problems-64bit-integer-constant .

The other issue is still open for me but I can work so we may look at it at another occasion.

Thanks for your support and cheers,

Roman

Subject: Re: Installation of ilcsoft v01-17-03 - Bug in DD4hep installation?!
Posted by [gaede](#) on Sat, 14 Dec 2013 13:10:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Roman,

thanks for your bug fix report. It seems that Markus had fixed this silently already 5 weeks ago. So the HEAD version should build also on 32-bit systems.

Cheers, Frank.

Subject: new ilcsoft developers release v01-17-05
Posted by [rosemann](#) on Fri, 07 Feb 2014 11:54:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

we have prepared another patch release in the v01-17 cycle of developers versions, namely v01-17-05

A lot of fixes and changes have been introduced, please take a look at the already compressed release notes below.

Use the ilcinstall tool with the appropriate configuration files in order to install ilcsoft v01-17-05.

As announced, we have added another OS in the list of reference installations, Ubuntu 12.04 in 64bit. This is experimental for the time being and needs still some work.

Reference installation in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc41_sl5/v01-17-05 # SL5 64bit  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-05 # SL6  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc46_ub1204/v01-17-05 # Ubuntu 12.04 64bit
```

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

Christoph, for the ilcsoft team.


```
#####  
#  
# iLCSoft release - v01-17-05  
#  
#####
```

Developers release that provides many smaller fixes and a number of new features as well as a number of new packages.

=====

Changes in v01-17-05 w.r.t. iLCSoft v01-17-04

=====

=====

MarlinTrkProcessor v01-10

=====

- added FPCCDTracking code by Tatsuya Mori

=====

MarlinUtil v01-08

=====

- improved FPCCD classes (T.Mori)
- added HelixClass_double

=====

Kaldet v01-13 (by F. Gaede)

=====

- modified ILDVXDKalDetector:
 - modified to optionally change the relative position of the measurement surface within the sensitive layer
=> needed for example in the FPCCD
 - optionally users can add
<parameter name="relative_position_of_measurement_surface" type="double" value="0.15"/>
to the VXD parameter section in the GEAR file

=====
dd4HEP v00-06
=====

2014/02/04 Frank Gaede

- fixed environment scripts thisdd4hep.sh and this\${PackageName}.sh to also export library pathes for external libraries (CLHEP, LCIO, gear)
- added some simple tests to ILDExDet, ILDExSimu and noviceN04
 - provide cmake script run_test_package.sh (configured as run_test_\${PackageName}) to call test binaries after initializing the environment (needed for automated/nightly tests)
- updated Readme.txt wrt build procedure and running the tests

2014/01/17 Markus Frank

- Prepare implementation of Alignment using Release and Grap features of TGeoVolume and TGeoNode. In principle this was done in a backwards compatible way, but I cannot really check 100% the #ifdef's.
- Fix handling of XML collections using XercesC. (Bug reported by P.Kostka).
- Fix several compiler warnings issued by Coverity.

2013/12/17 Markus Frank

- Whole load of fixes to keep coverity quiet....
- Fix bug in DDG4 field interface component
- Allow the usage of units, positions and lorentzvectors and in DDG4 properties. Example using a position property (ROOT::Math::XYZVector):
Idle> /ddg4/Gun/position (1*mm,1*cm,0.01*m)
Geant4UIMessenger: +++ Gun> Setting property value position = (1*mm,1*cm,0.01*m)
native:(1 , 10 , 10).
- Support for std::vector<T>/std::list<T>/std::set<T> properties.
- Support for include directives of the format
<include ref="file-name"/>
for the top level <lcdd> tags
Parent tag Included root tag Data content

```

-- <lccdd>    <define>      See below.
                <detectors>
                <detector>
                <display>
                <lccdd>      Nested includes (with DTD validation for Xerces-C)
-- <define>    <define>      Set of constants
-- <detectors> <detector>    Single XML subdetector description
-- <detectors> <detectors>  Multiple XML subdetector descriptions
-- <display>   <display>     Set of visualization attributes

```

DTD validation is only performed using Xerces-C for root tags <lccdd>.

Note:

<lccdd> include directives are executed BEFORE any other directive!

Other include directives are executed BEFORE the corresponding named directive!

2013/11/14 Markus Frank

- Reformat files in DDCode and DDG4 to remove the <TAB>s from the source files. Hopefully the content is now more readable.
- DDCore: Remove empty and named constructors for shapes. Empty constructors were simply a dangerous hook to memory leaks, named constructors are deprecated since along time.
- DDG4: add support for messengers to access and modify properties from the Geant4 prompt. For the test hit classes and the simple sensitive detector a ROOT I/O plugin was provided.
- DDG4: Split into several libraries:
 - libDD4hepG4.so pure link library without any plugins
 - libDD4hepG4Legacy.so 'Old' plugins. The old stuff at some point will have to disappear.
 - libDD4hepG4Plugins.so New framework plugins
 - libDD4hepG4LCIO.so LCIO output plugin for new framework

2013/11/03 Markus Frank

- doc: Add CHEP2013 paper about DD4hep as a start of the documentation section. More to come hopefully.
- DDCore: several small improvements:
 - Segmentations are no longer Ref_t's.
 - Base internal implementing object directly on the segmentations classes from Christian.
 - Rearrangement of some code from Handle.h to Primitives.h

- Allow to attach extensions to LCDD
- DDG4: Fix Frank's simulation problem, which he caused himself introducing his famous factories....
- DDG4: First attempt to fix simulation problem with mignetic field. Ready for testing.
- DDG4: First commit of a independent simulation aimed for LHCb and LHeC + other newcomers. Documentation needs to be written.

Features:

- Modular setup of all Geant4 action routines as plugins, including the Run-, event-, tracking-, stacking-, stepping- and generator-actions as well as the physics lists with particle and physics constructors.
- Setup is possible in XML or with C++ either as an AClick in root or standalone. Python based starter to come.
- Missing is a modular description of the G4 UI and VIS manager(s).
- Documentation is to come. Before documentation is not present the usage is discouraged.

Note:

- This version of DDG4 uses boost::spirit. It will compile without the BOOST include files and the old functionality is preserved, but it will not work.

To use BOOST add to the cmake statement the following tags:

```
cmake <regular tags> \
```

```
-DDD4HEP_USE_BOOST=ON \
```

```
-DBOOST_INCLUDE_DIR=${BOOST_INCLUDE_DIR} ....
```

Where BOOST_INCLUDE_DIR should point to the boost header directory. BOOST libraries are not used.

=====

ROOT v05-34-10

=====

- initially forced by dd4HEP requirement
- now uses git

=====

CEDViewer v01-07-02 (by F. Gaede)

=====

- some minor improvements to pyced
- added userDraw() method for drawing additional objects
 - added example for track cut function

=====
MarlinFastJet v00-02 (by A. Sailer)
=====

* Add optional feature to store PFOs clustered into physics jets in separate recParticleOut collection, which can be passed to later processors (S. Redford)

* Add support for FastJet v3

* bug fixes

=====
MarlinReco v01-08-01 (by G.Grenier)
=====

SDHCAL digitizer : charge simulation : simplifies Polya function. Charge
dispatching on pads : fix integration if using functions (slows the process)
and add new dispatching using sum of Erf functions (recommended option)

=====
GDML v03-00-02 (by J. McCormick)
=====

forced change, older version disappeared

=====
Gear v01-04
=====

- added set methods to SiPlanesLayerLayout (I.Rubinsky)
- bug fix for FixedPadAngleDiskLayout::getNearestPad (A.Muennich)
- Wrap around of angle was not done correctly.

=====
GeneralBrokenLines (GBL) v01-16-03
=====

- documentation augmented
- cmake config file exports ROOT dependencies
 - diagonalization of scatters
 - trajectories require at least two points

=====
pathfinder v00-06 (by I. Heinze)
=====

=====

- cleaned up code & directories , fix typos
- added doxygen comments everywhere
- added class HoughSpaceAnalyser to save plots of Hough spaces
- added class HoughMapMaximumCandidateFinder, which does the search of the Hough maps for the maxima, before this was done in every kind of Hough map
- added a class called HoughMap2DuntIntersectionMethod, which is not used yet, but it is an idea for further development
- fixed bug in TrackGenerator

=====

MarlinTPC v00-16

=====

(A. Munnich, P. Hayman, C. Kleinwort, F. Mueller, Li Bo, Ch. Krieger, D. Sankar, Ch. Rosemann)
-- lots of bug fixes

(P. Hayman)

- Addition of MM reconstruction (see log for details)

(A. Munnich)

- Added a new flag for the pulses with a plateau. They were previously flagges as anomolous shape. Added a electron particle gun for our LP1 test beam geometry in preparation for detailed simulation.
- First version of analysis processor for dEdx. Needs some more testing and tweaking probably. + root scripts
- several updates to detailed simulation (check logs for details)
- fixes in ModuleDistortionCalculator.
- Major rewrite of the HitDistributor (detailes see below). It should now work more correctly. Full testing not finished. Charge fluctuation on pads still seem somewhat large, but could also be due to electronics processor.

Modified: HitDistributor called from ChargeDistributionProcessor

- Fixed errors, e.g. list of hit pads was wrong
- Rewrite of code for clearer structure, avoiding the same code repeating in different

functions

- Introduced correct handling and usage of TPCModule
- Introduced local and global coordinate systems. Now also translated and rotated modules

work

Added:

- test for the behaviour of the HitDistributor
- test for the new TPCGas class

(Ch. Rosemann)

Changed BField access method from strange to

- 1) initialization of data by ConditionsProcessor
- 2) loading of the field with the LoadFieldFromConditionsDataProcessor [takes as option the coil coordinate system]
- 3) getting it with

```
gear::Vector3D bfield = marlin::Global::GEAR->getBField().at( gear::Vector3D  
arbitrary_position);
```

NOTE: needs a b field map; still to be loaded into the database

C. Kleinwort

- StepWiseHelixGeneralBrokenLineInterfaceProcessor (following changes in Bz) implemented. Needs track reference point inside magnet. RowTripletBasedTrackFinderProcessor uses as default now 1. hit as reference point.
- A local helix with bending plane perpendicular to an arbitrary magnetic field is now used by StepWiseHelixGeneralBrokenLineInterfaceProcessor to propagate in an changing (only locally constant) field

(Li Bo)

- Added modulesNotUsed in TrackMakingKalmanFilterProcessor

(Ch. Krieger)

Changed TimePixMaskCreatorProcessor so it does not need any more an occupancy histogram (which was required but not used before), dead and noisy pixels are now defined via processor params instead of setting them in the processors source code

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/ilcinsta>

ll/tags/v01-17-06/?op=dl&isdir=1&rev=3051&peg=30 51]

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/v01-17-0 6 # 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.Killenberg)

=====
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 DD4hepSegmentation 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 wireframe 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 Polyesterene 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 lcio 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 DDAAlign - 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 separate and nearly identical implementations for various objects.
 - Improved definition of basic objects describing Conditions and Alignment.
 - Improved implementation of DDAAlign, 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 available 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 drive && 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.Chaus, K.Zenker, Ch.Krieger, Bo Li

- Various bug fixes
- And added functionality

(M.Rogowski)

- Added: TimePixHoughTransformNormalProcessor. Track finding using a Houghtransformation

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 methode 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 MMAAdvancedCalibration 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.

=====
PandoraAnanlysis 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)

Subject: CMake with local package
Posted by [jfstrube](#) on Wed, 05 Nov 2014 05:46:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm trying to build LCFIPlus, which in turn depends on LCFIVertex.
I've checked out both packages, because I'd like to work on both of them.
How can I tell CMake to look for my local copy of LCFIVertex when building LCFIPlus rather than the one in \$ILCSOFT?

I could probably hack together something for the CMakeLists.txt, with a lot of googling.
If possible, I would much prefer a solution based on command-line options.
Otherwise, some hints on how to hack the CMakeLists.txt are also appreciated.

<edit>I've tried `cmake -C /home/ilc/jfstrube/public/LCFIVertex/LCFIVertexConfig.cmake -C /group/ilc/soft/ilcsoft/x86_64_gcc44/v01-17-06/ILCSOft.cmake ..`, but without success. Changing the order of the cmake files did not help.</edit>

Subject: Re: CMake with local package
Posted by [sailer](#) on Wed, 05 Nov 2014 06:35:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

This should work (did not test it though):
`cmake -C /group/ilc/soft/ilcsoft/x86_64_gcc44/v01-17-06/ILCSOft.cmake -D`

LCFIVertex_DIR=/home/ilc/jstrube/public/LCFIVertex/ ..
-C Loads a cache file
-D changes /defines variables. and Find_package (LCFIVetrtext) should then look in the LCFIVertex_DIR.
Otherwise LCFIVertex is found via the CMAKE_PREFIX_PATH defined in the \$ILCSOFT/ILCSOft.cmake

<http://www.cmake.org/cmake/help/v3.0/manual/cmake.1.html>

Edit: corrected statement about where LCFIVertex is found.

Subject: Re: CMake with local package
Posted by [jfstrube](#) on Wed, 05 Nov 2014 06:38:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Terrific!
I had tried LCFIVertex_ROOT before, as suggested on stackexchange.
LCFIVertex_DIR works.
Thank you.

Subject: Re: iLCSoft release v01-17-06
Posted by [grenier](#) on Tue, 03 Feb 2015 11:42:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I'm trying to install that release on my ubuntu 12.04 machine.

It fails in ROOT installation not finding Python which is installed on the machine.

I've run
./ilcsoft-install -i releases/v01-17-06/release-ubuntu-base.cfg

and I got :

```
#####  
#####  
##### Compiling ROOT version 5.34.18...  
#####
```


#####

...
...

Checking for Python.h ... no
Checking for python2.7, libpython2.7, libpython, python, or Python ... no
Explicitly required Python dependencies not fulfilled

*** ERROR in module [ROOT]: DEBUG INFO: {'PLATFORM': 'Linux - Ubuntu 12.04.5 LTS', 'GCC_VERSION': 'gcc (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3', 'LSB_RELEASE': 'Distributor ID:\tUbuntu\nDescription:\tUbuntu 12.04.5 LTS\nRelease:\t12.04\nCodename:\tprecise', 'UNAME': 'Linux lyopc608 3.5.0-54-generic #81~precise1-Ubuntu SMP Tue Jul 15 04:02:22 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux'}

*** ERROR in module [ROOT]: failed to configure!!

However, when I execute locate libpython2.7, I got
/usr/lib/libpython2.7.so.1
/usr/lib/libpython2.7.so.1.0
/usr/lib/python2.7/config/libpython2.7.so
/usr/share/doc/libpython2.7
/usr/share/lintian/overrides/libpython2.7
/var/cache/apt/archives/libpython2.7_2.7.3-0ubuntu3.6_amd64.deb
/var/lib/dpkg/info/libpython2.7.list
/var/lib/dpkg/info/libpython2.7.md5sums
/var/lib/dpkg/info/libpython2.7.postinst
/var/lib/dpkg/info/libpython2.7.postrm
/var/lib/dpkg/info/libpython2.7.shlibs
/var/lib/dpkg/info/libpython2.7.symbols

So the right python is installed.

Any clue on how to solve this issue ?

Regards

Subject: Re: iLCSoft release v01-17-06
Posted by [oschaefer](#) on Tue, 03 Feb 2015 11:47:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

If it is missing the header file, it could be that you only have the python library itself installed. There are special packages (with ending -dev) for developing in Ubuntu and Debian. These packages have the header files which are necessary if you need to link against a library. As you don't need the headers anymore for running the finished program, there are packages with library-only.

Have a look in aptitude or google for details.
Maybe that is already the solution.

Subject: Re: iLCSoft release v01-17-06
Posted by [grenier](#) on Tue, 03 Feb 2015 11:56:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks,

That was it, I had to install python-dev

Regards

Subject: Re: iLCSoft release v01-17-06
Posted by [grenier](#) on Tue, 03 Feb 2015 13:40:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I'm now facing problems to checkout software from
svn checkout [svn://svn.freehep.org/lcdet/projects/](http://svn.freehep.org/lcdet/projects/)

Is there a need to update the server name or is it momentarily down or is it some kind of unsecure site that I won't be able to access from behind a typical lab firewall ?

Thanks for your help.

Regards

Subject: Re: iLCSoft release v01-17-06
Posted by [tonyj](#) on Tue, 03 Feb 2015 15:17:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

The server and URL are working fine for me. What is the error you get? The server is hosted at SLAC, and just requires that access to the svn port (3690) is allowed.

Tony

Subject: Re: iLCSoft release v01-17-06
Posted by [grenier](#) on Tue, 03 Feb 2015 15:26:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for the answer,

Apparently, the firewall at my local site is blocking that kind of svn connection.

I'll retry it from a different network tomorrow.

Regards

Subject: Re: iLCSoft release v01-17-06
Posted by [jfstrube](#) on Thu, 19 Feb 2015 22:23:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm trying to install release-ilcsoft.cfg from ilcinstall.
I managed to get through all of release-base.cfg, but when trying to install the remaining files after that, it dies.

+ Following modules will be used for the installation:

```
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/xercesc/3.1.1 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/xercesc/3.1.1 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/CED/v01-09-01 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/CED/v01-09-01 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/FastJet/3.1.1 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/FastJet/3.1.1 ]
```

```
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/root/5.34.18 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/root/5.34.18 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/CLHEP/2.1.1.0 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/CLHEP/2.1.1.0 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/gsl/1.14 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/gsl/1.14 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/QT/4.7.4 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/QT/4.7.4 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/geant4/9.5.p02 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/geant4/9.5.p02 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/cernlib/2006b ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/cernlib/2006b ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/mysql/usr ] -> [ /usr ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/CMake/2.8.5 ]
+ [ /home/stru821/ILC/ilcsoft/v01-17-06/ilcutil/v01-02 ] -> [
/home/stru821/ILC/ilcsoft/v01-17-06/ilcutil/v01-02 ]
+ [ /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.55.x86_64 ]
```

```
===== End of Installation Summary:
=====
```

```
***** Starting installation *****
```

```
***** Starting ILC Software installation process *****
```

```
***** Creating symlinks *****
```

```
*** ERROR in module [ XercesC ]: DEBUG INFO: {'PLATFORM': 'Linux - Scientific Linux release
6.5 (Carbon)', 'GCC_VERSION': 'gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4)', 'LSB_RELEASE':
'LSB Version:\tbase-4.0-amd64:base-4.0-noarch:core-4.0-amd64:cor e-4.0-noarch\nDistributor
ID:\tScientific\nDescription:\tScientific Linux release 6.5
(Carbon)\nRelease:\t6.5\nCodename:\tCarbon', 'UNAME': 'Linux belle2-user.hep.pnnl.gov
2.6.32-431.17.1.el6.x86_64 #1 SMP Wed May 7 23:32:49 UTC 2014 x86_64 x86_64 x86_64
GNU/Linux'}
```

```
*** ERROR in module [ XercesC ]: could not create link to [
/home/stru821/ILC/ilcsoft/v01-17-06/xercesc/3.1.1 ]
```

in [3.1.1]!!!

I guess somehow the system tries to create cyclic links, which fortunately fails. I'm assuming this has to do with how the different paths are set, but I cannot figure out how to fix this.

Any help as to which path variable should point where would be much appreciated.

File Attachments

1) [release-ilcsoft.cfg](#), downloaded 256 times

Subject: Re: iLCSoft release v01-17-06
Posted by [sailer](#) on Fri, 20 Feb 2015 07:23:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

If I read the things correctly, then the problem is that the link is created in the very same place that the folder already exists, and it cannot overwrite the folder.

So just replace "link" by "use"

Change this:

```
ilcsoft.link( XercesC( ilcPath + "xercesc/" + XercesC_version ))
```

To this:

```
ilcsoft.use( XercesC( ilcPath + "xercesc/" + XercesC_version ))
```

Subject: Re: iLCSoft release v01-17-06
Posted by [gaede](#) on Fri, 20 Feb 2015 09:55:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Jan,

the idea is that you make the base installation outside of any versioned release.

For this you will have to change the installation pathes accordingly.

The following is what I use:

base installation:

```
ilcsoft_install_prefix = "/data/ilcsoft/"
```

```
# don't add version: ilcsoft_install_dir = os.path.join( ilcsoft_install_prefix, ilcsoft_release )
```

```
# ----- ilcsoft home -----  
# python variable for referring the ILC Home directory  
# used to link or use already installed packages ...
```

```
ilcPath = "/data/ilcsoft"
```

-> all external base tools will be installed in "/data/ilcsoft/"

and then for the actual ilcsoft installation:

```
ilcsoft_install_prefix = "/data/ilcsoft/"
```

```
# add version:
```

```
ilcsoft_install_dir = os.path.join( ilcsoft_install_prefix, ilcsoft_release )
```

```
# ----- ilcsoft home -----  
# python variable for referring the ILC Home directory  
# used to link or use already installed packages ...
```

```
ilcPath = "/data/ilcsoft"
```

Cheers, Frank.

PS: you might have to adjust some of the pathes for truely external packages like mysql, etc.

Subject: Re: iLCSoft release v01-17-06
Posted by [jfstrube](#) on Fri, 20 Feb 2015 17:19:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

That part was totally not obvious to me, but it makes a lot of sense to set it up like this.
Thank you for the explanation.
Seems to work now...Cool

Subject: crash when compiling DD4hep in ILCSoft-01-17-6
Posted by [fabsalva](#) on Fri, 26 Jun 2015 13:01:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I'm having some issues with the DD4hep package when I try to install v-0-17-06 of the ILCSoft packages. The issue seems to be in the constructor for DDG4/src/Geant4ParticleHandler (see below). I've tried to look for the documentation of this package (see <http://www-f9.ijs.si/~marko/DD4hep/32.html>) and indeed I see differences between what is on the WEB and what is in the ILCSoft release, but maybe this is intentional...

Has anyone got the same problem or have any idea how to solve it?

Thanks a lot for any help!

Regards,

Fab

=====

```
Scanning dependencies of target DD4hepEveRootmap
[ 58%] Generating libDD4hepEve.rootmap
-- *** MakeROOTMap.cmake run command :
/research/epp/ilc/software/ilcsoft/v01-17-06/root/v5.34.30/b in/genmap -debug -i libDD4hepEve.so
-o libDD4hepEve.rootmap
      WORKING_DIRECTORY
/research/epp/ilc/software/ilcsoft/v01-17-06/DD4hep/v00-10/b uild/lib
Input Library: 'libDD4hepEve.so'
ROOT Map file: 'libDD4hepEve.rootmap'
Output directory: '.'
[ 58%] Built target DD4hepEveRootmap
```

```
[ 59%] Building CXX object DDG4/CMakeFiles/DD4hepG4.dir/src/Geant4ParticleHandler.cpp.o
/research/epp/ilc/software/ilcsoft/v01-17-06/DD4hep/v00-10/D
DG4/src/Geant4ParticleHandler.cpp: In constructor Ã¸:
/research/epp/ilc/software/ilcsoft/v01-17-06/DD4hep/v00-10/D
DG4/src/Geant4ParticleHandler.cpp:70: error: Ã¸ was not declared in this scope
make[2]: *** [DDG4/CMakeFiles/DD4hepG4.dir/src/Geant4ParticleHandler.cpp.o] Error 1
make[1]: *** [DDG4/CMakeFiles/DD4hepG4.dir/all] Error 2
make: *** [all] Error 2
```

```
*** ERROR in module [ DD4hep ]: DEBUG INFO: {'PLATFORM': 'Linux - Scientific Linux release
6.6 (Carbon)', 'GCC_VERSION': 'gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-11)',
'LSB_RELEASE': 'LSB Version:\t:base-4.0-amd64:base-4.0-noarch:core-4.0-amd64:cor
e-4.0-noarch:graphics-4.0-amd64:graphics-4.0-noarch:printing
-4.0-amd64:printing-4.0-noarch\nDistributor ID:\tScientific\nDescription:\tScientific Linux release
6.6 (Carbon)\nRelease:\t6.6\nCodename:\tCarbon', 'UNAME': 'Linux feynman
2.6.32-358.14.1.el6.x86_64 #1 SMP Tue Jul 16 14:24:33 CDT 2013 x86_64 x86_64 x86_64
GNU/Linux'}
```

```
*** ERROR in module [ DD4hep ]: At least one of these files:
['/research/epp/ilc/software/ilcsoft/v01-17-06/DD4hep/v00-10 /lib/libDD4hepCore.so',
'/research/epp/ilc/software/ilcsoft/v01-17-06/DD4hep/v00-10/ lib/libDD4hepCore.dylib']
is required for using this installation of DD4hep
```

Subject: ilcsoft release v01-17-07
Posted by [vouts](#) on Fri, 26 Jun 2015 13:19:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new developers release of ilcsoft is available: v01-17-07.

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 v01-17-07

<https://svnsrv.desy.de/websvn/wsvn/General.ilctools/ilcinstall/tags/v01-17-07/?op=dl&isdir=1&rev=3170&peg=3170>

We have improved the software installation with ilcinstall - please see ./ilcinstall/README for details.

As usual the reference installations in afs are available at:

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-07 # SL6 gcc 4.4  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-17-07 # SL6 gcc 4.8  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc46_ub1204/v01-17-07 # Ubuntu 12.04 64bit
```

and cvmfs at:

```
/cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-07 # SL6 gcc 4.4
```

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

The ilcsoft team:
Frank, Shaojun, Yorgos

```
#####  
#  
# iLCSoft release - v01-17-07  
#  
#####
```

=====

Changes in v01-17-07 w.r.t. iLCSoft v01-17-06

=====

Introduces 5 new package:

=====

DD4KalTest v00-01

=====

by F. Gaede

- DDKalTest - interface between KalTest fitter and DD4hep based geometry

=====
ILDPerformance v00-01

=====
by Y. Voutsinas & Alberto Guerrero Llorrente

Collection of scripts and steering files for creating canonical ILD performance plots

=====
AidaTT v00-01

=====
by C. Rosemann et al

- tracking toolkit developed in the AIDA project
- implementation of GBL fitting using a DD4hep model that provides DDRec::Surfaces
- implementation of multiple scattering still under validation.

=====
DD4hepExamples v00-01

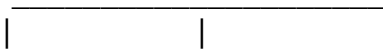
=====
by F. Gaede & M. Frank

- standalone package build from the examples in DD4hep
- mostly needed for nightly builds

=====
MarlinDD4hep v00-01

=====
by F. Gaede

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



| Updated packages |
|_____|

=====
CEDViewer v01-09
=====

- use cluster position for drawing orientation of clusters (pointing to IP) if ITheat,IPhi are 0.0 (F. Gaede)
- fixed drawing of strip hits: now based on bit ILDTrkHitTypeBit::ONE_DIMENSIONAL in type (was collection name) (F. Gaede)
- Modifications to template and gear file for proper functionality with DSTViewer (M. Petric)
- fixed layer id description (F. Gaede)

=====
Clupatra: v00-11
=====

by F. Gaede

- add option to use DDKalTest
 - parameter "TrackSystemName"
- added parameter "SetSideForTPC" to allow for setting the side to +/- 1
- added new parameters
 - FTDCollectionName: set cellIDs for FTD
 - SetSideForVXD: set side to +-1 if true
 - LayerOffset: offset applied to layer number (ZPalanar)
- added SET hits (used for debugging mainly)

=====
DD4Hep v00-12
=====

2015/06/17 Markus Frank

- Add possibility to block access to constants by name from LCDD.
Functionality enabled by a constant named "LCDD_InhibitConstants" with value "1", "True", "Yes".

Test: geoDisplay -compact
file:../DD4hep.trunk/examples/ClientTests/compact/InhibitConstants.xml

- Add LCDDHelper handle object to easily access the sensitive detector object of a detector using either the subdetector name or the detector element (or one of its children).

See: DD4hep/LCDDHelper.h
Test: geoPluginRun -input file:../DD4hep.trunk/examples/CLICSiD/compact/compact.xml \
-plugin CLICSiD_LCDDHelperTest \
optional: -<detector-name (default:SiVertexEndcap)> [Note the '-!!!]

- Check when adding new sub-detector detector elements if a given one already exists.
if yes: exception.

Test: geoDisplay -compact
file:../DD4hep.trunk/examples/ClientTests/compact/DuplicateSysID.xml

A. Sailer

- DDSegmentation: Added function cellDimension which returns the dimension of any given cell

S.Lu:

- Added a switch for BirksLaw to Geant4StepHandler.
For backward compatible, by default, do not apply BirksLaw, and apply BirksLaw with call 'doApplyBirksLaw()'.
added optional Geant4ScintillatorCalorimeterAction

- Added one 'Geant4ScintillatorCalorimeterAction' into DDG4 plugins Geant4SDActions for scintillator to take
Geant4 BirksLaw effect as sensitive detector active layer.

F.Gaede

- added SurfaceManager class that holds maps of surfaces for
"world", "tracker", "calorimeter", "passive" and all individual subdetectors
- to be used in reconstruction code where a lookup from cellID
to corresponding surface is needed

- needs plugin "InstallSurfaceManager" to be in the compact file
- access via:


```
SurfaceManager surfMan = *Icdd.extension< SurfaceManager >() ;
const SurfaceMap& surfMap = *surfMan.map( "world" ) ;
```
- renamed old SurfaceManager to SurfaceHelper

- added method: Geometry::Volume createPlacedEnvelope()
 - to be used in all LC detector drivers to create a placed envelope volume
 - rest of the detector is then instantiate in this volume

- added BooleanShape plugin for creating generic boolean shapes (subtraction, union or intersection)
 - to be used for envelope definition in xml

- added new build_type: BUILD_ENVELOPE
 - can be used to visualize envelopes only (geoDisplay, teveDisplay,...)

2015/05/22 Nikiforos Nikiforou

- Fixed bug in DDG4/lcio/LCIOConversions.cpp, which stored only the PDG of the primary particle in multiple particle contributions in SimCalorimeterHits instead of both the primary and secondary PDGs.

2015/05/13 Markus Frank

- Fix bug in DDG4, which forced DDG4 to exit if a property from the Geant4 prompt could not be set. No an error message is printed and the prompt is returned to the user. (Bug submitted by A. Sailer)

2015/05/11 Andre Sailer

- CMake updates:
 - Split libraries into components: use find_package(DD4hep COMPONENTS <component> [...]) to find the components you need.
 - At the moment there are these components are sensible to link against: DDRec, DDG4, DDEve, DDSegmentation
 - To link against the librarie use either DD4hep_COMPONENT_LIBRARIES or DD4hep_<COMPONENT>_LIBRARY, where <COMPONENT> needs to be replaced by the UPPER case name of the component

2015/05/09 Markus Frank

- Allow to access detectors by type from lcdd.
 - The sensitive type of a detector is set in the 'detector constructor'.
 - Not sensitive detector structures have the name 'passive'
 - Compounds (ie. nested detectors) are of type 'compound'
- Dump detector types using plugin:
geoPluginRun -plugin DD4hepDetectorTypes -input <compact-file>

2015/03/12 Markus Frank

- Add support for ellipsoids in gdml/lcdd and geant4 conversion.
- Allow to include files with environment variables from compact notation.

2015/03/09 Markus Frank

- Finalize move to geant4 version 10, started by A.Sailer
- Add non-isotrop particle generator in DDG4 (request P.Kostka)
- Isolate ROOT plugin manager as start to move to ROOT 6

=====

ForwardTracking v01-08

=====

- modified ForwardTracking (F.Gaede):
 - use a constant map in getOverlapConnectionMap
(was filled with unneeded empty hit vectors)
 - added processor parameters:
TrackSystemName: KalTest, DDKalTest, ...
GetTrackStateAtCaloFace: allow to run w/o Ecal surfaces
- no change in algorithm

=====

Garlic v3.0.3

=====

by D. Jeans

- fix to pseudo-layer calculation (for last ECAL layer)
- fix calculation of ECAL layer positions for older mokka models (mokka-08-04 and earlier)
- implemented energy-dependent cluster merge distance cut

=====

KalDet v01-13-02

=====

- added some debug printout (F. Gaede)

=====

KalTest v02-00

=====

- implemented the optional use of a non-uniform B field (Li Bo) as described in
Bo Li, Keisuke Fujii, Yuanning Gao, "Kalman-filter-based track fitting in non-uniform magnetic field with segment-wise helical track model"
[<http://arxiv.org/abs/1305.7300>]

=====

KiTrack v01-06

=====

- adding criterio for minimum PT to mini-vector tracking (Y. Voutsinas)

=====

KiTrackMarlin v01-06

=====

- replaced hardcoded B-Field in tools/Fitter with gear::BField (R.Simoniello)

=====

LCFIPlus v00-05-03

=====

- New joint probability introduced (T. Suehara)
- added utility processor to convert Track to ReconstructedParticle (T. Tanabe)
- Durham with beam jets added (T. Suehara)

=====

LCFIVertex v00-07

=====

- allow for using external boost (F. Gaede)

=====

Marlin: v01-06

=====

- added geant library if built w/ dd4hep (F. Gaede)

=====

MarlinTrk v02-00

=====

by F. Gaede

- added implementation of MarlinDDKalTest:
KalTest implementation of MarlinTrk working
with DD4hep surfaces and DDKalTest

- changed Marlin(DD)KalTestTrack::initialise(const EVENT::TrackState& ts, double bfield_z,
bool fitDirection)

 - to ignore the bfield_z parameter and get it from the first hit

 - fixes issue in MarlinTrkUtils/finaliseLCIOTrack() where the method above is used
but the bfield is not available

 - should consider changing IMarlinTrkTrack interface ...

 - use new LCDD detector lists to create DDKalDetectors only
for "tracker" and "passive" sub detectors,

i.e. calorimeters are ignored

- made MarlinTrk::Factory a singleton
 - sub sequent calls to init() can only change the options (MSQ,Eloss)
 - Factory caches one IMarlinTrkSystem of every type
- USERS SHOULD NO LONGER DELETE THE IMarlinTrkSystem POINTER IN THEIR CODE (Marlin processor)
- fixed MarlinTrkUtils::createFinalisedLCIOTrack for fitting insode out (forward):
 - smooth back to last constrained hit
 - add innermost hits with a proper Kalman filter step to get track state at the IP
 - added fit_direction to finaliseLCIOTrack() et al
 - made treatment of fit_direction more consistent
-
- added an IP layer to MarlinDDKalTest (DDCylinderMeasLayer w/ smallest radius)
 - debug printout in MarlinTrkUtils
-
- ignore missing track state at calo face (for now)
 - > just print warning
 - added debug printout

=====
MarlinTrkProcessors v02-00
=====

- adapted to DD4hep::Surfaces and DDKalTest
- DDCellsAutomatonMV: adapted CA to DD4hep (R.Simoniello)
- modified RefitProcessor (F.Gaede)
 - InitialTrackState: allows to select one of the track states (or use hits)
 - FitDirection: allow to fit forwards or backwards (default)

- > can use for CLIC SiBarrel tracker to refit TrackState AtIP in forward direction
- adapted to new MarlinTrk::Factory: no longer delete the IMarlinTrkSystem pointer at the end of the processing for all fitters
- ExtrToTracker: add new processor that extracts an existing track to another tracking subdetector (R.Simoniello)
- added DDPlanarDigiProcessor which uses the DDDRec::Surface for smearing the hits (no gear needed) (F.Gaede)
- add option to use DDKalTest to all fitters (TrackSubsetProcessor, FullLDCTracking_MarlinTrk, TrackSubsetProcessor, RefittProcessor) (F.Gaede)
 - set parameter "TrackSystemName" to "DDKalTest"
- PlanarDigiProcessor: add new utility processor that allows to split a collection of hits into several collections, based on the layerID of the hits (works for all Lcio hit types) (F.Gaede)
- ExtrToSIT: added processor for refitting and propagating track from TPC or VXD to SIT (Y.Voutsinas)
- reverted changes for dEdx calculation in FullLDCMarlinTrk: moved to new processor (M.Kurata)

```

=====
MarlinKinfit v00-01-05
=====

```

- J. List
 - adjusted version number in CMakeLists.txt, removed obsolete PConstraint, added directory not_used and moved there all the Track / Vertex fitting stuff which had only recently (r4750) been transferred from H1 version and is now being overhauled by Daniel Jeans.

First version after refactoring of FitObject inheritance tree (D. Jeans)
 Minimized duplication of code in derived classes at the price of a
 overall limit on the maximum number of parameters, to be set in BaseDefs.h/cc
 As a consequence, the logic of the copy constructors / assignment operators

had to be adjusted.

=====
LCTuple v01-04
=====

F. Mueller, C. Duerig, A. Ebrahimi

- Added JetBranches which holds branches created from a jet collection created by Satoru Jet Finder processor.

JetBranches has two steerable parameters which are optionally set in the steering file:

* writeExtraParameters: if set to TRUE, the fill function calculates some extra parameters using parameters provided by the jet finding algorithm. Default value is FALSE.

See class data members for a list of extra parameters.

* writeTaggingParameters: if set to TRUE, the fill function creates branches for jet parameters coming from flavor tagging processor. If no flavor tagging processor is used, this should be set to FALSE to prevent errors. Default value is FALSE.

See class data members for a list of extra parameters.

- Added isolated lepton branches
- Added MC particle from relation branches
- Added MC particle remove overlay branches

=====
MarlinPandora v02-00-00
=====

by J. Marshall

Update to reflect change to calo hit creation API.

Allow for names "layer" and "stave" in cellID encoding string.

TAG v01-00-01

Improve specification of CMAKE_CXX_FLAGS.
Handle cases where c++11 is not available.

TAG v01-00-00

Milestone v01 release.
Update to reflect recent changes to PandoraSDK.
Improved const-correctness. Algorithms are now only exposed to pointers to const managed-objects. Access to managed-object non-const functionality is now only via Pandora APIs.
Use LCContentFast if c++11 is available: contains faster versions of algorithms in the LCContent library, using e.g. KD-trees and relying on c++11.
Corrected radiation length and nuclear interaction lengths specified for Pandora CaloHits and Pandora SubDetectors.
Restored PandoraSettingsBasic.xml to repository.
Added to list of cheated neutral hadrons in PandoraSettingsPerfectPhotonNeutronK0L.xml
Updated PandoraSettings readme.
CMakeLists overhaul.

TAG v00-15

Update to reflect recent changes to PandoraSDK and LCContent.
Overhaul, with emphasis on ensuring that client applications can instantiate and use multiple concurrent pandora instances.
Address issues highlighted by Coverity.

Improved member variable initialization, with default constructors for settings and local algorithms.

Move some re-usable plugins from MarlinPandora into LCContent.

Update PandoraSettings files, adding new CaloHitPreparation algorithm, using new configuration for plugins and improving list management.

Add HCAL Ring geometry details (missing from gear file; ring is treated as an endcap, so symmetry order is not specified)

=====
MarlinReco v01-11
=====

LDCCaloDigi/ILDCaloDigi: D.Jeans & O. Hartbrich

- improved realism of silicon and scintillator/PPD calorimeter hit digitisation
- ScintillatorPpdDigi class defines PPD response model
- default behaviour is no realism: must be switched on in steering file
- added documentation (also described in LC-TOOL-2014-011)
- can regroup the virtual cells of a scintillator strip

Clustering/hybridEcalSplitter: D. Jeans

- can deal with scintillator strips with >1 virtual cells

- S.Bilokin: new package TruthVertexFinder added
(see ./Analysis/TruthVertexFinder)

- M.Berggren: added new packages TrueJet and TrueJet_Parser
(./Analysis/TrueJet, ./Analysis/TrueJet_Parser)

- F.Gaede: add parameter DontEncodeSide to TPCDigiProcessor
- allows to process old Mokka and new DD4hep based simulation

- F.Gaede: modified calo digitizers in CaloDigi/LDCCaloDigi:
ILDCaloDigi, NewLDCCaloDigi, SimpleLCalDigi, SimpleLHCalDigi, SimpleMuonDigi
added parameters

CellIDLayerString, CellIDModuleString, CellIDStaveString

to allow to switch between old cellId encoding string

using K/K-1,M,S and new one using layer,module,sensor

(additionally added CellIDIndexIString, CellIDIndexJString to ILDCaloDigi)

- C.Calanca: new package Analysis/FourMomentumCovMat

- M.Kurata: new package Analysis/PIDTools: TPC track dEdx calculation, Cluster shower profile extraction, and Likelihood Particle Identification

=====
MarlinTPC v01-00
=====

Major changes! (O. Schaefer)

Restructured build mechanism of MarlinTPC and TPCCondData, such that the latter is now a subproject and can be built stand alone with way less dependencies. The MarlinTPC build works the same as in the past and automatically triggers the build of TPCCondData. Details are specified in the cmake_README and README files.

There were some dependencies introduced from TPCCondData on MarlinTPC (Exceptions). These were rectified. Such dependencies can't be reintroduced in future, as in that case the build of TPCCondData, and thus MarlinTPC, will fail. TPCCondData doesn't know about MarlinTPC anymore, but vice versa.

The build procedure for documentation was also affected. If the option `-DINSTALL_DOC=ON` was supplied to the cmake command, there is the possibility to execute "make TPCCondDatadoc". "make doc" was already in use for MarlinTPC documentation, so this other name was chosen instead.

Several formal errors were discovered in the documentation. Due to a former mistake in the script `createProcessor.py`, comments would be placed above the "namespace" keyword and be included into the namespace documentation. I corrected for that in the script and the affected processors. Additionally there is now a header file "general/MarlinTPCNamespaceDoc.h" to contain the namespace documentation in a central place. A similar header exists for TPCCondData. In the affected processors also a credit was placed, indicating the usage of the `createProcessor.py` script. This should allow to track possible errors in the script and affected processors in future.

Furthermore, all compiler warnings were removed. There were "ambiguous else statements", "variable set but not used", "signed/unsigned comparison" and "not used" warnings. I corrected these and added TODO comments, for the programmers to check, if these corrections didn't alter the meaning of the code.

- Many other developments - bug fixes

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

- added functions for cluster shape calculation (M. Kurata)

=====
MOKKA mokka-08-05
=====

What is new in this Mokka release
=====

- patch by A.Sailer, CERN
made compatible w/ Geant4 10.1
Changes:
 - G4EmSaturation() constructor now takes verbosity as mandatory argument (default used to be 1)
 - RunManager previousEvents changed from std::vector to std::list (which does not have operator[], but front() both of them have)
 - Added "CLHEP::" to some units, which were not known otherwise
- S.Lu, DESY:
Added a new driver TubeX02, same as TubeX01. Added userInitString Beam_Residual_Gas_Material,
and allow user to update the beam gas material at mokka run time. The available beam gas materials will be defined in the mokka database.
- D.Jeans:
 - bug fix in ./source/Kernel/src/LCAscHepRdr.cc
 - the vertex (and time) info which is read in a .hepevt file, is completely ignored in the current code,
and doesn't make it into the lcio MCParticle, the added correction reads vertex information
 - updated ./source/Geometry/LDC/src/SEcal05.cc
added additional parameters to the ECAL description in the gear file including also for the preshower

- E. Becheva:
Adapt Mokka for working with geant4.10
(many changes - almost all drivers modified)

=====
PandoraAnanalysis v01-00-01
=====

by J. Marshall
Fix production of output root files when using multiple instances of the PfoAnalysis processor.

TAG v01-00-00

Milestone v01 release.
Added full set of calibration tools for automated calibration of PandoraPFA in ilcsoft.
Added calibration documentation.
Reorganised PandoraAnalysis test applications into calibration and performance directories.
Tweaked MC Pfo selection mechanisms for filling Pfo analysis tree.
Addressed issues highlighted by Coverity.
CMakeLists overhaul.

TAG v00-07

Unify MC Pfo selection with approach used within Pandora.
Simplify use of MC particles in pfo analysis, removing need for separate input MCPfo and QuarkPfo collections.
Address issues highlighted by Coverity.
Remove some unused functionality.

Fix for MIP direction-correction.

=====
PandoraPFANew v02-00-00
=====

=====
PhysSim v00-02
=====

- a major update including more processes and features; see detail of update in my talk at ILD Analysis/Software Meeting, <https://agenda.linearcollider.org/event/6599/> (J. Tian)

=====
Gear v01-04-02
=====

- F. Gaede
 - fixed a few warning wrt unused parameters (F. Gaede)
 - made some methods public in FTDPParametersImpl
 - fix in ZPlanarParametersImpl::isPointInLadder for missing shell (shell parameters zero)

=====
lcgeo v00-04
=====

- ddsim renamed to lcgeo
- many changes and developements:
 - first 'complete' prototypes of ILD_o1_v05 and CLIC_o2_v01 simulation models
 - introduced envelopes for all sub detector
 - added examples/ddsim.py for running the simulation

- plugins for adding surfaces to the CLIC model

=====
LCIO v02-06
=====

- patched C++ documentation (F.Gaede):
 - create a search field on the doxygen page
 - allow to create doxygen only (no latex and/or html)
- added method ParticleIDVec UTIL::PIDHandler::getParticleIDs() (F.G.)
- improved printing of ParticleIDs for ReconstructedParticles (F.G.)
(e.g. in dumpevent)
- use Jaxen runtime in siodump (J.McCormick)
- Include Jaxen runtime dep so siodump does not barf. (J. McCormick)

File Attachments

1) [release_notes_ilcsoft_v01-17-07.txt](#), downloaded 274 times

Subject: Re: ilcsoft release v01-17-07
Posted by [fabsalva](#) on Fri, 26 Jun 2015 14:17:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Frank, all,

could you please advise how to use the version that is installed in cvmfs? We have cvmfs running on our system for ATLAS and it would certainly be a lot easier to link that one, rather than to install everything from scratch. Is there some config file that one should run and are there instructions on how to link e.g. a Marlin processor to all the rest of the libraries in the version that is installed in cvmfs?

Thanks!

Fab

Subject: Re: ilcsoft release v01-17-07
Posted by [sailer](#) on Fri, 26 Jun 2015 14:32:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

fabsalva wrote on Fri, 26 June 2015 07:17Hi Frank, all,

could you please advise how to use the version that is installed in cvmfs? We have cvmfs running on our system for ATLAS and it would certainly be a lot easier to link that one, rather than to install everything from scratch.

You need to add the /cvmfs/ilc.desy.de repository to your machine(s)

If you have root access you can try it with these steps (<https://twiki.cern.ch/twiki/bin/view/CLIC/CLICcvmfs>) at cern, but no guarantee if this works for you. You can probably start at "Config cmvfs ", because cvmfs is already installed and running.

Quote:

Is there some config file that one should run

Once you have the cvmfs running there is an init_ilcsoft.sh script in the folder of ilcsoft.

`/cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-07/init_ilcsoft.sh`

Quote:

and are there instructions on how to link e.g. a Marlin processor to all the rest of the libraries in the version that is installed in cvmfs?

Just as usual, after sourcing init_ilcsoft.sh

```
cd yourMarlinProcessor
mkdir build
cd build
cmake -C $ILCSOFT/ILCSoft.cmake [plus your other options] ..
make install
```

Cheers,
Andre

Subject: Re: ilcsoft release v01-17-07
Posted by [fabsalva](#) on Fri, 26 Jun 2015 14:40:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks a lot for your reply.

We link quite happily with cvmfs at CERN, so I hope that, if there are not permission/firewall issues at Desy, we should be able to link to the Desy one too. I've asked our sysadmin to look into it.

I have been quite into the ILC software some time ago, but I am now a 'newbie' to the more recent updates, so I'm not sure what you mean when you say 'just as usual'. Is there some 'User Guide' or 'Reference Guide' that one can use to navigate in all this?

Thanks a lot!

Fab

Subject: Re: ilcsoft release v01-17-07
Posted by [sailer](#) on Fri, 26 Jun 2015 14:47:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Usual in the sense that an installation on cvmfs is not different than your own or one from afs.

This contains explanations on how to create you own processor

<https://svnsrv.desy.de/viewvc/marlin/Marlin/trunk/README?view=markup>

Cheers,
Andre

Subject: ilcsoft v01-17-07 and gcc48 with c++11
Posted by [gaede](#) on Wed, 01 Jul 2015 11:14:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

as reported in the previous thread, we now also provide 'experimental' versions of ilcsoft that use c++11 (and gcc4.8).

Reference installations can be found in afs and now also in cvmfs and can be initialized with:

```
source /afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-17-07/init_ilcsoft.sh
source /cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-07/init_ilcsoft.sh
```

In order to build your own package against these, you need to have a compatible compiler available or use the ones in afs/cvms provided by CERN SFT.

These can be instantiated with

```
source /afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/init_gcc48.sh
source /cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/init_gcc48.sh
```

When building your own package make sure you set the c++11 flag at the configure command, e.g.

```
cmake -C $ILCSOFT/ILCSoft.cmake -D CMAKE_CXX_FLAGS='-std=c++11' ..
```

Please report any problems with using these ilcsoft installations here.

Cheers, Frank.

Subject: Re: ilcsoft release v01-17-07
Posted by [fabsalva](#) on Mon, 06 Jul 2015 13:33:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Andre,

we have been able to install cvmfs here at Sussex, so we are now testing that we are able to run a Marlin processor to make some simple histograms.

We follow the instructions you sent us to create a Marlin directory and then compile in the build dir and so on, but one problem we are finding is that the compiler complains that it cannot find the 'TH1F.h' include file from ROOT.

Before doing anything we source the /cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-07/init_ilcsoft.sh and we have all the various envvars defined (ROOTSYS, LD_LIBRARY_PATH, etc), but we noticed that there is no mention

in the Makefile of the ROOT include dirs. I was expecting to see something like:

```
ROOTCFLAGS=$(shell root-config --cflags)
```

in the Makefile to define where the compiler should get all the flags from ROOT, including the position of the include directory. When we list all the envvars we can see that nowhere it appears where the \$ROOTSYS/include dir is defined. Is that expected? Does it mean that somehow we need to define this within our Marlin processor somewhere?

Thanks for any help you might be able to give us.

Cheers,

Fab

Subject: Re: ilcsoft release v01-17-07
Posted by [sailer](#) on Mon, 06 Jul 2015 13:45:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Fab,

We are using cmake to create makefiles, so you have to include this in the CMakeLists.txt

In the CMakeLists.txt you have add this

```
FIND_PACKAGE( ROOT REQUIRED )  
INCLUDE_DIRECTORIES ( SYSTEM ${ROOT_INCLUDE_DIR} )  
ADD_DEFINITIONS( ${ROOT_DEFINITIONS} )  
LINK_LIBRARIES ( ${ROOT_LIBRARIES} )
```

In the second line the correct variable might also be \${ROOT_INCLUDE_DIRS} this is sometimes different

After you add this go to the build folder and run "cmake .." again.

For other dependencies one has to add similar lines.

Cheers,
Andre

Subject: Re: ilcsoft release v01-17-07
Posted by [fabsalva](#) on Tue, 07 Jul 2015 13:48:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Andre,

thanks a lot for all your suggestions! We managed to get it to work (including with AIDA).

Thanks!

Fab

Subject: ilcsoft release v01-17-08
Posted by [gaede](#) on Fri, 31 Jul 2015 12:35:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new developers release v01-17-08 is available.

It contains a lot of developments from the High Level Reconstruction Week at DESY and a number of improvements for the new DD4hep based simulation and reconstruction.

Please use the corresponding ilcinstall tag for installation:
<https://svnsrv.desy.de/websvn/wsvn/General.ilctools/ilcinstall/tags/v01-17-08/?op=dl&isdir=1&rev=3197&peg=3197>

A reference installation is available at

[/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-07 # SL6 gcc 4.4](#)

(cvmfs and other OS to follow).

There is also an ILDConfig release v01-17-08 that has steering files for running the simulation and reconstruction (both old and new) with this release:

<https://svnsrv.desy.de/public/marlinreco/ILDConfig/tags/v01-17-08>

For instructions see:

<https://svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags>

/v01-17-08/StandardConfig/current/README

and

https://svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags/v01-17-08/StandardConfig/lcgeo_current/README

Please report problems and questions related to this release in this thread.

Frank, for the ilcsoft team.

```
#####  
#  
# iLCSoft release - v01-17-08  
#  
#####
```

This is a developers release that adds some functionality to the recent v01-17-07 release.

In particular there have been many new subpackages added to MarlinReco as a result of the High Level Reconstruction Week held at DESY: <http://agenda.linearcollider.org/event/6787/>

Also some important changes have been made to the 'new' reconstruction code for DD4hep (lcgeo) based simulations.

=====

Changes in v01-17-08 w.r.t. iLCSoft v01-17-07

=====

Introduced 1 new package:

=====
DDMarlinPandora v00-01
=====

N.Nikiforou, CERN

- copy of MarlinPandora that replaces the geometry definition from GEAR with DD4hep/DDRec
- staus: experiemental

Changed Packages:

=====
MarlinReco v01-13
=====

F.Gaede

- renamed SimpleLHCalDigi to SimpleFCalDigi
- deleted SimpleLCalDigi
- => old steering files need to be updated !!

M. Berggren

- added parameter FixLCalHits SimpleLHCalDigi in order to:
 - fix of the wrong xyz for hits in LCal. LCal CaloHits (collection LCAL) are now within a few micron of their correct position. LCal SimCaloHits (collection LumiCalCollection) still have the flawed xyz.

J.List

- updated SimpleLHCalDigi to include functionality from SimpleLCalDigi
- > can be used for all fwd calos

J.Tian

- added new package:
Analysis/OverlayRemoval

H.Sert

- new processor low momentum pi/mu separation:
Analysis/PIDTools/LowMomentumMuPiSeparationPID_BDTG

G.Wilson

- new photon finder package:
Analysis/GammaGammaCandidateFinder
- provides a 4-vector w/ mass-constrained fit (using MarlinKinfit - new dependency)

M Berggren

- fixed links in RecoMCTruthLink/src/RecoMCTruthLinker.cc
- full links in both directions (true->seen, seen->>true) for PFOs, Tracks and Clusters.

| v01-12 | (v01-17-07 contained v01-11 !)

J.Tian

- added new Analysis/IsolatedLeptonTagging
MVA based isolated lepton finder

T.Calanca

- fixes for Analysis/FourMomentumCovMat
- printout and read-only exception

M.Kurata

- update PIDTools for several kinds of likelihood calculation

S.Bilokin

- added Analysis/VertexChargeRecovery

T.Suehara:

- Analysis/TauFinder added

F.Gaede

- added copy of ClusterShapes from MarlinUtil
- > use "include "ClusterShapesMR.hh"
and marlinreco::ClusterShapes
- applied to ComputeShowerShapesProcessor.cc
- eventull changes should be merged back to MarlinUtil

- fixed warning -Wc++11-narrowing in TPCDigitizer

=====

DD4hep v00-14

=====

F.Gaede

- fixed writing of generator status for `Icio::MCParticle` in `Geant4Output2LCIO.cpp`

A.Sailer

- Add missing scope to `/DD4hep/Factories.h`, to not force everyone to use using namespace

F.Gaede: 2015-07-15

- changed type in `SurfaceList` and `SurfaceMap` from `Surface` to `ISurface`
- added methods `length_along_u/v()` to `ISurface`
- moved setting of `CMAKE_CXX_FLAGS` (if `DD4HEP_USE_CXX11`) from `DD4hep.cmake` to `CMakeLists.txt`

2015-07-11 M.Frank

- added starter docs `DD4hepStartersGuide.pdf`

*** Important ***

Before updating, backup your existing and working checkout. Though I was running the basic tests, there may be unchecked corners of the software.

Notes:

- Backwards compatibility mode for the usage of ROOT 5 and ROOT 6 alternatively.
- For ROOT 6 the Gaudi PluginService is used and added to the distribution.
- DD4hep is distributed with a licence. See `$DD4hepINSTALL/LICENSE` for details.
- In the doc area the `$DD4hepINSTALL/doc/CREDITS` file everybody should add her/his name, if contributed significantly.
- Unfortunately this meant to add/change the headers of all files and give a pointer to the licence file. If I accidentally changed the author name, please change it back.

- The plugin factory declaration statements were changed to accomodate both ROOT 5 and ROOT 6 and to keep the number of 'ifdef' statements at a minimum.

- TODO: Properly change the cmake scripts to accomodate for ROOT 6 and the automatic switch when building DD4hep.

For reference reasons: this commits is revision 1812 + 1813 (DDDectors)

| v00-13 | (v01-17-07 contained v00-12)

2015/07/02 Nikiforos Nikiforou

- Added isRadiator() helper function in DDCore/XML/XMLDetector.h/cpp
Mirroring functionality of isSensitive() and used in geometry drivers
e.g. to calculate total absorber thickness
- Extended LayeredCalorimeterStruct in DDRec/DetectorData.h to include
additional parameters required by Pandora:
 - inner_phi0 (e.g. alignment for inner symmetry)
 - outer_phi0 (e.g. alignment for outer symmetry)
 - NOTE: phi0 still remains but is deprecated
 - gap0 (e.g. stave gap along r-phi)
 - gap1 (e.g. middle stave gap along z)
 - gap2 (reserved for future use)

2015-06-29 A.Sailer

- always store lcio collections, even if they are empty in Geant4Output2LCIO (DDG4/ddsim)

2015-06-29 F.Gaede

- changed env scripts to prepend to library pathes (DD4hep and dependant packages)
-> using a newer lcgeo version than the one provided in ilcsoft can simply
be done w/ source ./bin/thislcgeo.sh

=====
DD4hepExamples v00-14
=====

- no changes wrt. v00-12

=====
lcgeo v00-05

=====

N. Nikiforou

- Added TrackerEndcap_o1_v02_geo.cpp (svn cp'd from TrackerEndcap_o1_v01_geo.cpp) which supports
ILD ID encoding and implements ring modules. Elements (enumerated as "sensors") in one ring
(constant R) have the same "module" number but different sensor number.
Also modified the CLIC_o2_v02 model compact (InnerTracker_o2_v01_01.xml and OuterTracker_o2_v01_01.xml) to use the new driver. Also now implements the "side" bitfield
side=0 is positive, side=1 is negative.
- Introduced new driver VertexEndcap_o1_v02 and necessary changes to compact file for CLIC_o2_v02.
The new xml also implements double layers in the endcap but with each layer having its own ID.
Also, in the new VertexEndcap driver similar changes as above VertexEndcap.

S.Lu

- updated Ecal and Hcal in Share_ILD_o1_CLIC

T.Quast

- fixed DetectorData structure for several drivers
(as needed for drawing the detector w/ DD4hep in CED)

M. Petric

- added CLIC_o2_v02 model w/ simplified Ecal barrel
ECalBarrel_o2_v01_01.xml in order to set correct W thicknesses

F.Gaede

- added TPCSDAction.cpp
- ported from Mokka/TPCSD04.cc
- introduces Geant4 dependency
-> to be addressed ...
- activated in TPC10_geo.cpp
-> hits should be exactly on pad row centers ...
- updated ddsim.py :
- use Geant4ScintillatorCalorimeterAction

- as defaults for all calorimeters
- use TPCSDAction for TPC

A.Sailer

- Implemented FieldMapBrBz.cpp, based on the 2D Fieldmap of Mokka FieldX03
- Example XML for the fields section of the compact XML

```

<field name="DetectorMap" type="FieldBrBz"
  filename="${lcgeo_DIR}/fieldmaps/ILDMap_KB_20150204_BRhoZ.root"
  tree="fieldmap:rho:z:Brho:Bz"
  rScale = "1.0"
  zScale = "1.0"
  bScale = "1.0"
  rhoMin = "5*mm"
  zMin = "5*mm"
  rhoMax = "7005*mm"
  zMax = "7005*mm"
  nRho = "701"
  nZ = "701"
  >
</field>

```

- added ./fieldmaps/ILDMap_KB_20150204_BRhoZ.root latest field simulation for ILD by K.Buesser

- requires BOOST

N. Nikiforou

- added EcalBarrelFace_v00.xml/EcalEndcapFace_v00.xml to CLIC_o2_v01
- Enabled detailed shower mode for calorimeters by default in ddsim.py/DD4hepSimulation.py

F.Gaede

- added PolyhedralBarrelSurfaces_geo.cpp/PolyhedralBarrelSurfaces_geo.cpp
- to be used for track states at the calorimeters
- added as EcalBarrelFace_v00.xml/EcalEndcapFace_v00.xml to ILD_o1_v05

```

=====
DDKaITest v00-02
=====

```

F.Gaede

- changed to just use abstract ISurface and ICylinder

=====
aidaTT v00-02
=====

-F.Gaede:

- fixed major memory leak in Vector5 and fiveByFiveMatrix (unneeded alloc in assignment operator)
- made c'tors and accessors in Vector5 and fiveByFiveMatrix more efficient by removing unneeded range checks and initializations

-Y.Voutsinas:

- adding test for functions calculateCurvature & calculateLambda
- correcting transformation from perigee to L3

=====
MarlinTrk v02-00-01
=====

F.Gaede

- made createTrackStateAtCaloFace() work for DD4hep (and old Mokka/Gear) based tracking

=====
MarlinTrkProcessors v02-01
=====

R.Simoniello

- updated Refitting/src/ExtrToTracker.cc
- extrapolation also in endcap subdetectors + CA selecting longest (nhits) tracks

F.Gaede

- handle empty collections gracefully (for lcgeo/ddsim) in Digitisers/src/PlanarDigiProcessor.cc
- use abstract ISurface

Clupatra v00-12

=====

F.Gaede

- added parameters CaloFaceBarrelID/CaloFaceEndcapID to ClupatraProcessor (needed for lcgeo based simulations)
default value for both : ILDDetID::Ecal
- adopted LCIOTrackConverter to use these paramters

=====

KiTrackMarlin v01-07

=====

F.Gaede

- added workaround to Tools/Fitter.cc for getting the correct system ID of the calorimeter face (barrel and endcap) if run w/ lcgeo output (F.Gaede)
-> depends now on DD4hep

=====

MarlinUtil v01-10

=====

J.List

- added helper function in CHT
CHT::CaloType caloTypeFromString(const std::string& name)

F.Gaede

- fixes in MarlinCED
- fixed warning -Wc++11-narrowing
- draw inner and outer edge of FTD disks in wireframe

=====

Marlin v01-07

=====

F.Gaede

- introduced global flag: AllowToModifyEvent
allowing to modify LCIO collections that have
been read from an input file in the processEvent()
method

=====

CEDViewer v01-10

=====

T.Quast, RWTH Aachen

- add DrawDetectorDD4hep
- general detector (CLIC & ILD) drawing for the event display without GEAR
- adapted ced2go to transparently use DD4hep if a compact file is given with -d instead of a Gear file
- add option to ced2go to draw tracking surfaces from DDRec (-s)

=> depends now on DD4hep/Root

v01-09-01:

- M.Petric:
 - updated ced2go template files for new CLIC simulation model

Subject: Re: ilcsoft release v01-17-08
Posted by [zarnecki](#) on Thu, 06 Aug 2015 19:43:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Frank,
Dear developers,

Thank you for all your work and the new release. Unfortunately I need to ask you for some help/guidance...

It is already some time since I installed ilcsoft by myself. I decided to install new release on the server at our Faculty. It is a new machine with AMD processors running SLC 6.5. It was basically "clean", nothing installed except for OS...

The hardest step was to install... cernlib. Then, I tried to follow the standard procedure and it worked OK. I had to go back few times (eg. to recompile cernlib with -fPIC flag), but I could understand what the problems are. However, I am now stacked at compiling ForwardTracking inside MarlinReco. I get the following message:

Linking CXX executable bin/param_runner_background

```
Linking CXX executable bin/CritRunner
Linking CXX executable bin/param_runner
Linking CXX executable bin/QuantileAnalyser
/scratch/zarnecki/ilcsoft/v01-17-08/geant4/10.01/lib64/libG4 OpenGL.so: undefined reference to
`QListData::detach(int)'
/scratch/zarnecki/ilcsoft/v01-17-08/geant4/10.01/lib64/libG4 OpenGL.so: undefined reference to
`QListData::detach_grow(int*, int)'
collect2: ld returned 1 exit status
make[2]: *** [bin/param_runner_background] Error 1
make[1]: *** [CMakeFiles/param_runner_background.dir/all] Error 2
```

I have found a hint that I should probably add QtXml library to the list of linked libraries. But I was not able to find a way to check it. I am not an expert in cmake etc. If you have any hint, suggestion, I will be very grateful.

By the way, it would be nice to have a list of ilcsoft prerequisites prepared. I started from installing what is needed for root (described nicely on root web pages), but it turned out that there is much more. Some requirements seem to be not obvious (eg. latex and ghostscript).

Regards
Filip Zarnecki

Subject: freehep.org website missing and svn server down?
Posted by [arummler](#) on Thu, 28 Jan 2016 16:07:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Okay, just learned that there was a power outage and that they are trying to get everything running again.

Hello,

it seems that in the moment the svn server at svn.freehep.org is not available. Is this a scheduled maintenance or who shall we contact?

Kind regards,
Andre

Subject: Re: freehep.org website missing and svn server down?

Posted by [NormanGraf](#) on Thu, 28 Jan 2016 16:13:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Andre,
That is correct. It should be back up later today.
Norman

Subject: iLCSoft release v01-17-09
Posted by [gaede](#) on Sat, 20 Feb 2016 12:37:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new developers release v01-17-09 is available.

It contains a lot of developments for the new DD4hep based simulation and reconstruction - for details see release notes below.

Please use the corresponding ilcinstall tag for installation:

download:

<https://svnsrv.desy.de/websvn/wsvn/General.ilctools/ilcinstall/tags/v01-17-09/?op=dl&isdir=1&rev=3230&peg=32>

or checkout:

svn co <https://svnsrv.desy.de/public/ilctools/ilcinstall/tags/v01-17-09>

Reference installations are available at

[/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-09](#) # SL6 gcc 4.4

[/afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-17-09](#) # SL6 gcc 4.8 c++11

(cvmfs and other OS to follow).

There is also an ILDConfig release v01-17-09 that has steering files for running the simulation and reconstruction (both old and new) for ILD with this release:

svn co <https://svnsrv.desy.de/public/marlinreco/ILDConfig/tags/v01-17-09>

For instructions see:

<https://svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags/v01-17-09/StandardConfig/current/README>

and

https://svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags/v01-17-09/StandardConfig/lcgeo_current/README

Please report problems and questions related to this release in this thread.

Frank, for the ilcsoft team.

```
#####  
#  
# iLCSoft release - v01-17-09  
#  
#####
```

This is a developers release with many changes wrt. v01-17-08, particularly to DD4hep and lcgeo, tracking and PFA code...

=====

Changes in v01-17-09 w.r.t. iLCSoft v01-17-08

=====

Introduced 1 new package:

```
=====  
MarlinKinfitProcessors v00-01  
=====
```

J.List

moved example processors from MarlinKinfit to this new package in order to reduce cross-dependencies with MarlinReco (eg via GammaGammaFinder).

Changed Packages:

=====

LCIO v02-07

=====

N.Graf

- Added support for new momentumAtEndpoint feature in Java

F.Gaede

C++ :

- added MCParticle::getMomentumAtEndpoint()
- removed long deprecated methods
 - MCParticle::getNumberOfParents()
 - MCParticle::getParent(int i)
 - MCParticle::getNumberOfDaughters()
 - MCParticle::getDaughter(int i)
- fixed many compiler warnings for clang
- added example for setting particle id to recjob
- added IDs for coild and calo endcaps to UTIL/ILDConf.h
- fixed setting of CMAKE_INSTALL_PREFIX to allow installation in 'usr/local'
- fixed incorrect file names printed at the end of anajob
- fixed compiler warnings for clang and c++11 in TESTs:
non-constant-expression cannot be narrowed from type 'double' to 'float' in initializer list
[-Wc++11-narrowing]

T.Bisanz

- made UTIL::LCTime and UTIL::LCStdHepRdr compatible with gcc 5.3.1

=====
Gear v01-05
=====

- Vector3D.h: fixed compiler warnings of type: declaration of 'phi' shadows a member of 'this'
- added search field to doxygen doc

=====
CED v01-09-02
=====

- M /CED/trunk/src/server/glced.cc
- replace << std::cout w/ std::endl (bug for c++11)

=====
DD4hep v00-15
=====

2016-02-15 A.Sailer

- LCIOStdHepReader.cpp
- fixed memory leak, compiled and valgrinded

2016-02-10 F.Gaede

- added utility DD4hep::XML::setDetectorTypeFlag() to set the TypeFlag from xml element <type_flag type="0x42"/>
- select detectors with
DD4hep::Geometry::DetectorSelector(lcdd).detectors((DD4hep::DetType::TRACKER | DD4hep::DetType::ENDCAP));
- used in UtilityApps/dumpdetector.cc
- added detector_types.xml with int constants defined in DetType.h, include with :
<include ref="{DD4hepINSTALL}/DDDdetectors/compact/detector_types.xml"/>
- add elements to xml, e.g.
<type_flags type="DetType_TRACKER + DetType_ENDCAP + DetType_PIXEL + DetType_STRIP "/>

2016-02-10 M.Frank

DDCond

First implementation of conditions access. Required some movements of files

from DDG4 (Properties) and some changes in the core conditions implementation.
The basic infrastructure is now present. Don't know yet however how buggy it is...

2016-02-05 F.Gaede

- added DetElement::typeFlag() and DetElement::setTypeFlag(unsigned int types)
- added utility class DetType for setting detector type properties in type flag

2016-02-03 N.Nikiforou

DDDetectors

- Added plugin DD4hep_GenericSurfaceInstallerPlugin, copied from lcgco to allow installation of surfaces to any sliced detector
- cmake/Doxyfile.in
- Added DDDetectors to the sources directory so Doxygen picks up the plugin documentation

2016-02-02 F.Gaede

- set momentum at endpoint (if lcio version > 2.6) for MCParticles in Geant4Output2LCIO.cpp
- fixed printlevel in Geant4Action::print()
- removed random lines drawn in teveDisplay
- don't print DetectorData by default in dumpdetector

2016-02-02 A.Sailer

DDG4

- Implemented SimulatorFlags (DDFORHEP-17)
 - * In ParticleHandler: vertexIsNotEndpointOfParent, Stopped, leftDetector
 - * Steerable distance for vertexIsNotEndpointOfParent
 - * TCUserParticleHandler Simulator flags: DecayedInTracker, DecayedInCalorimeter

2016-01-21 A.Sailer/M.Frank

DDG4

- Improved user defined physics constructors
 - * Allows user defined global range cut
 - * Allows extending particle table (e.g. B-Hadrons unknown to geant4)

2016-01-18 F.Gaede

- fixed default c'tors in DDRec/DetectorData.h to zero initialize sub-structures (LayerLayout etc)

2016-01-13 F.Gaede

- allow the default sensitive action to be a tuple with parameter dictionary in

DDG4.py/ddsim.steer e.g.:

```
SIM.action.tracker = ("Geant4TrackerWeightedAction", {"HitPositionCombination" : 1 ,
```

"CollectSingleDeposits" : False })
- fixed endpoint in MC-truth particles

2016-01-11 F.Gaede
- fix hit momentum in Geant4TrackerAction (Geant4SDActions.cpp)

2016-01-11 A.Sailer
DDG4
- Add LargestAcceptableStep to field setup

2016-01-07 A.Sailer
DDG4
- Fix the logic for skipping of events at the start of input files: LCIO, HepMC, stdhep and HepEvt
- Event number in lcio outputfile takes skipped events into account now

2016-01-06 F.Gaede
- bug fix in DDSurfaces::ISurface for surface properties (orthogonal/parallel to z)
(affected ILD VXD)

2016-01-05 F.Gaede
- added print functions for DDRec::DetectorData objects
- used in dumpdetector:
dumpdetector compact.xml -d

2015-12-09 N.Nikiforou
UtilityApps/teveDisplay
- Modified teveDisplay to have three separate collections of surface vectors (u,v,n) instead of one collection. Now the vectors are colored differently.
- Changed signature of getSurfaceVectors() to allow selection of o,u,v,n vectors to add to the collection, added option for name and color of collection.
- Renamed surface list collections to "HelperSurfaces" and "SensitiveSurfaces".
- Changed signature of getSurfaces() to allow setting of name and color of collection.

2015-12-01 M.Frank
DDG4
- Remove some of the compilation warnings
- Remove energy cut in sensitive detectors
If you though need, use a Geant4Filter to mask the action processing
examples/CLICSID
- Add a few tests for materialscan and simulation setups using XML or CINT.

examples/ClientTests

- Add a few tests for materialscan and simulation

2015-11-24 M.Frank

DDG4

- Implementation of multi-threading mode.
To use and understand, please consult the DDG4 manual from the doc area.
 - "Old" Single threaded mode and functionality is preserved.
 - Numerous new classes supporting thread and master specific setup functionality.
 - Support for python configuration (see manual for details)
 - Support for global Geant4Actions executing in thread reentrant shared mode. (see manual for details)
 - DDG4 examples: DDG4/examples/CLICSidSimu.py deleted and replaced with:
 - 1) DDG4/examples/SiDSim.py (single threaded, old example)
 - 2) DDG4/examples/SiDSim_MT.py (multi threaded version of SiDSim.py)
- DDCore:
- Support user formats for the default printout statements for nicer printouts.
 - Removal of compiler warnings

2015-11-23 F.Gaede

- added glbal method DD4hep::versionString()

2015-11-13 S.Lu

- Added a new AHcal Barrel segmentation: TiledLayerGridXY, to be used e.g. for the ILD AHcal Barrel Sensitive layer segmentation design.

2015-10-30 F.Gaede

- fixed LCIOFileReader to use sequential reading mode
as using direct access always required runNum==0 and sequential evt nums ...

2015-10-15 F.Gaede

DDSurfaces::ISurface/Surface

- added implementation of conical surfaces
 - ICone, VolConeImpl, VolCone, ConeSurface

2015-10-13 M.Frank

DDG4

- Remove explicit constructors for modular physics lists.
Use the native Geant4 provided G4PhysListFactory instead.
The physics list is instantiated as before with the "Extends" property.

- Implement angular ranges in the Geant4IsotopeGenerator for phi [0,2pi] and theta[0,pi]
- New sensitive detector Geant4TrackerWeightedAction
Attempt to properly process combined deposits in tracking detectors.
- Debug component Geant4SurfaceTest similar to the standalone program test_surfaces, but to be appended as a DDG4 event action for event by event tests of hits.

2015-10-09 M.Frank

DDG4

- Extend the functionality of the DDG4 plugins Geant4InteractionVertexBoost and Geant4InteractionVertexSmear.
If the Mask property is set to -1, all interactions present at call time are smeared/boosted according to the parameters.
Note, that the position of the plugins within the generation sequence is important.
- Add a new generator Geant4GeneratorWrapper to allow the use of native G4 generator classes such as the G4ParticleGun or the G4GeneralParticleSource. Still to be tested.
- Extend the Geant4Random functionality: May now be instantiated like any other Geant4Action. Users may set the seed or the luxury level.
All calls to HepRandom (CLHEP) or gRandom (ROOT) will re-use this instance. See DDG4/examples/CLICRandom.py for illustration.
- Updated DDG4 manual.

2015-09-30 F.Gaede

- updated createGearForILD.cpp to use new parameters in LayeredCalorimeterData (innerThickness, innerRadLength, etc)

2015-09-15 F.Gaede

- refactoring of Surface classes:
 - made VolSurface a reference counting handle to a polymorphic pointer of type VolSurfaceBase
 - this should facilitate the creation of dedicated surface classes for special cases such as inside the beampipe or the face of the calorimeter

2015-08-27 F.Gaede

- added macros DD4HEP_VERSION_GE(MAJV,MINV) and DD4HEP_VERSION_GT(MAJV,MINV) to LCDD.h

- increased version number to v00-15

2015-08-20 M.Frank

- DDCore: Modularize/abstract plugin manager
- VolumeManager: Fix bug propagating sensitive detectors
Showed up in the examples of nested detectors

2015-08-12 N.Nikiforou

-
- DDCore/XML: Added new helper functions to Layering engine:
 - double absorberThicknessInLayer(XML::Element e) :
returns total absorber thickness in given layer
 - void sensitivePositionsInLayer(XML::Element e, std::vector<double>& sens_pos) :
provides positions of sensitive slices within a layer with respect to the center of the layer

2015-07-25 M.Frank

-
- DDSegmentation: Remove several 'shadow' warnings.
 - New build system for the DD4hep core, which greatly simplifies the cmake files for the various sub-packages.
Tested with Andre's build script doc/CompileAllOptionPermutations.sh.
(GEAR usage not tested though)

=====
lcgeo v00-16
=====

F.Gaede

- added detector type flags to xml files for ILD_o1_v05 and Simplified_ILD_o1_v05
- updated (ILD) geometry constructors to call XML::setDetectorTypeFlag(e,det)
- fixed thicknesses for surfaces in FTD (FTD_Simple_Staggered_geo.cpp)
- added surface for Be endplate of VXD shell (VXD04_geo.cpp)
- added surface for support shell barrel (VXD04_geo.cpp)
- fixed layer structure BeamCal08.xml: include graphite in first layer
- ILD_o1_v05.xml
 - fixed length of B-field

- fixed z-position of EcalEndcap caloface surface
- changed calo face layers to be constructed by plugin now
- added example reconstruction steering file for SiD: ./example/run_sid_reco.xml
- fixed material for encap helper surfaces
 - (moved origin away from centerwhich is in the beam pipe)
 - ./other/TrackerEndcapSupport_o1_v01.cpp
 - ./tracker/TPC10_geo.cpp
 - ./tracker/VXD04_geo.cpp
- added lcgeo.h defining version macros and method versionString()
- renamed DDSimExceptions.h to LcgeoExceptions.h
- added RunHeader to lcio file in lcio_particle_gun.py
- added first example implementation of SiDloi3:
 - ./SiD/compact/sidloi3/sidloi3_v00.xml
- added conical surfaces to beam pipe:
 - ./detector/other/Beampipe_o1_v01_geo.cpp
- added sensitive action CaloPreShowerSDAction
 - creates pre-shower collections for hits from first layer
 - to be used for (ILD) Ecal
- protect usage of new member variables in LayeredCalorimeterData by usage of #if DD4HEP_VERSION_GE(0, 15)
 - (added workaround for this to work with ilcsoft v01-17-08/DD4hep v00-14)
 - ./detector/calorimeter/GenericCalBarrel_o1_v01_geo.cpp
 - ./detector/calorimeter/GenericCalEndcap_o1_v01_geo.cpp
 - ./detector/fcal/BeamCal_o1_v01_geo.cpp
 - ./detector/fcal/LumiCal_o1_v01_geo.cpp

S.Lu

- ILD_o1_v05.xml
- Remove unused 'slice' from ILD calorimeters
- Added two simplified SEcal drivers by removing detail Si wafer structure in the Si layer, which could improve the performance by saving 9.4 s/event in simulation. Todo: thinking about a sensitive driver (segmentation) for virtual cell and virtual gap from guard_ring and so on:
 - ./detector/calorimeter/SEcal04_Barrel_v01.cpp
 - ./detector/calorimeter/SEcal04_Endcaps_v01.cpp
- ./detector/calorimeter/SEcal04_Barrel.cpp:
 - Added 'Ecal_fiber_thickness' and 'Ecal_Slab_shielding' thickness to the layer placement position.

- Update the Ecal wafer placement position 'wafer_pos_z' with correct dimension parameter to work together with 'Ecal_guard_ring_size'
- Remove 'slice' from the following calorimeter drivers
 - ./detector/calorimeter/Yoke05_Barrel.cpp
 - ./detector/calorimeter/Yoke05_Endcaps.cpp
 - ./detector/calorimeter/SHcalSc04_EndcapRing.cpp
 - ./detector/calorimeter/SHcalSc04_Endcaps.cpp
 - ./detector/calorimeter/SHcalSc04_Barrel_v02.cpp
 - ./detector/calorimeter/SHcalSc04_Barrel_v01.cpp
 - ./detector/calorimeter/SHcalSc04_Barrel.cpp

A.Sailer 02/02/2016

- ddsim
 - * Add ExtraParticle tables
 - * Add flag to define distance for vertexIsNotEndpointOfParent

N.Nikiforou 12/02/2016

- Changed Solenoid_o1_v01_geo.cpp and SCoil02_geo.cpp for proper filling of solenoid extent in the LayeredCalorimeterStructure for use by DDMarlinPandora

N.Nikiforou 11/02/2016

- Fixed errors due to change of namespace of "_toDouble"
- Added using of "DetType_AUXILIARY" for CLIC_o2_v04 ECalPlug and HCalRing

N.Nikiforou 10/02/2016

- Modified Tracker*_geo.cpp, Vertex*_geo.cpp, ZPlanarTracker_geo.cpp, GenericCal*_geo.cpp to add setting detector type via xml tag
- Moved CLIC_o2_v04 to using TrackerEndcap_o1_v04 drivers that set ZPetalDisksData according to minimum/maximum R and average Z of all rings in one endcap tracker disk (for DDMarlinPandora)
- Added "type_flags" tags to all CLIC_o2_v04 subdetector xml files and added include of

detector_types.xml

N.Nikiforou 03/02/2016

- Added deprecation warning for GenericSurfaceInstaller plugin which is now part of DD4hep.
- Moved CLIC_o2_v04 to using DDhep_GenericSurfaceInstallerPlugin rather than the local lcggeo implementation which is deprecated

A.Sailer 18/01/2016

- ddsim
 - * Add RangeCut parameter
 - * Add LargestAcceptableStep for magneticField (10*m)

N.Nikiforou 15/12/2015

- Switched CLIC_o2_v04 to using the GenericSurfaceInstaller. Changed therefore to using the following includes (notice _02 at the end of the name):
 - * Vertex_o2_v04_02.xml
 - * InnerTracker_o2_v03_02.xml
 - * OuterTracker_o2_v03_02.xml
- Changed the orientation of the endcap uvn vectors for Vertex and Tracker to $u(-1.,0.,0.)$, $v(0.,0.,1.)$, $n(0.,1.,0.)$, $o(0.,0.,0.)$ using the GenericSurfaceInstaller rather than the individual surface installers (for which the orientation of uvn remains as before). The vectors were set using plugin arguments in the xml. The resulting orientation puts v along R i.e. the "bad" measurement direction and u along -phi (clockwise).
- GenericSurfaceInstaller: by default all u,v,n,o vector components are now set to 0 which means that the user has to define the non-zero components as arguments, otherwise the plugin fails

A.Sailer 15/12/2016

- ddsim
 - * Add default filter with Edep > 0

N.Nikiforou 10/12/2015

- Added GenericSurfaceInstaller in detector/other to install surfaces to a given box-like volume. The u,v,n,o vectors are provided component-wise as arguments to the plugin. If the volumes are trapezoids (as in the case of the endcap) the thickness dimension is supposed to be

the same for both sides of the trapezoid (for example, $dy_1=dy_2=dy$) which is supposed to be handled

when "casting" the Trapezoid shape into a Box

- Added InnerTracker_o2_v03_02.xml and OuterTracker_o2_v03_02.xml in CLIC_o2_v04 (not used by default)

as examples using the new GenericSurfaceInstaller

N.Nikiforou 04/12/2015

- Modified CLIC_o2_v04 to use proper convention for listing slices in module composition. In this convention, when you list a layer or a slice/component in a module, the top-most element in the xml is the component that is closest to the IP (top-most is inner-most).
- Created new drivers and new xml files to use them
 - * TrackerBarrel_o1_v03 and TrackerEndcap_o1_v03 used by Inner/OuterTracker_o2_v03.xml
 - * VertexEndcap_o2_v03 used by Vertex_o2_v04_01
- Created new surface installer plugins copied from DD4hep that follow the convention discussed above

and used by the new drivers. They are listed below with the u,v,n and origin vectors as defined in

the driver coordinate system :

- * TrackerBarrelSurfacePlugin: $u(-1.,0.,0.)$, $v(0.,-1.,0.)$, $n(0.,0.,1.)$, $o(0.,0.,0.)$
 - + Assumes the driver builds the module along the original +z axis, with the top-most slice stacked perpendicular to z, closest to (0,0,0) with eventual rotations to place the module in the barrel so that the slice is inner-most. This is the case now in TrackerBarrel_o1_v03
- * TrackerEndcapSurfacePlugin: $u(0.,0.,1.)$, $v(1.,0.,0.)$, $n(0.,1.,0.)$, $o(0.,0.,0.)$;
 - + Assumes the driver builds the module along the original +y axis, with the top-most slice stacked perpendicular to y, closest to (0,0,0) with eventual rotations to place the module in the barrel so that the slice is inner-most. This is the case now in TrackerEndcap_o1_v03

A.Sailer 30/11/2015

- Added steering of filter to ddsim
 - * assign filters to sensitive detectors
 - * create your own filters
- Added dumpSteeringFile option to ddsim
 - * prints steeringFile to stdout with current values as default

A.Sailer 23/11/2015

- Added information to the lcio runHeader in ddsim. This needs the DD4hep revision 1984!
 - * The steeringFile content, commandline and resulting ddsim parameters are written
 - * geant4 and dd4hep version is written directly in dd4hep.

* user, workingdirectory and date are written as well

A.Sailer 06/10/2015

-
- ddsim python simulator
 - * "ddsim --help" to print all the command line parameters currently available
 - * example/steeringFile.py to see an example steering file with some options
 - * to enable tab-completion for ddsim: requires bash (or zsh), bash-completion, and python argcomplete
(e.g. install with: [sudo] pip install argcomplete)
in the shell run: eval "\$(register-python-argcomplete ddsim)"

N.Nikiforou 24/08/2015

-
- Modified CLIC_o2_v03/ECalBarrel_o2_v01_01.xml, ECalEndcap_02_v01_01.xml and ECalPlug_o1_v01_01.xml to avoid having layers without active element which complicates implementation for reconstruction.
This was the case for the last (outermost) dead layer (absorber only).
Instead, added one more type of layer with only one repeat which includes the extra absorber plate
 - Fixed bug (?) in CLIC_o2_v03/ECalEndcap_02_v01_01.xml and ECalPlug_o1_v01_01.xml which had different siPCBMix thickness compared to the ECalBarrel (1.20 mm -> 0.80 mm).

N. Nikiforou 19/08/2015

M.Petric

- Made a new version for CLIC named CLIC_o2_v03 which includes better representations of X0 in the Trackers

N. Nikiforou

- Added new Generic Calorimeter drivers based on Polyhedra Calorimeters named GenericCalBarrel and GenericCalEndcap. The Generic drivers were obtained by svn cp from HCalBarrel and HCalEndcap respectively and are used in the CLIC_o2_v03 model by the ECal, HCal and Yoke Endcaps, Barrels and Rings. These new drivers (almost identical to the old ones) will be used to avoid making changes in all the drivers while developing the reconstruction framework.
Changes include:
 - Cleaning up of old comments, old code snippets and features no longer used (like multiple sensitive elements per layer which will not be supported by the reco).
 - Introduction of the new LayeredCalorimeterStruct variables defined in DDRec:

- inner/outer thicknesses, nRad/nIntLengths and sensitive element thicknesses
- Avoidance of cloning of element volumes in the barrel case
- Start of counters (e.g. layer, module, stave, etc) from 0 rather than 1

=====
KalTest v02-01
=====

F.Gaede

- default measurement dimension (kSdim) set to 5
- added flag BUILD_WITH_T0_FIT (default OFF)
setting the measurement dimension to 6

- TKalDetCradle.cxx
- bug fix for intersection calculation using the newtonian solver such as cones for example
(resetting the deflection angle to 0 in TKalDetCradle::Transport())

=====
aidaTT v00-03
=====

- many changes, fixes and updates (for details, see svn -v log)

F.Gaede

- added material_ntuples.cpp to create ntuple with material properties from model and surfaces
- drawx0.C makes comparison plots for x0

- fixed scattering matrices
(setting off diagonal elements to 0.)

- implmented cone intersection
using a newtonian method

- added optional use of streamlog
- adapted lcio_tracks.cpp to (re)fit tracks including material effects
- fixed some compiler warnings
- fixed geometry initialization for examples
(added IGeometry& instance(const std::string& initString))
- added example material_effects/check_materials.cpp

- cleaned up trajectory class (documentation, code,...)
 - fixed issue w/ energy loss for jacobian
 - removed unneeded members and methods
 - surface intersections, bfield,...
- fixed issue w/ calculation of curvilinear coordinate system (use s==0.)
- pulls for central muons are now as good as w/ DDKalTest
- finished first implementation of taking energy loss into account for initial track parameters:
 - trajectoryElements now hold a track state that is valid until the next element
 - track states are computed in addMeasurement/addScatterer
 - implemented pointAt/tangentAt for trajectory with many track parameters
- moved energy loss calculation to materialutils.cc
- updated examples to using IGeometry instead of DD4hepGeometry
- some code cleanup ...
- cleaned up geometry interface:
 - made IGeometry a singleton
 - implemented access to B field map
 - added doxygen documentation
- added materialUtils.hh
 - computeQMS (needs debugging)
- added a track state to trajectoryElement
- restructured code base:
 - renamed helixGymnastics.hh to helixUtil.hh
 - removed helixHelpers.hh
 - > combine all helix relates utility functions in helixUtil.hh
 - > combine fitting related helper functions in utilities.hh
 - documented helix functions
 - created second version taking only Vector5 and Vector3D as arguments to have less overhead when used outside aidaTT
 - cleaned up definition of track parameterization
 - > access functions defined in trackParameterizationLCIO.hh
 - moved Vector3D, Vector5 and fiveByfiveMatrix to core
 - renamed helpers to util
- added implementations of Vector5 and fiveByFiveMatrix using the Eigen library for later use ...
- bug fix for intersection calculation w/ cylinders:
 - return solution w/ positive s if both solutions

- are ~equal
- bug fix for calculation of phi from x,y:
 - take abs(dphi) to be less than M_PI
- added code to treat 1D measurements (commented out for now)
- write out Millipede-2 file for debugging in GBLInterface
- restrict intersections to first half arc of the track
- GBLInterface
 - calculate only intersections with positive s and sort list
 - fix calculation of qms
- changed IFittingAlgorithm, GBLInterface and trajectory to be able to return more than one result (at different sites) for the fit

Y.Voutsinas

- calculating energy loss only in surfaces that are considered scatterers
- correction in the calculation of particle's energy
- first implementation of energy loss - still needs some debugging & testing
- modify trajectoryElement as to obtain an element that isn't a scatterer or a measurement, use IP as such an element and add it to the trajectory
- re-implement seeding and iterative fitting in the example, removing some not necessary debugging output
- modifying function intersectsWithinZCylinderBounds in order to return both solutions and not only the closest one
- correcting the sign in the calculation of qop
- fixing a bug in the calculation of the first's element arc-length & modify the lcio_tracks example in order to cope with TPC tracks
- crosschecking the system's transformations
- unitTests/finalTrackTest.cc
 - adding test to check the creation of an lcio track from an aida track
 - adding test of covariance matrix in initial track parameters test
 - test for comparing the trajectory cov. mat with the one that is written out to lcio track

=====
DDKaITest v00-03
=====

F.Gaede

- use new (faster) standalone helix intersection methods
- workaround for hits that are smeared off the DDRec::surface in the old digitizer
- added DDConcMeasLayer
- fixed bug in energy loss and multiple scattering for non-planar surfaces (compute normal at x'ing point)
- distinguish between zplanes, disks and generic planes for sorting policy
- switch back to use SurfaceHelper class in order to get all insensitive surfaces (lost in SurfaceMap)
- fixed some compiler warnings

=====
MarlinTrk v02-01
=====

F.Gaede

- MarlinTrkUtils.cc
- fix condition for setting the z0 to 0 for calo TS for curling tracks
- generalized the subdetectorHitNumbers to work for any detector (assuming that the tracker system ids are smaller than the Ecal system id !)
- fix bug with track state at calo endcap for old tracking (use Lcio::ILDDetID::ECAL also for endcap)
- MarlinAidaTTTrack:
added a prefit to the initialize:
use up to ~25 hits for an initial fit (w/o material)
before doing the actual fit with all hits

- added IMarlinTrkSystem::name()
 - in order to allow tracking utilities to adapt to special features of fitters
- allow to fall back to IMarlinTrkTrack implementations
 - initialise() in createFinalisedLCIOTrack()
 - > for aidaTT see procedure above
 - set precision of 2nd direction v to 0. for 1D hits
 - ignore virtual surface with no material (e.g. inside the beam pipe)
 - fixed issue w/ result labels
 - fixed point labels if QMS is turned off
 - treat 1D measurements correctly
 - finalized implementation of IMarlinTrkTrack interface
 - can be used in RefittProcessor
 - added first implementation if IMarlinTrk for aidaTT
 - needs debugging ...
- MarlinDDKalTest.cc
 - fixed track state at calo creation
 - consider only DDCylinderMeasLayers as IP layers

```
=====
MarlinTrkProcessors v02-02
=====
```

R.Simoniello

- RefitProcessor
 - for each subdetector X, store nhits in fit in lcio::ILDDetID::X -1 and nhits in track in lcio::ILDDetIDX -2. Structure needed for Pandora.
 - file added for correct comparison for float and double: fpcompare.h
- DDCellsAutomatonMV/ ExtrToTracker
 - criteria based on nhits and QI for CA vertex tracks, remove the possibility of geo from gear file (option GEO1 in compilation)
 - for extrapolated tracks

F.Gaede

- DDPlanarDigiProcessor:
 - introduced parameter ForceHitsOntoSurface:
 - if true hits are projected onto the planar surface if they are outside the bounds
 - no check on distance is applied !
 - store the resolution from the length of the wafer for 1D hits

(could be treated as 2d measurements)

- RefitProcessor:
 - no longer write incomplete tracks (w/o) track state at calo
 - rely on MarlinTrkTrack's implementation of initialise() if no already fitted TrackState is used

=====
Clupatra v00-13
=====

F.Gaede

- fix for TS @calo: set z0 to 0
- protect against failure in adding a trackerhit in TrackSegmentMerger
- make track state at calo again compatible w/ old sim/rec
- fixed creation of track state at calo
- activate propagete in track fit
(-> needed for pull distributions)

=====
KiTrackMarlin v01-08
=====

F.Gaede

M /KiTrackMarlin/trunk/src/Tools/Fitter.cc

- make track state at calo compatible with old sim/rec again
- fixed creation of track state at calo

=====
ForwardTracking v01-09
=====

Rosa Simoniello

- added DDForwardTracking
- ATT: Endcap* and Sector* files for CLIC temporary added in this package instead of KiTrack till the pattern recognition for CLIC is stable. DDForwardTracking added. It contains the patter recognition for CLIC in the endcap vertex+tracker region. It is interfaced with DD4hep.

=====
MarlinUtil v01-11
=====

M.Berggren

- Added WeightedPoints3D class, a class that analyses a set of points in 3 dimensions (possibly with weights), and calculates things like covariance matrix, average position, eigen-vectors/values ...

F.Gaede

M /MarlinUtil/trunk/source/src/ClusterShapes.cc

- merge back changes from version copied to MarlinReco for HLR week at Desy:
 - added dummy error functions to ClusterShapes for MarlinPandora development (J.List)
 - cluster shape variables added for mu-pi separation (H.Sert)

T.Quast (RWTH Aachen)

- added DDMarlinCED.cc utility and detector draw routines for CEDViewer

=====
Marlin v01-08
=====

M /Marlin/trunk/source/include/marlin/Processor.h

- commented out old define for m_out (conflicts with a boost header)

=====
DDMarlinPandora v00-03
=====

N. Nikiforou 19/02/2016

- Merged changes from ILDCaloDigi (MarlinReco rev. 5432) into DDCaloDigi. It incorporates changes to ILDCaloDigi by Daniel. The introduction of parameters for the cell encoding fields was not merged, since Lcio files produced with DDSim will have the standard conventions defined in the calorimeter segmentations
- Implemented access of subdetectors by DetType flags instead of strings
- Elevated missing ECal parameters to ERROR

N. Nikiforou 17/02/2016

- Merged changes from MalrinPandora 02-02-00 up to today (7c88d10)

v00-02

F.Gaede

- enforce unique selection in `getExtension(include,exclude)`
 - throws exception if not unique
- bug fix: `eCalBarrelExtension` -> `eCalEndcapExtension` for setting `m_eCalEndCapInnerZ`
- added `DetType::FORWARD` to `excludeFlags` for main calorimeters
- added `DetType::AUXILIARY` to `excludeFlag` for LCal (exclude BeamCal)
- `DDTrackCreatorILD.cc` :
 - added dd4hep units to parameters extracted from DD4hep
 - protect against poor tracks (`ndf < 0` or missing/poor parameters)
 - added debug printout for track selection

N.Nikiforou 12/02/2016:

- Changed handling of Coil parameters: now the info is accessed from `LayeredCalorimeterData` extensions attached to the drivers (Solenoid for CLICdp, SCoil02 for ILD).
- Removed/commented out obsolete and unused auxiliary static functions

N.Nikiforou 11/02/2016 (pm):

- Major overhaul using the detector type flags through the package. No more string processor parameters to access detectors.
- Protected some more against uncaught exceptions, but still more work needed
- Added more static helper functions to access extensions by data type flags

N.Nikiforou 11/02/2016 (am):

- Modified `DDTrackCreatorILD` to use det type flags. Dropped tracker/vertex detector name parameters
- Modified `DDTrackCreatorILD` to actually access hits in fit (offset 2) which is the definition now.

N.Nikiforou 10/02/2016:

- Started using the new facility by DD4hep that allows the accessing of detectors based on their detector type flags. For now, just `DDTrackCreatorCLIC` was modified. The Marlin Processor

parameters

associated with the vertex and tracker will soon be removed.

N.Nikiforou 07/09/2015:

- Many changes that require full implementation of DDRec structures for all subdetectors
- Introduced DDTrackCreatorBase class and derived implementation classes for ILD and CLIC
- Removed references to "TPC" from DDTrackCreatorBase::Settings. Whatever was a configuration parameter has its relevant variable and Processor parameter names changed from "Tpc" to "BarrelTracker". Also, moved any TPC geometry-related variables from DDTrackCreatorBase::Settings to DDTrackCreatorILD proper.
- Added Parameter for DDPandoraPFANewProcessor: TrackCreatorName to select the relevant implementation of DDTrackCreatorBase for the detector model in use (Currently ILD or CLIC). Still need to implement a proper factory to select the proper algorithm.

=====
Mokka mokka-08-05-01

=====
F.Gaede

- patch in PhysicsListFactory.cc:
 - use G4PhysListFactory if specified list not found (allows e.g. to use FTFP_BERT_EMY)

=====
MarlinReco v01-14

M. Berggren

- Much ameliorated linking for clusters, taking back-scattering and decays in flight into account in a more correct way when finding what MCP to connect to a sim-hit.
Work-around for the inverted meaning of the vertexIsNotEndpointOfParent method in Mokka. Set processor parameter InvertedNonDestructiveInteractionLogic to activate the work-around (default is false)
More consistent debugging. All levels from 0 to 9 used.
- improved RecoMCTruthLinker

Changed weight in hit->mcparticle relation to the CaloHit energy (instead of the SimHit one). Also means the sim-hits not digitised (ie. without related CaloHit) are ignored. This also feeds through to the weights in the Cluster->MCparticle relation, which should now be more correct: correct calibration factors applied for each hit.

When doing the final relinking in problematic cases, explicitly check *only* back-scatters.

Added MCP->cluster link for the case with the fixup of the missing truth-link for the LCAL in the DBD samples. Before only cluster->MCP was filled in this case.

- Corrected the true-particle to calorimeter-hit link for the case that the original true particle has been squeezed out by the cluster-linker, typically because it was created before the calorimeter, but after the last sensitive layer in the tracker.

- AddClusterProperties:

Use the PFO energy (not the cluster) when assign the PFO 4-momentum.

Also add the sum of weights (=hit energy) to the cluster shape parameters.

(In the DBD production, this is the same as the cluster energy, but it is not guaranteed to be the case).

- Better/safer use of collection-parameters for indexing in sub-detector energy and shape-parameter arrays.

- Different/simpler(?) calculation of Cov(4mom) in PFO.

F.Gaede

- changed log level for "processing event" from MESSAGE to DEBUG8 in TPCDigiProcessor.cc
- > use Statusmonitor instead

G.Wilson

- added GammaGammaCandidateTruthFilter

New processor to select those GammaGammaCandidates (fitted pairs of Pandora PFO photons) based

on MC information indicating actual parentage from a true pi0, eta or eta' meson decaying to gamma gamma.

Output collections are called TrueGammaGammaPi0s etc.

Cuts are made on the parent meson being prompt (production vertex distance < 10 cm), and the amount of the PFO

energy coming from the MC photon following similar criteria developed in ILDPerformance/pi0.

This aspect will need some further investigation/tuning.

J.List

- improve TrueJet:
- First version that runs also over Higgs samples. No crash/obvious problem in several 1000s of events,
but more sophisticated checks pending. When running with full RecoMCTruthLink, estimate of of seen of true energy needs fix, both on standard Whizard and Higgs samples

M. Kurata

- change overall structure for particle ID in LikelihoodPIDProcessor
- change dEdx error estimation

M.Berggren:

Added AddClusterProperties processor, a processor that adds cluster position and direction (w/uncertainties) to clusters, energy error (if not already set) to clusters and 4-mom (w/ covariance) to neutral PFOs. Depends on a new MarlinUtil class (WeightedPoints3D), so for now, the processor is out-commented in CMakeLists.txt, to avoid problems if MarlinUtil is not up-to-date !

G.Wilson

- update GammaGammaCandidateFinder
 - compute and fill ReconstructedParticle covariance matrix
- Add GammaGammaSolutionFinder and DistilledPFOCreator processors
- Add some steering options - and fit quality checks
- Add 1st implementation of graph based solution finding in GammaGammaSolutionFinder - depends on BOOST

T.Calanca

- The correlations of the energy (e) with x,y,z,e were stored in the wrong position of the matrix in FourMomentumCovMat. Thx to Ali Ebrahimi.

D.Jeans

- electronics dynamic range applied before unfolding of PPD saturation in ILDCaloDigi.cc / ScintillatorPpdDigi.cc

H.Sert

- a separate algorithm was created for low P mu-pi ID in LikelihoodPIDProcessor.cc

J.Tian

- leave skipping or not an event without isolep to user in IsolatedLeptonTaggingProcessor.cc

T.Suehara

- bugfix on track charge in TrackToRecoParticleConverter

J.List

- added dummy error functions to ClusterShapesMR for MarlinPandora development, to be filled with life
 - once proper calculations are available
- made SimpleFCalDigi use MarlinUtil:caloTypeFromString(_caloType)

=====
FCalClusterer v00-02
=====

LumiCalReco:

- * Bugfixes/improvements by Bogdan Pawlik
 - Read Parameters from Gear
 - More steering parameters from xml
 - Fix to create the cluster on the proper side

BeamCalReco:

- * Bugfixes/improvements from Andrey Saprnov
 - BackgroundMethod Parametrised renamed to Gaussian
 - new BackgroundMethod Parametrised with Gaus/x parametrisation
- * More examples in the example folder
- * Bugfixes from Andre Sailer

=====
ILDPerformance v00-02
=====

- Jenny added PID subdirectory
- additional directories: JER, PID, pi0

-> for details see Changelog.txt

<https://svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDPerformance/tags/v00-02/Changelog.txt>

=====
LCFIVertex v00-07-01
=====

- same code as v00-07 but correct version number (was v00-06-02 in v00-07)

=====
LCFIPlus v00-07
=====

F.Gaede: the releases mentioned below have never been created in SVN
but they are contained in this release:

v00-06-03

- Bugfix in JetFinder (default behaviour changed at v00-06-01)
- Durham and Valencia jet algorithms with beam jets introduced

v00-06-02

- Automatic creation of JointProbability histograms implemented
- Calculation of joint probability with histograms automatically created (jprob2)
Use v03_p01 steering, vertex probability and weight files in ILDConfig
for the new joint probability.

v00-06-01?

- Added possibility to include beam jets in JetFinder (thanks to Philipp Roloff)

v00-06-00a

***** Development release: please use at your own risk *****

- 6-category training (b,c,o,bb,bc,cc) implemented
- Extracting all variables used for the flavor tagging now available

- Added MC-related input variables (MCnb, MCnc etc.) in flavor tagging to be used for training (available only if MCParticle collection is supplied)
- Added more input variables for flavor tagging

=====
MarlinKinfit v00-02

=====
J.List
moved example processors now into separate package MarlinKinfitProcessors,
added ZinvisibleFitObject,
debugged covariance matrix on fitted 4-momenta.

=====
PandoraPFANew v02-05-00

many changes - see individual packages release notes:
PandoraSDK: v02-02-00
PandoraMonitoring: v02-02-00
LArContent: v02-04-01
LCCContent: v02-02-00
ExampleContent: v02-01-02
---- previous release:
PandoraSDK: v02-00-00
PandoraMonitoring: v02-00-00
LArContent: v02-00-00
LCCContent: v02-00-00

=====
MarlinPandora v02-02-00

PfoCreator updates to set Cluster and ReconstructedParticle properties. Requires new functionality from MarlinUtil ClusterShapes class.
Changed definition of some output properties and provided some new outputs, as agreed at High Level Reconstruction Workshop in June 2015.
Update Track creator to calculate track time at calorimeter.
Bug fix added to validate track relationship information in the case where pattern recognition is cheated.

Added MarlinPandoraValidation.xml to scripts directory. This script is to be used when validating changes to Pandora.

Modified MarlinPandora.xml to show necessary settings for running of multiple concurrent pandora instances.

TAG v02-01-00

PandoraSettings updates, switch version control and source distribution to git.

=====
CEDViewer v01-11
=====

F.Gaede

- ced2go
 - increased size of tracker hits drawn for tracks
 - added optional EventSelector to ced2go/ced2go-template-DD4.xml (commented out)
- DDCEDViewer.cc
 - added printout of shown collections and event number to DDCEDViewer
 - fixed duplicate drawing of collections in DDCEDViewer
 - fixed some log levels
 - introduced flag DrawEllipsoidForPFOClusters (default is Disabled)
 - draw calorimeter hits for PFO clusters ...

T.Quast, RWTH Aachen

- ced2go
 - Allow for Marlin options '--', '-n' option removed due to default 'global.SkipNEvents=...' option
 - Extend ced2go's help message
- DDCEDViewer
 - Jet collections to be defined in DrawInLayer, jets are drawn if the collection contains the substring 'Jet'
 - use logarithmic (ROOT) color map for reconstructed clusters' energy visualization
 - energy-weighted track lengths for jet entities
 - energy-weighted track lengths for jet entities
 - Draw energy clusters as ellipsoids. Energy scaling with colors, (energy weighted) spatial extension <--> size of according ellipsoid

- Improve jet visualization for CEDViewer
- removed GEAR dependencies, merge of processors, adjusted ced2go and add skipNEvents option,
implement jet visualization (commit requires tip version of MarlinUtil)

=====
MarlinTPC v01-01
=====

F.Gaede

- replaced m_out with streamlog_out
(m_out no longer defined in Marlin)

P.Malek

- updated pulse fraction plot implementation and included plot with fraction of hits with 4 or more pulses

- added the possibility to run distortion corrections based on calculations done with Millepede II, for which the data is prepared by SimpleHelixGeneralBrokenLineInterfaceProcessor and GeneralBrokenLineInterfaceHelpers
also added SelectNthEventsProcessor (originally written by Uli Einhaus, improved by me) which allows to select a subset of events to run the distortion calculations on

- I split off the pulse shape part of PulsePerHitProcessor into it's own Processor
I also added a new plot to the PulsePerHit rootscript, which shows the fractions of hits with specific numbers of pulses vs the drift distance

A.Vauth

- Changes in the simulation of readout electronics (ADC pulses):
 - adding the possibility to include time jitter for Gaussian ADC and ALTRO ADC.
 - changed shaper function in ALTROADC from TF1 to a plain double function to save computing time - TF1::EvalPar comes with a lot of extra stuff which is run every time Eval is called, when all we really need is to calculate the function result).
 - (- and also some indentation changes in the GaussianADC.)

- calculation of hit time: added option to take neighbour pulses into account in addition to the maximum.

- Simulation of primary ionisation: more flexibility in reading in electrons-per-cluster file (e.g. no more hard-coded line numbers). Replaces all printf's and cerr's with matching streamlog_out.

- Added the possibility to include noise to simulated ADC spectra.

- One can switch between different models to describe the (options are 0=no noise, 1=gaussian, 2=landau):

When noise is included, the amount of noise to add to the ADC spectrum is taken from a Gaus distributed around zero. The with of this Gaus is chosen for each pad individually from either a Gaus or a Landau distribution around the parameter NoiseLevel (mean of the pedestal width distribution) with width parameter NoiseWidth.

- To account for the fact that horribly noisy channels would probably be excluded from the analysis of real data, a cutoff for the allowed noise level can be set.

- Fixed the indentation of a few files I'm planning to commit (decided to do it in a seperate commit, so that svn diff will still show meaningful differences for the actual changes in future commits, and not be flooded by whitespace / linebreak / ... changes).

Indentation now done according to the coding conventions as described in the MarlinTPC wiki (i.e. the result of running `astyle -s4 -C -c -M80 -U -p -k1 -A1 -S -Y` on the files...).

- Fixes/changes in MMHitTimeCorrectionProcessor

- now the readout frequency set in ReadoutFrequencyOverride is actually used

- in methods 6 and 7 (Gaussian Inflexion Point and Gaussian Mean): changing from arrays to vectors for the fitted bins, to account for the case where less than 5 bins before the maximum are available

- fixed indentation of the code

- Simulation of primary ionisation: adding option to use user-given parametrisation for the cluster density.

This was necessary since the parametrisation for TDR/P5/P10 does not match T2K gas as well (seems to be mostly due to the isobutane).

- Adding the T2K parametrisation to simulation inputs;

also including the code used to obtain it in tools/SimulationClusterDensity for potential future use with other gas mixtures.

- Simulation: bug fix in HitDistributor. A factor of 0.5 was missing in the gaussian integral for the

z-distribution calculated when longitudinal distribution in the GEM stack is switched on. This resulted in depositing a total charge factor 2 too high.

- GEM simulation: allow setting of the theta-parameter for polya gain as an optional parameter. Some minor changes to other files in addition (mostly graph/histogram names).

=====
LCTuple v01-05
=====

F.Gaede

- protect against missing collections in CreateRecoRelations.cc

S.Lu

- Implement CalorimeterHitBranches into LCTuple.

A.Ebrahimi

- Added covariance matrix histograms to jet branches

J.List

- added lctuple_DST.xml to convert current DST to root

=====
=====

Subject: DDG4 particle gun
Posted by [jeans](#) on Tue, 01 Mar 2016 06:19:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

The isotropic particle gun in DDG4 results in a distribution uniform in theta, while a truly isotropic distribution should be uniform in $\cos(\theta)$. Flat-in-theta gives a too-large number of events in the forward region. If one is an FCAL expert, the current definition is of course favourable, but for more general studies, a truly isotropic distribution would be more convenient!

(This same behaviour was also present in the Mokka particle gun.)

Would it be possible to change the definition of isotropic (in DD4hep/DDG4/src/Geant4IsotropeGenerator.cpp), or to add the option of using the "correct" definition?

Cheers,
Daniel.

Subject: Re: DDG4 particle gun
Posted by [gaede](#) on Tue, 01 Mar 2016 09:35:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel,

we are already aware of the problem and it should be fixed soon.

Cheers, Frank.

Subject: Re: DDG4 particle gun
Posted by [jeans](#) on Wed, 02 Mar 2016 06:54:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Frank,

Great!

I have another gun-related request:

The ability to smear the gun direction in a limited range of polar and azimuthal angles was a useful feature in Mokka, which seems not to be implemented in ddsim.

For example, I am now testing the new simulation model hit maps, and would like to be able to focus my simulation on a limited zone of the detector, but I cannot work out a way to do this with ddsim.

Cheers,
Daniel.

Subject: Re: DDG4 particle gun
Posted by [gaede](#) on Wed, 02 Mar 2016 09:15:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel,

we'll have to see if we can provide the old Mokka Gun functionality on a short timescale.

In the mean time, you could simply write your own "particle gun" using a python script that creates an LCIO file with MCParticles.

Have a look at

`$lcgeo_DIR/example/lcio_particle_gun.py`

this currently writes out single muons at a fixed theta, but adding any other distribution should be straight forward.

Cheers, Frank.

Subject: Re: DDG4 particle gun
Posted by [sailer](#) on Wed, 02 Mar 2016 10:48:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel,

I just added phiMin/Max thetaMinMax steering to ddsim, so if you have an lcgeo HEAD (don't forget make install) you could use these options:

```
ddsim --enableGun --runType batch -N 4 --compactFile  
CLIC/compact/CLIC_o2_v04/CLIC_o2_v04.xml --gun.thetaMin 80*deg --gun.thetaMax 85*deg  
--gun.phiMin 0.0 --gun.phiMax 1*rad
```

If you cannot update:

Create a macroFile e.g., limit.mac, with these options

```
/ddg4/Gun/PhiMin 0*deg  
/ddg4/Gun/PhiMax 90*deg
```

```
/ddg4/Gun/ThetaMin 85*deg  
/ddg4/Gun/ThetaMax 90*deg  
/ddg4/Gun/isotrop True
```

isotrop is Mandatory.

And the run (note the runType run) like this for example

```
ddsim --enableGun --runType run -N 10 --compactFile  
CLIC/compact/CLIC_o2_v04/CLIC_o2_v04.xml --macroFile limit.mac
```

Cheers,
Andre

Subject: Re: DDG4 particle gun
Posted by [jeans](#) on Thu, 03 Mar 2016 04:57:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Andr  , just what I need!

Subject: Re: DDG4 particle gun
Posted by [jeans](#) on Wed, 16 Mar 2016 07:49:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello again ddsim experts,

sorry to resurrect this subject...

With my CALICE hat on, I'd like to point out that the Gaussian smearing of gun position and direction (which is available in Mokka) is a very useful feature for the simulation of test beams, and would be welcome feature in ddsim.

(I know that we can achieve this in a 2-step process as suggested by Frank, but direct functionality would make our lives easier!)

Thanks,
Daniel.

Subject: Re: DDG4 particle gun
Posted by [sailer](#) on Wed, 16 Mar 2016 08:21:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel,

For about a week now the geant4 particle gun and general particle source can be used inside dd4hep/ddsim.

It needs the head version of lcgeo and dd4hep (or the HEAD from a week ago...)

And then one can run with

```
ddsim --enableG4GPS --macroFile gps.mac --compactFile  
path/To/CompactFile.xml -O events.slcio
```

There is an example gps.mac file in lcgeo/example
which produces a square source.

There is also a gun.mac in lcgeo/example.

But the geant4 macros from mokka should work more or less directly.

This should give you all the freedom you seek, right?

Cheers,
Andre

Subject: Re: DDG4 particle gun
Posted by [jeans](#) on Wed, 16 Mar 2016 08:45:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Andre,

After a quick look, I think that this will give us (more than enough) flexibility to define beam profiles.

Thanks for anticipating my request!
Daniel.

Subject: Re: iLCSoft release v01-17-09: centOS7 installation report

Hello ilcsofters,

I've installed v01-17-09 on CentOS7:
"CentOS Linux release 7.2.1511 (Core)
Derived from Red Hat Enterprise Linux 7.2 (Source)"
gcc version 4.8.5 20150623 (Red Hat 4.8.5-4) (GCC)

I ran into a few small problems which prevented it from being a totally pain-free experience. I post them here in case they are useful for others.

1) I gave up trying to compile and install CERNLIB, it did not seem to want to compile with gfortran...

This required a couple of changes to release-ilcsoft.cfg:

```
ignore fortran parts of MarlinReco
ilcsoft.install( MarlinReco( MarlinReco_version ))
# the following means we don't need cernlb
ilcsoft.module("MarlinReco").envcmake["MARLINRECO_FORTRAN"]='OFF'
```

and turn off FastJetClustering.

2) my dual core laptop froze completely during installation: change to
`ilcsoft.env["MAKEOPTS"]="-j1"`
in release-ilcsoft.cfg to compile serially.
The stages of "Building CXX object
DDCore/CMakeFiles/DDCore.dir/src/parsers/ParsersStandard*" in particular seem to require a lot of memory.

3) Then I bumped into a problem when compiling DD4hep.
I already had ftgl installed on my system, which meant that the root installation did not compile its own internal version of this package. The root version of the library is called libFTGL.so, while the "native" one is libftgl.so. DD4hep looks for -IFTGL, and therefore didn't find the native library. I fixed this by removing ftgl from the system and reinstalling/compiling root, forcing it to install its own version.

4) added following to release-ilcsoft.cfg for c++11:
`ilcsoft.module("MarlinPandora").envcmake["CMAKE_CXX_FLAGS"]='-std=c++11 -Wall'`
`ilcsoft.module("PandoraPFANew").envcmake["CMAKE_CXX_FLAGS"]='-std=c++11 -Wall'`

I'm not sure both were needed, but it did the trick.
(I had set
use_cpp11 = False
in release-versions.py)

5) KalTest tried to build against "native" root (i.e. the one I had installed via "yum install root*") installation, rather than version installed during ilcsoft base installation. Removing the "native" version fixed it.

Now all seems OK.
Regards,
Daniel.

Subject: running CED on output of lcgeo/CaloTB example
Posted by [jeans](#) on Wed, 23 Mar 2016 02:15:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I am having problems running CED on the output of the CaloTB example in
ilcsoft/v01-17-09/lcgeo/v00-07/CaloTB/.

I first simulate a few events in this model (just running the supplied run_sim.sh script).

Displaying the geometry with geoDisplay works OK, and the simulation itself looks OK (at least as far as I can tell from anajob).

I then try to visualise the resulting events using a steering file based on that written by ced2go (just changing the collection names to display).

The CED Marlin job exits before displaying anything, with a message:

```
*****
```

```
A runtime error occurred - (uncaught exception):
```

```
  DD4hep: extension: Object has no extension of type:DD4hep::DDRec::SurfaceManager.
```

```
Marlin will have to be terminated, sorry.
```

```
*****
```

I attach the full output.

Thanks for any help you can give!
Regards,
Daniel.

File Attachments

1) [out.log](#), downloaded 162 times

Subject: Re: running CED on output of lcgeo/CaloTB example
Posted by [gaede](#) on Wed, 23 Mar 2016 14:12:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel,

this is fixed in the trunk of MarlinUtil.

As a workaround for now, you can add the following

```
<plugins>  
<plugin name="InstallSurfaceManager"/>  
</plugins>
```

to your compact file CaloTB/compact/MainTestBeamSetup.xml.

Cheers, Frank.

PS: this will allow you to view the hits - but no detector geometry will be drawn (this would need extra code in MarlinCED).

Subject: iLCSoft release v01-17-10
Posted by [gaede](#) on Fri, 05 Aug 2016 16:44:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new i|LCSoft release v01-17-10 is available.

Please use the corresponding ilcinstall tag for installation:

```
svn co https://svnsrv.desy.de/public/ilctools/ilcinstall/tags/v01-17-10
```

Reference installations are available at

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-10 # SL6 gcc 4.4  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-17-10 # SL6 gcc 4.8 c++11  
/cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-10 # SL6 gcc 4.4  
/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-17-10 # SL6 gcc 4.8 c++11
```

There is also an ILDConfig release v01-17-10 that has steering files for running the simulation and reconstruction (both old and new) for ILD with this release:

```
svn co https://svnsrv.desy.de/public/marlinreco/ILDConfig/tags/v01-17-10
```

For instructions see:

```
https:// svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags  
/v01-17-10/StandardConfig/current/README
```

and

```
https:// svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags  
/v01-17-10/StandardConfig/lcgeo_current/README.md
```

Please report problems and questions related to this release in this thread.

Frank, for the ilcsoft team.

#

```
# iLCSoft release - v01-17-10
#
#####
```

This is a new developers release with some updates and fixes to v01-17-09 and should be the last release with c++98, ROOT5, Mokka, ... (up to patches)

=====

Changes in v01-17-10 w.r.t. iLCSoft v01-17-09

=====

- consistently handle CMAKE_CXX_FLAGS and building for c++11 in ilcutil and updated all packages accordingly
- removed local CMAKE_CXX_FLAGS

ilcutil v01-03

A.Sailer

- Add Macro to set the default cxx flags for all packages depending on ilcutil
- Add option for USE_CXX11 and set it to True by default

=====

lcio v02-07-02

=====

F.Gaede: - add EXAMPLE/readmcparticles.cc

- fix some documentation
 - Cluster: itheta/iPhi and deprecated methods in MCParticle
- bug fix in SIOReader::skipNEvents:
 - allow to call with 0 as argument
- fix handling of default CXX_FLAGS
 - can be overwritten on command line
- install also the python source code and examples (M.Petric)

J.McCormick: Implement handling of daughter to mother relationships in StdHep reader;
use separate classes for now: LCStdHepRdrNew (and EXAMPLE/stdhepjob_new.cc)
- pending more testing.

DDKalTest v01-00

F.Gaede

- added DDSurfaces::ISurface* surface() and some debug output
- remove hard coded muon mass for track fits
- fixed the sorting policy for zdisks
-> major improvement in forward track fitting
- crosscheck of energy loss from aidaTT
 - commented out
- made compatible with c++11 and ROOT6

MarlinTrk v02-02

F. Gaede

- added setMass()/getMass to IMarlinTrack interface
- implemented for KalTest, DDKalTest and aidaTT tracks
- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- fixed narrowing in initializer lists

FCalClusterer v0-03

BeamCalReco (A.Sailer):

- * Reduce memory footprint and fix memory leaks for Parameterized background method
- * Silence warnigns from root integral and getrandom, enabled for DEBUG0

* Compatible with ROOT 6

LumiCalReco (B.Pawlik):

* Massive fixes, mainly coordinate system, cluster merging. Creating ClustersClass Map and root output is optional now

* Fix LCIO interface to pass hit positions, geo info. Create LucasGear for Marlin reco, and more...

=====
Physsim v00-03
=====

F.Gaede

- made compatible w/ c++11
- remove -Wall -ansi

=====
lcgeo v00-08
=====

S. Lu

- Adapted to Hcal endcap Ring to DD4hep envelope, and improve the code to read the envelope information clearly from compact file directly.

- Updated to read the 'HcalEndcapRing_inner_radius' directly from compact file, and derivative dependence on 'Ecal_endcap_outer_radius' implemented in compact file.

- added 'Hcal_endcap_thickness' the value came from $26.5\text{mm} \cdot 48 + 15.0\text{mm}$. Fixed the 'Hcal_barrel_thickness' the value came from engineer, it is about $26.5\text{mm} \cdot 48$ and 0.3mm of tolerance for the 'Hcal_outer_radius' maximum 3395.5mm

- Added two parameters 'Ecal_Barrel_thickness', and 'Hcal_Barrel_thickness', and use them to increase the ILD calorimeters (ECAL, HCAL) radius automatically.

- Update material for the Ecal ECRing module as Mokka used

- Change the thickness of the scintillator in the active layer.

- Updated 'detector/CaloTB/CaloPrototype_v02.cpp' and remove the hardcode layer/ identifier 'layer/K' in the testbeam geometry driver, get this string via segmentation, and use the definition in the compact xml file, whatever the user wanted, 'layer' or, 'K'

- added ID 'slice' in the geometry driver, and to be used in the cellEncoding string in compact file.

- Updated SEcal04_Endcaps.cpp to set the Magic Wafer size in group tower, and set the group identifier in the compact file for both Barrel (in layer group) and Endcaps (in tower group).

- Use WaferGridXY segmentation for the EcalEndcapsCollection digitization in the DD4hep

ILD_o1_v05, to improve the Magic wafer part in each layer too, as EcalBarrelCollection

D. Protopopescu

- AHCAL Barrel implementation of the layout from 70th SiD Optimization Meeting presentation, from Ross Gordon McCoy (ross.mccoy@mavs.uta.edu), HCalEndcap with the same layer structure, and SiD_o1_v03 including the two new XMLs.

Perhaps the Scintillator HCal should be o2 (option 2)?

- Added new materials, and terminator absorber layer to the Scintillator HCal
- Fix for SiD 'make test'
- Updated Muon endcap and barrel XMLs with latest dimensions, and using the Generic drivers (will have to change this!). Added envelope (Aidan).

Modified Fe slice thicknesses (20->19.6cm) to fit within the new dimensions.

- Added test for SiD_o1_v1

M. Petric

- Make the sum of readout bits 32 otherwise we have a problem with the encoding of the surfaces

- Fix problem with replication of color name
- Add interlink to tracker
- Change CI to new Geant4 10.2.2
- converged design for tracker
- Add diagnostic color to gcc 4.9 and move CI build to Ninja and add output on failure to tests
- New tracker layout as requested by the tracking group
- Adopt to 40 layers ECal in the endscaps
- Add change ECal to 40 layers and move everything in barrel after ECal 27mm outwards
- Converted readme to markdown and added badge
- change dd4hep init
- Make lcgeo ROOT6 compatible. There is no Reflex in 6.

N. Nikiforou

- Modified CLIC_o2_v04 BeamCal to avoid dummy layer

F. Gaede

- updated envelope parameters of ILD_o1_v05 simulation model
- updated tex file
- added picture for barrel envelopes

K. Coterra

- Sc_Si_hybrid Ecal drivers for barrel and endcaps were created. So far, they dedicated for Sc-strip ECAL.

- parameter files for ILD_o3_v05 model, Sc_Si_hybridECAL with AHCAL, were created. So far, for only Sc strip ECal, not for Hybrid ECAL.

Y. Voutsinas

- updates in VXD material/surface description
- external cabling and internal strip lines for the innermost VXD layer added
- adding surfaces for beryllium annulus blocks
- adding surfaces for the electronics at the end of the ladders
- adding surface for the beryllium shell cone

A.Sailer 05/07/2016

- DDSim: * If no random seed is defined we get a random random seed.
* new option random.enableEventSeed to calculate reproducible random seeds using the same method as the EventSeeder in Marlin

A.Sailer 08/03/2016

- DDSim: Add enableG4Gun and enableG4GPS flags to enable the Geant4 Gun or GeneralParticleSource
see examples/gun.mac or examples/gps.mac
use with ddsim --enablgeG4Gun --macroFile gun.mac --compactFile ...

=====
DD4hep v00-16
=====

Shaojun Lu | 2016-07-26

Added new MegatileLayerGridXY segmentation for Scintillator strip Ecal, which is implemented by K.Kotera,
and used by SEcal04Hybrid geometry driver.

- many (minor fixes) by F. Gaede, M. Petric, A Sailer, ...

2016-06-24 M.Frank

Implement multiple segmentations.

Though one readout objects (associated to one DetElement) may only have on segmentation, The MultiSegmentation type allows to have several sub-segmentations, which can be chosen from.

Please see examples/ClientTests/*/MultiSegmentations

At the same time allow the readout object to defined multiple collections through the IDDescriptor.

Please see examples/ClientTests/*/MultiCollections

The combined example can be found in

Please see examples/ClientTests/*/NestedSegmentations

2016-05-03 M.Frank

Green light is ON. You may use revision 2237 and above.

2016-05-03 M.Frank

+++++ NOTE:

+++++

- Mega commit. Do not use until green signal arrives with seperate release notice.

+++++

+++++

- Package DDDDB: Prove of concept, that we can populate the geometry also from other sources than xml files obeying the lccdd notation.
- DDCore/XML: Implement the possibility to use external entity resolution for Xerces. There is not equivalent in TinyXML (AFAIK).
- Remove some of the Spirit parsers, since the libraries are getting way too big. This implicitly assumes, that object properties of 'exotic' types such as deque<long double> etc. are not really used. If you need properties stay to the obvious: string, int float, double and vectors, maps and sets thereof!
- DDG4 needed to public the DDG4Python include directory (Now in DDG4/Python). Reason: try to create ROOT 6 dictionaries, which can be relocated for release purposes.
- ROOT 6 dictionary generation: Create dictionaries with the additional flag: "-inlineInputHeader" which physically includes the header files used to generate the dictionaries --> relocatability.

2016-03-21 M.Frank

- Add additional vertex argument to EventReader::readParticles, which supposedly should

2016-03-03 M.Frank

- Improve/complete job options type palette.
- Allow modify the Geant4IsotropGenerator to shoot particle distributions according to several distributions:
flat in : uniform, cos(theta), eta/pseudorapidity, fbar (1+cos^2 theta)
Option: generator.Distribution = 'cos(theta)'

Default is uniform.

```
=====
LCTuple v01-06
=====
```

F.Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long

S.Lu

- Added root headers and declared the parameters to fix the error: use of undeclared identifier 'MyLCTuple' in ROOT 6.06/02.

```
=====
MarlinTPC v01-02
=====
```

```
-----
r5055 | kleinwrt | 2016-06-17 13:38:40 +0200 (Fri, 17 Jun 2016) | 1 line
```

Changed paths:

- M /trunk/reconstruction/trackreco/include/GeneralBrokenLineInterfaceHelpers.h
- M /trunk/reconstruction/trackreco/src/GeneralBrokenLineInterfaceHelpers.cc
- M /trunk/reconstruction/trackreco/src/SimpleHelixGeneralBrokenLineInterfaceProcessor.cc
- M /trunk/reconstruction/trackreco/src/StepWiseHelixGeneralBrokenLineInterfaceProcessor.cc

GeneralBrokenLineInterfaceHelpers updated for TimePix

```
-----
r5054 | kleinwrt | 2016-06-17 10:10:01 +0200 (Fri, 17 Jun 2016) | 1 line
```

Changed paths:

- M /trunk/reconstruction/trackreco/include/SimpleHelixGeneralBrokenLineInterfaceProcessor.h
- M /trunk/reconstruction/trackreco/include/SimpleHelixTrackModel.h
- M /trunk/reconstruction/trackreco/include/StepWiseHelixGeneralBrokenLineInterfaceProcessor.h
- M /trunk/reconstruction/trackreco/src/SimpleHelixTrackModel.cc

Improved numerical stability for very small curvature

```
-----
r5038 | krieger@physik.uni-bonn.de | 2016-04-25 12:47:32 +0200 (Mon, 25 Apr 2016) | 1 line
```

Changed paths:

- M /trunk/tools/processors/src/TimePixXYReaderProcessor.cc

Merged small bugfixes for TimePixXYReaderProcessor from branches/krieger

r5036 | krieger@physik.uni-bonn.de | 2016-04-20 18:30:28 +0200 (Wed, 20 Apr 2016) | 5 lines

Changed paths:

M /trunk/tools/processors/include/TimePixXYReaderProcessor.h
M /trunk/tools/processors/src/TimePixXYReaderProcessor.cc

Merged changes done to the TimePixXYReaderProcessor from branches/krieger:
Processor is now also able to correctly read in frames with the new header format used by
SRS-TOS

Compatibility to old header versions and frames without header is assured

r5034 | krieger@physik.uni-bonn.de | 2016-04-06 09:53:46 +0200 (Wed, 06 Apr 2016) | 15 lines

Changed paths:

A /trunk/analysis/include/WriteXrayObjectsToROOTTreeProcessor.h (from
/branches/krieger/analysis/include/WriteXrayObjectsToROOTTreeProcessor.h:5033)

A /trunk/analysis/include/XrayLikelihoodProcessor.h (from
/branches/krieger/analysis/include/XrayLikelihoodProcessor.h:5032)

A /trunk/analysis/src/WriteXrayObjectsToROOTTreeProcessor.cc (from
/branches/krieger/analysis/src/WriteXrayObjectsToROOTTreeProcessor.cc:5033)

A /trunk/analysis/src/XrayLikelihoodProcessor.cc (from
/branches/krieger/analysis/src/XrayLikelihoodProcessor.cc:5032)

Added two additional processors used for X-ray photons recorded with an InGrid based detector:

XrayLikelihoodProcessor

-Calculates likelihood value for each GridPixXrayObject based on reference histograms
loaded from root file(s)

-Different reference sets for defined energy ranges can be given

-Reference histograms are created with XrayGeometryAnalysisProcessor (may need data
selection with XrayCutApplicationProcessor before)

WriteXrayObjectsToROOTTreeProcessor

-Exports all GridPixXrayObjects given as collection to a root tree saved as .root file

-Several event shape variables are exported

-Energy and likelihood are exported

-Pixel positions and values are exported as three vectors per event

-Event number, run number and timestamp are exported

r5032 | gaede | 2016-03-08 15:14:34 +0100 (Tue, 08 Mar 2016) | 3 lines

Changed paths:

M /trunk/reconstruction/trackreco/src/TrackFitterLikelihoodProcessor.cc

- added namespace std to isinf and isnan (SL6/gcc4.8)

r5031 | gaede | 2016-03-07 16:41:16 +0100 (Mon, 07 Mar 2016) | 16 lines

Changed paths:

M /trunk/CMakeLists.txt

M /trunk/calibration/src/MMScatterProcessor.cc

M /trunk/reconstruction/padbased/src/MMHitFinderProcessor.cc

M /trunk/reconstruction/photoelectric/src/PhotoelectricClusterTools.cc

M /trunk/tools/DataConversionUtils/src/TEventBlock_V42.cpp

M /trunk/validation/testSimHitDistributor/testSimHitDistributor.cpp

- made compatible with c++11 and ROOT6
 - removed -ansi -pedantic -Wno-long-long ...
 - added std:: to i/ofstream
 - fixed narrowing in initializer list

- ***** NOTE: *****

If build with ROOT 6 the following processor is excluded:

./reconstruction/pixelbased/src/GridPixXrayObjectCalculatorProcessor.cc

as it uses TFitterMinut which no longer exists in ROOT6

-> needs to be fixed by an expert ...

r5030 | kleinwrt | 2016-02-24 15:22:54 +0100 (Wed, 24 Feb 2016) | 1 line

Changed paths:

M /trunk/reconstruction/trackreco/include/SimpleHelixGeneralBrokenLineInterfaceProcessor.h

M

/trunk/reconstruction/trackreco/include/StepWiseHelixGeneralBrokenLineInterfaceProcessor.h

M /trunk/reconstruction/trackreco/src/SimpleHelixGeneralBrokenLineInterfaceProcessor.cc

M /trunk/reconstruction/trackreco/src/SimpleHelixTrackModel.cc

M /trunk/reconstruction/trackreco/src/StepWiseHelixGeneralBrokenLineInterfaceProcessor.cc

TPC gas (between first and last hit) implemented as homogeneous scatterer for track fitting with GBL

=====
Overlay v00-15
=====

F. Gaede
- protect against calling LCReader::skipNEvents() with 0 (now also fixed in LCIO trunk)

R. Simoniello
- new collections added for CLIC

A. Sailer
- Remove Ansi and other definitions, allow compiling with different standard

=====
CEDViewer v01-12
=====

F. Gaede
- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- fixed narrowing in initializer lists

=====
MarlinKinfitter v00-03
=====

J. List
- added FourJetZHPairing

F. Gaede
- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long

G. Wilson

- Revise LeptonFitObject to use (q/pt, theta, phi) and track error matrix.
- Also add some additional print methods to BaseFitObject.
- Fix benign index issue with jlocal<ilocal elements incorrectly set for mirror off-diagonal elements in getSecondDerivative (affected all fit objects).
- Add constructor based on TrackState requested by Daniel

=====
 ILDPerformance v01-00
 =====

D. Jeans

- add simulation validation processor, examples and short documentation
- extend for CalorimeterHits and TrackerHits; a few other cleanups

M. Petric

- Added GSL because required. It would however better to fix aidaTT to propagate GSL properly

H.L. Tran

- Modify CMakeLists.txt file to include the validateSim folder.

J. List

- added ZHWW5CFit
- added ZH example
- added first version of test on WW->4jets
- added likelihood values for all hypotheses as LiPDG_mu etc to PIDtree; == 9 means is -inf on DST =>check with Masakazu?; == 2 means no PID or no recoParticle existing for this MCP
- significantly improved truth vs reco navigation
- added dEdx distance and dEdx likelihood to tree)

Y. Voutsinas

- add a diagnostics script that run a simple analysis hZ->mumuX for Mokka as well
- processor that calculates track - hit residuals based on aidaTT intersection calculation added
- updating tracking performance macros, track resolution performance plots

F. Gaede

- - finalized plotting of dE/dx for all particle types (code from P.McKeown)
- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- added trueMass to PID TTree
- added macro plotdEdx
- bug fix for getParents.size()==0

P. McKeown

- added plots for
- efficiency and fake rate for likelihoodPID
- dE/dx
- likelihood PID
- efficiency vs. purity for likelihoodPID

=====
MarlinReco v01-15
=====

M. Kurata

- Include dEdx angular correction, which is a feedback from Sviatoslav
- add PDF creation processor and steering file
- add hadron type likelihood and probability
- PIDTools: Adding PIDVariables and PIDParticles to trunk
- assign 999 to likelihood when cannot perform PID

F. Gaede

- introduced anonymous namespace to compile w/ -pedantic
- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- build sub-package Relastic only if DD4hep is found
- added implementation of getExtension() - copied from DDMarlinPandora

D. Jeans

- introduction of RealisticCaloDigi/Reco processors: refactorisation and cleaning up of ILDCaloDigi
- gap corrections split off into BruteForceEcalGapFiller processor; change to setup of reco hit relations
- fixed logic in determination of calibration by layer

M. Chera

- fixed bug such that the processor now skips events with empty PFO collections without crashing

=====
DDMarlinPandora v00-04
=====

N. Nikiforou

- Merged changes in MarlinPandora by Bono up to Mar. 30t
- Added note to handle exception not caught for very small omega in TrackCreator. Not a problem currently

=====
MarlinTrkProcessors v02-03
=====

R. Simoniello

- Add procedure to handle with spiral tracks. Fit direction and fit initialisation configurable by steering macro (default fit direction: backward)
- improved speed, map of hits on each element introduced, find neighbours of element ID and extrapolate to them (to be improved to some problem in the geo datat structure)

F. Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- fixed narrowing in initializer lists

Y. Voutsinas

- Minivector creation from 1D SIT hits added

=====
MarlinDD4hep v00-02
=====

F. Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long

=====
Marlin v01-09
=====

F. Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- fix comparison of ostream and std::cout

A. Sailer

- Add program to compare two marlinSteering files and print out differences between the two files based on parameter values installs the compareMarlinSteeringFiles.py into the Marlin bin folder, which is usually in the PATH

R. Simoniello

- removed -Wall etc from mymarlin example for c++11

=====
MarlinUtil v01-12
=====

F. Gaede

- protect against missing SurfaceManager object in DD4hep model
- updated to use gsl 2.1
- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long

=====
RAIDA v01-07
=====

F. Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long

=====
ForwardTracking v01-10
=====

F.Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long

- fixed narrowing in initializer lists

=====
KiTrackMarlin v01-09
=====

F. Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- fixed narrowing in initializer lists
- added std to i/ofstream

Y. Voutsinas

- VXD cellular automaton can cope with mini-vectors made of 1 D hits

=====
KiTrack v01-07
=====

F. Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long

=====
Clupatra v00-14
=====

F. Gaede

- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- fixed template parameters in nnclu::(not)inRange<int,int>

=====
MarlinTrk v02-02
=====

F. Gaede

- added setMass()/getMass to IMarlinTrack interface
- implemented for KalTest, DDKalTest and aidaTT tracks
- made compatible with c++11
- removed -ansi -pedantic -Wno-long-long
- fixed narrowing in initializer lists

Subject: compilation with TMinuit
Posted by [jeans](#) on Thu, 22 Sep 2016 14:00:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I'm having difficulties compiling a processor which uses TMinuit.

Since ilcsoft v01-17-10 (setup using the script at
/cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-10/init_ilcsoft.sh), when I try to compile a
processor containing e.g.

```
"#include "TMinuit.h"  
.  
.  
.  
TMinuit dd(2);  
.  
.  
."
```

I get a compiler error

```
CMakeFiles/simplest.dir/src/simple.cc.o: In function  
`simplestProcessor::processEvent(EVENT::LCEvent*)':  
/home/ilc/jeans/myProcessors/simplestProcessor/src/simple.cc :38: undefined reference to  
`TMinuit::TMinuit(int)'
```

I don't see a similar error when building against previous central installations of ilcsoft (at least I've tested v01-17-09 and v01-17-08).

My CMakeLists.txt contains

```
SET( ROOT_USE_COMPONENTS Minit )  
FIND_PACKAGE( ROOT REQUIRED )  
INCLUDE_DIRECTORIES( ${ROOT_INCLUDE_DIRS} )  
LINK_LIBRARIES( ${ROOT_LIBRARIES} )  
ADD_DEFINITIONS( ${ROOT_DEFINITIONS} )
```

Thanks for any insights/suggestions,
Daniel.

Subject: Re: compilation with TMinuit
Posted by [gaede](#) on Thu, 22 Sep 2016 14:18:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel,

can you try the following:

```
FIND_PACKAGE( ROOT REQUIRED COMPONENTS Minit )  
INCLUDE_DIRECTORIES( ${ROOT_INCLUDE_DIRS} )  
LINK_LIBRARIES( ${ROOT_LIBRARIES} ${ROOT_COMPONENT_LIBRARIES} )
```

Cheers, Frank.

Subject: Re: compilation with TMinuit
Posted by [jeans](#) on Thu, 22 Sep 2016 14:25:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks, Frank.

adding `#{ROOT_COMPONENT_LIBRARIES}` fixed it.

Daniel.

Subject: pyLCIO in v01-17-10

Posted by [miyamoto](#) on Fri, 30 Sep 2016 05:28:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

I'm using `ilcsoft-v01-17-10` in `/cvmfs/ilc.desy.de/sw`.

But, the Python binding of LCIO doesn't work, because

(1) `liblcioDict.so` does not exist in

`/cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-10/lcio/v02-07 -02/lib`

(2) Those in `/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-17-10`

produces following errors, when importing `pyLCIO`

```
Python 2.7.4 (default, May 5 2013, 17:33:10)
```

```
[GCC 4.8.0] on linux2
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import pyLCIO
```

```
Loading LCIO ROOT dictionaries ...
```

```
Error in <TCling::RegisterModule>: cannot find dictionary module rootDict_rdict.pcm
```

```
Error in <TCling::RegisterModule>: cannot find dictionary module IOIMPL_rdict.pcm
```

```
Error in <TCling::RegisterModule>: cannot find dictionary module IMPL_rdict.pcm
```

```
Error in <TCling::RegisterModule>: cannot find dictionary module EVENT_rdict.pcm
```

```
Error in <TCling::RegisterModule>: cannot find dictionary module IO_rdict.pcm
```

```
Error in <TCling::RegisterModule>: cannot find dictionary module UTIL_rdict.pcm
```

Additionally, `init_ilcsoft.sh` in `gcc4.8/v01-17-10` uses `/afs` for python and gcc. Since access to `afs` directory from remote site is very slow, would you modify to use those in `/cvmfs`

best regards

Akiya Miyamoto

KEK

Subject: Re: pyLCIO in v01-17-10
Posted by [gaede](#) on Fri, 30 Sep 2016 08:16:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Akiya,

thanks for the bug report.

(1) we cannot build a root dictionary for lcio with ROOT 5 and gcc44 any more - so this is a 'feature'

(2) there actually is a problem in the cmake installation procedure for lcio that sometimes causes the *.pcm file to not be installed properly - it is currently under investigation.

In any case it is fixed for the reference installations in:

```
/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-17-10/  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-17-10/
```

by copying the *.pcm files manually:

```
cp $LCIO/build/rootdict/*.pcm $LCIO/lib
```

in case other people have the same problem for their installations.

The init script in cvmfs now also uses gcc and python from cvmfs.

Cheers, Frank.

Subject: iLCSoft release v01-17-11
Posted by [gaede](#) on Tue, 15 Nov 2016 16:42:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft (patch) release v01-17-11 is available.

NB: This will be the last release compatible with gcc44 and ROOT5 and Mokka.

Please use the corresponding ilcinstall tag for installation:

```
svn co https://svnsrv.desy.de/public/ilctools/ilcinstall/tags/v01-17-11
```

Reference installations are available at

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/v01-17-11 # SL6 gcc 4.4  
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-17-11 # SL6 gcc 4.8 c++11  
/cvmfs/ilc.desy.de/sw/x86_64_gcc44_sl6/v01-17-11 # SL6 gcc 4.4  
/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-17-11 # SL6 gcc 4.8 c++11
```

There is also an ILDConfig release v01-17-11 that has steering files for running the simulation and reconstruction (both old and new) for ILD with this release:

```
svn co https://svnsrv.desy.de/public/marlinreco/ILDConfig/tags/v01-17-11
```

For instructions see:

```
https:// svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags  
/v01-17-11/StandardConfig/current/README
```

and

```
https:// svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags  
/v01-17-11/StandardConfig/lcgeo_current/README.md
```

Please report problems and questions related to this release in this thread.

Frank, for the ilcsoft team.

```
#####  
#  
# iLCSoft release - v01-17-11  
#  
#####
```

This is a patch release to v01-17-10 that fixes a few known issues and provides some updates and should be the last release with c++98, ROOT5, Mokka, ...

```
=====
```

```
# Changes in v01-17-11 w.r.t. iLCSoft v01-17-10
```

```
=====
```

```
## LCFIPlus: v00-06-05
```

- deal with TMVA API changes in ROOT 6.08
- fixes known issue with TMVA training in ROOT versions between 5.34 and 6.06

```
## CED: v01-09-02
```

- wrong version was included in v01-17-10

```
## MarlinTrkProcessors: v02-04
```

- improvement in MiniVectorCA based patrec
- added particle mass to RefitProcessor
- cleaned up ExtrToTracker and DDCellsAutomatonMV

```
## Overlay: v00-16
```

- add parameter 'KeepPairsMCPinfo' tp OverlayBX

```
## MarlinKinfit: v00-04
```

- fixed error calculation in presence of unmeasured parameters in OPALFitter, New(ton)Fitter was ok already!
- the OPALFitter does `_not_` work with SoftConstraints (runs, but SCs don't "do" anything)

PandoraPFA: v02-09-01

- new patch release with small bug fix for software compensation

LCIO: v02-07-03

- fix issue w/ installing pcm files

Gear: v01-06-01

- bug fix for computing number of TPC pad rows
- observed in small ILD model

MarlinReco: v01-16

- add package ErrorFlow (compute jet-specific energy uncertainty)

ILDPerformance: v01-01

- kinfit:
 - new reference histogram file for ZH5CFit
 - reference histograms for pull distributions
- tracking:
 - adding macros that compare the resolutions with one from DBD plus a batch script to run the diagnostics to the example analysos reco files

Subject: iLCSoft release v01-19
Posted by [gaede](#) on Fri, 25 Nov 2016 17:22:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft developers release v01-19 is available.

This is the first release that exclusively works with DD4hep based simulation and tools, as well as c++11.

Please use the corresponding ilcinstall tag for installation:

```
svn co https://svnsrv.desy.de/public/ilctools/ilcinstall/tags/v01-19
```

Reference installations are available at

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-19 # SL6 gcc 4.8  
/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-19 # SL6 gcc 4.8
```

There is also an ILDConfig release v01-19 that has steering files for running the new simulation and reconstruction for ILD with this release:

```
svn co https://svnsrv.desy.de/public/marlinreco/ILDConfig/tags/v01-19
```

For instructions see:

```
https:// svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags  
/v01-19/StandardConfig/lcgeo_current/README.md
```

Please report problems and questions related to this release in this thread.

Frank, for the ilcsoft team.

```
#####  
#  
# iLCSoft release - v01-19  
#  
#####
```

This is the first developers release that exclusively works with the new DD4hep based software packages only. It requires a compiler compatible with c++11 (e.g. gcc48 or higher) such that c++11 features can be used from now on.

```
=====
```

```
# Changes in v01-19 w.r.t. iLCSoft v01-17-11
```

```
=====
```

```
## Mokka  
- dropped
```

```
## MarlinPandora  
- dropped
```

```
## Marlin v01-10
```

- F.Gaede
 - add a licence to marlin: GPLv3
- S.Lu
 - fix buidling the MarlinGUI for gcc48
- D.Hynds
 - added escape behaviour mapped to SIGQUIT signal (^).
 - Hitting this during running will run the end routine for each processor and then exit Marlin, allowing you to stop event processing and still preserve histograms/results/etc.

```
## MarlinDD4hep v00-03
```

- A.Sailer
 - add parameter "EncodingStringParameterName" to overwrite ILDCellID0::encoder_string as workaround for CLIC/SiD tracking

DDMarlinPandora v00-05

- A.Sailer

* DDMarlinPandora::CaloHitCreator:

Get normal vector from DD4hep::DetElement instead of using assumptions about calorimeter construction to calculate phi, which are no longer true

Add options to set local NormalVector for ecal, hcal, and muon barrels in case they are not build with layers increasing along the z-axis

ECalBarrelNormalVector

HCalBarrelNormalVector

YokeBarrelNormalVector

All three are three vectors

Correct and simplify calculation of NormalVector for cylindrical barrel calorimeters

* Fix many compiler warnings

PandoraPFANew v03-01-00

many updates in subpackages since previous version v02-09-02 :

PandoraSDK: v03-00-00 - prev: v02-03-02

PandoraMonitoring: v03-01-00 - prev: v02-03-01

LArContent: v03_01_00 - prev: v02_07_09

LCContent: v03-00-00 - prev: v02-04-01

ExampleContent: v03-00-00 - prev: v02-01-03

for details see:

<https://raw.githubusercontent.com/PandoraPFA/PandoraSDK/master/ChangeLog.txt>

<https://raw.githubusercontent.com/PandoraPFA/PandoraMonitoring/master/ChangeLog.txt>

<https://raw.githubusercontent.com/PandoraPFA/LArContent/master/ChangeLog.txt>

<https://raw.githubusercontent.com/PandoraPFA/LCContent/master/ChangeLog.txt>

<https://raw.githubusercontent.com/PandoraPFA/ExampleContent/master/ChangeLog.txt>

DD4hep v00-19

2016-11-17 D.Jeans

- rewritten MegatileLayerGrid class
- not backward compatible

M.Petric:

- improved Travis CI configuration
- remove some tests from CI

v00-18

2016-11-09 F.Gaede

- updated DDRec::LayeredCalorimeterData::Layer:
 - remove deprecated thickness
 - add phi0
- add copy assignment to DDRec::MaterialData

2016-11-08 M.Frank

- Improve conditions handling. Started to implement using simple telescope detector from C.Burr.
- Add variable DD4HEP_LIBRARY_PATH to be used on MacOSX El Capitan to overcome DYLD_LIBRARY_PATH protection issues.
- Fix to copy with negative(rebased) IDDDescriptor fields (affects IDDDescriptor and VolumeManager).

2 New tests:

- ClientTests/Bitfield_SidesTest.xml to test readout string:
system:8,barrel:-2
- ClientTests/Bitfield_SidesTest2.xml to test readout string:
system:16,barrel:16:-5

2016-10-18 M.Frank

Due to pressure of the FCC folks, I tried to implement a more DD4hep like implementation of the the segmentation objects. For testing only CartesianGridXY. If this mechanism works, it could be a starting recipe for the rest of the segmentations. The draw-back of this approach is, that assignments are not reversible:

```
DD4hep::CartesianGridXY xy = readout.segmentation(); // Works
DD4hep::Segmentation seg = xy; // Should not work
```

Reason: the managed objects are different....at some point in time I will have to find a clean solution for this, but the required changes for such a solution shall be manageable.

lcgeo v00-09

- developers release with many changes and updates to CLIC, ILD and SiD simulation models, e.g.:
- made compatible w/ DD4hep v00-18 (removed LayeredCalorimeterData::Layer::thickness)
- ILD:
 - add models ILD_l1_v01 and ILD_s1_v01
 - new LCal driver
 - new Hcal barrel w/ staircase layout
 - ...
- CLIC:
 - new models CLIC_o3_v06 and CLIC_o3_v07
 - ...
- SiD:
 - new model: SiD_o2_v01
 - make compatible w/ reconstruction
 - ...

aidaTT v00-05

F. Gaede

- replaced GSL with Eigen for Vector5 and fiveByfiveMatrix
- needs Eigen Headers to be installed
- removed simpleFitter for now (depends on GSL and is used nowhere)

DDKalTest v01-00-01

A.Sailer

- DDParallelPlanarMeasLayer: protect against basically infinite loop if phi value is very very large
- DDKaltest: ignore warnings from external header files

lcio v02-07-04

F.Gaede

- make LCIO relocatable (as suggested by M.Petric/A.Sailer)
 - install cmake macros
 - use CMAKE_CURRENT_LIST_DIR in cmake config files
- remove const from ILDCellID0::encoder_string
- so it can be changed for CLIC/SiD as
as workaround

Subject: Re: iLCSoft release v01-19
Posted by [jeans](#) on Mon, 28 Nov 2016 09:04:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

There seems to be some problem with setting of Geant4 path in DD4hep when using the reference installation?

```
[jeans@cw12 ~]$ source
/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-19/DD4hep/v00-19/bin/thisdd4hep.sh
[jeans@cw12 ~]$ echo $Geant4_DIR
/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-19/geant4/10.02.p02/lib64/Geant4-10.2.2
[jeans@cw12 ~]$ ls $Geant4_DIR
ls: cannot access
/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-19/geant4/10.02.p02/lib64/Geant4-10.2.2: No such
file or directory
```

Cheers,
Daniel.

Subject: Re: iLCSoft release v01-19
Posted by [gaede](#) on Mon, 28 Nov 2016 11:17:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Subject: Re: iLCSoft release v01-19
Posted by [gaede](#) on Mon, 28 Nov 2016 11:18:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Daniel,

should be fixed now (or in an hour after things have been propagated through cvmfs.)

CHeers, Frank.

Subject: ilcsoft release v01-19-01
Posted by [gaede](#) on Thu, 19 Jan 2017 17:33:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft developers release v01-19-01 is available.

Please note that this release requires at least gcc4.9

Please use the corresponding ilcinstall tag for installation:

```
svn co https://svnsrv.desy.de/public/ilctools/ilcinstall/tags/v01-19-01
```

Reference installations are available at

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc49_sl6/v01-19-01 # SL6 gcc 4.9  
/cvmfs/ilc.desy.de/sw/x86_64_gcc49_sl6/v01-19-01 # SL6 gcc 4.9
```

There is also an ILDConfig release v01-19-01 that has steering files for running the new simulation and reconstruction for ILD with this release:

```
svn co https://svnsrv.desy.de/public/marlinreco/ILDConfig/tags/v01-19-01
```

For instructions see:

```
https://svnsrv.desy.de/websvn/wsvn/General.marlinreco/ILDConfig/tags  
/v01-19-01/StandardConfig/lcgeo_current/README.md
```

Please report problems and questions related to this release in this thread.

Frank, for the ilcsoft team.

```
#####
```

```
#
```

```
# iLCSoft release - v01-19-01
```

```
#  
#####
```

This is the first update to the new developers release series v01-19-xx that exclusively works with the new DD4hep based software packages only.

It requires a compiler compatible with c++11 (e.g. gcc49 or higher) such that c++11 features can be used from now on.

```
=====
```

```
# Changes in v01-19 w.r.t. iLCSoft v01-19-01
```

```
=====
```

```
## MarlinReco v0-17
```

S. Lukic

- Adding SiPID package to MarlinReco.
For the moment it contains the processor which calculates dE/dx for a silicon tracker and writes it to the lcio file.
Primarily intended for CLICdet but generally applicable.
DD4hep is used to access geometry.
CMakeLists.txt in MarlinReco was adjusted. The package is added only if DD4hep and MarlinTrk are found.

M.Petric

- fix in TruthVertexFinder

D. Jeans

- copy calodigi documentation into Realistic package (+ a few changes to help latex compile)

S. Lu

- comment out 'ADD_MARLINRECO_PKG(./Clustering/BCalReco)' for solving '*** Break *** segmentation violation' at the end of Marlin run.
ToDo: To ckeck which ilcsoft package use the same resource (maybe root resource).

```
## lcgeo v00-10
```

Shaojun Lu 2017-01-16

- Added tube_defs.xml for ILD_s1_v01.

Frank Gaede 2017-01-16

- activate TPC_endplate_mix that was used before

Daniel Jeans 2017-01-16

- tidy up for new beampipe changes
- fix lumical overlaps (internal + with beamtube)
- fix overlaps between rewritten beampipe and ftd
- fix vtx/beampipe overlaps
- update beam pipe description

Frank Gaede 2017-01-12

- fix TPC cathode volumes (include in cathodeLog)

Shaojun Lu 2017-01-12

- Included 'detector_types.xml' and added missed parameters 'tracker_region_rmax' and 'tracker_region_zmax' for reconstruction.

Shaojun Lu 2017-01-11

- define Yoke relative to Hcal and Coil.

Frank Gaede 2017-01-05

- update version and release notes for v00-10

Daniel Jeans 2016-12-26

- define yoke barrel size with respect to outer radius of coil
- tidy up ILD_?1_v01 compact description. Move all common definitions to ILD_common_v01. Keep only top-level steering file and mode-specific definitions in ILD_I1 and ILD_s1 directories. move unused files to extra_stuff directory.

simoniel 2016-12-15

- map cellid of surf <--> vector of cellid of neighbouring surf on the same layer (new NeighbourSurfacesStruct) filled in drivers currently used by CLIC

simoniel 2016-12-09

- fill of neighboring surfaces map done also for CLIC_o2_v04_p1 drivers and Vertex drivers in use
- Dependence from LCIO re-introduced. Compute neighbours of every surface and fill map of cellID of surface <--> vector of cellID of neighbouring surfaces for zPlanar and zPetalDisk data struct.

Shaojun Lu 2016-12-16

- Updated Yoke05 parameters in compact XML files.

- Replaced old parameter 'Hcal_endcap_zmin' with the updated new naming convention parameter 'HcalEndcap_min_z'.

Andre Sailer 2016-12-16

- Declare Beampipe, Mask, and Solenoid driver deprecated, copied to DD4hep

Andre Sailer 2016-12-15

- Add badge to README, fix typos, modify readme for github
- Add Travis Configuration

Frank Gaede 2016-12-15

- link `$(Geant4_LIBRARIES)` to TestSensThickness

Shaojun Lu 2016-12-15

- Added a new parameter `Coil_Yoke_gap` and value `249.0*mm`, to scale 'Yoke_barrel_inner_radius' following 'Coil_outer_radius'.

Frank Gaede 2016-12-13

- link ROOT libraries to TestSensThickness

Andre Sailer 2016-12-06

- Add tests to check that sensitiveThickness is correctly set for the different drivers used in latest clic models

- set thicknessSensitive for VertexEndcap_o1_v[456] datastructs

- set thicknessSensitive for TrackerEndcap_o2 datastructs

Marko Petric 2016-12-06

- Add comment to TrackerBarrel_o1_v03 and TrackerBarrel_o1_v04 to explain the difference

Daniel Jeans 2016-12-06

- enabled service drivers in ILD_l1 and ILD_s1 models

Marko Petric 2016-12-05

- Add correct version number

Marko Petric 2016-12-02

- Revert the Vertex endcap geometry back to previous one and use new one in the new model

- Add new model for the updated material budget of the tracker

- Inherit compiler flags from DD4hep so that TLS is inherited and drop ROOT5

Marko Petric 2016-11-30

- Update design to use the new driver

- Updated vertex endcap design with corrected airflow

Frank Gaede 2016-11-30

- - update release notes and version for v00-09-01

Daniel Jeans 2016-11-30

- bug fixes (character to int conversion; strip layer configuration)

Andre Sailer 2016-11-28

- Add sensitive thickness to DataStructs for TrackerEndcap drivers

Bogdan Pawlik 2016-11-24

- updated LumiCal for ILD_s1_v01
- store layer phi stagger in LayeredCalorimeterData

Shaojun Lu 2016-11-24

- - follow the DD4hep update of DDRec::LayeredCalorimeterData::Layer - remove usage of DDRec::LayeredCalorimeterData::Layer.thickness - replace where needed with inner/outer_thickness

Dan Protopopescu 2016-11-23

- Files from Amanda
- Updates by Ross
- GitHub copy updated by Ross Gordon McCoy

Shaojun Lu 2016-11-23

- Fix the missing 'EcalEndcapParameters', which is needed in Marlin reconstruction. And the 'layoutType' of 'DDRec::LayeredCalorimeterData' will be used by 'convertToGear' to create the correct layout type 'Endcap' for EcalEndcap.

Marko Petric 2016-11-23

- Unify ReadoutID with the same schema as in o2_v04_p1

Marko Petric 2016-11-22

- Make it forward compatible with CLIC_o3_v07
- The new patched CLIC_o2_v04 with patched readouts
- Add tracker version for patch
- Remove obsolete tracker models
- Make a patch to CLIC_o2_v04

Daniel Jeans 2016-11-22

- fix offset of magic megatiles to give consistent cell indices

- added forgotten xml files to svn...

Andre Sailer 2016-11-21

- SHcalSc04_Barrel_v01: add DetElements for staves, needed for reconstruction

Daniel Jeans 2016-11-21

- bug fix (identified by overlap check)

Shaojun Lu 2016-11-21

- Fix the 'SEcal05_Helpers' to follow the update in the 'DD4hep::DDRec::LayeredCalorimeterStruct::Layer' by removing thickness.
- remove ECAL preshower layer (barrel&endcap) from ILD_l1_v01 and ILD_s1_v01 models
- use SEcal05 model for ILD_o3_v05 (scECAL)
- update ILD_l1_v01 and ILD_s1_v01 to use new SEcal05 driver
- remove SEcal04Hybrid* (second try...)

Daniel Jeans 2016-11-18

- removed SEcal04Hybrid (not compatible with updated Megatile segmentation class); added new SEcal05* drivers

Dan Protopopescu 2016-11-17

- Included end plate, air gaps, and adjusted total layer thickness to 19mm

Shaojun Lu 2016-11-17

- Updated 'CaloPrototype_v01.cpp' to be a more generic driver.

Daniel Jeans 2016-11-17

- remove new SEcal05 drivers until new segmentation class is available
- new SEcal05 drivers

Bogdan Pawlik 2016-11-15

- printout modification
- Set non-zero phi-offset to fix improper number of phi sectors reported by ddsim, 49 instead of 48

Shaojun Lu 2016-11-15

- Clean up the unused variable 'motherVol'. When DD4hep/lcgeo moved to 'mandatory envelope volume', each sub-detector has been placed into 'envelope' volume. The envelope has been implemented into 'DD4hep/DDCore/src/XML/Utilities.cpp', and placed into the 'mother' volume (world volume) there.
- We do not need this user defined envelope shape. The envelope has been implemented into 'DD4hep/DDCore/src/XML/Utilities.cpp'

- fix warning unused variable, and commented out the lines. If the users have further implementation with the unused variable, the lines could be used again.
- fix warning: declaration of 'RailSeparation' shadows a member of 'this'.
- fix many warning for shadowed declaration, and comment out or remove unused variable.
- fix warning for shadowed declaration, and remove unused variable.

Aidan Robson 2016-11-14

- systemID must be 29 (defined by UTIL::ILDDetID) in CaloFaceEndcapSurfacePlugin for MarlinTrk extrapolation to calo surface to work. Not the same as detector id.

Andre Sailer 2016-11-14

- CLIC_o3_v07, LumiCal: Correct segmentation to start first bin at $\phi=0$, not have $\phi=0$ be the center of the 'first' bin which causes problems at $\phi=\pm\pi$

DD4hep

Frank Gaede 2017-01-13

- fix computation of radiation and interaction lengths
- require the density to be specified in the materials xml file

Frank Gaede 2017-01-04

- adopt numbers in test_surface.cc to elements.xml
- use elements.xml generated with plugin in DDTTest
- add ROOT version to dumped element table

Frank Gaede 2016-12-22

- fix test_surfaces by adapting to elements.xml
- fix test_units by including elements.xml

Markus Frank 2016-12-21

- Add plugin to dump the default ROOT element table

Markus Frank 2016-12-20

- Remove obsolete build flags

Markus Frank 2016-12-19

- Add Multi-threading conditions example

Andre Sailer 2016-12-16

- Add drivers for Beampipe, Mask and Solenoid from lcgce, changed name to DD4hep_*

Rosa Simoniello, Frank Gaede 2016-12-15

- add new struct DDRec::NeighbourSurfacesStruct defined for neighbouring surfaces

Frank Gaede 2016-12-14

- fix library pathes in env scripts for macos
- use DD4HEP_LIBRARY_PATH and full lib path on mac
- apply rpath compiler settings to GaudiPluginService
- make compatible w/ Geant4 10.3

Marko Petric 2016-12-13

- Fix compiler flag handling

Daniel Jeans 2016-12-08

- add utility graphicalMaterialScan

Markus Frank 2016-12-07

- Fix compiler error on MacOSX gcc 4.9

Marko Petric 2016-12-07

- Remove few tests from Travis
- move flag to CMAKE_SHARED_LINKER_FLAGS
- Remove the dynamic lookup on runtime for libs on mac

Markus Frank 2016-12-05

- First version of conditions and alignments

Andre Sailer 2016-12-06

- DDTest: fix location to install DDtest header files
- Remove minimum required cmake version from DD4hepBuild, this interferes with other packages depending on DD4hep

Marko Petric 2016-12-02

- Fix missing CLHEP in thisdd4hep.sh
- Clean up FindPYTHON.cmake file
- Change gaudi auto_ptr to unique_ptr since the auto is deprecated
- Fix initAClick

Marko Petric 2016-11-29

- Fix rpath issues on mac with python

- Add custom DynamicPath on mac when loading python libs
- Add function to set the DD4HEP_LIBRARY_PATH

Marko Petric 2016-11-28

- Update all minimum CMake versions
- Fix gitlab builds
- Rename MakeRootMap to MakeGaudiMap for consistency
- Remove ROOT5 things and make appropriate fixed to the usage of ROOTConfig.cmake
- Remove FindROOT.cmake and use from now on ROOTConfig.cmake

Markus Frank 2016-11-30

- New version of conditions handling

Andre Sailer 2016-11-25

- Fix ProductionCut conversion in Geant4Converter.cpp: cut is a range

Markus Frank 2016-11-24

- Fix linker errors on MAC
- Remove compiler warnings on MACOSX
- First fixes to version of alignment constant processing also remove where visited the \$ statements in the files.
They are useless, since git does not support them. Fixed some tests, which no longer properly worked in the conditions area.
- First usable version of alignment constant processing

Markus Frank 2016-11-16

- Try to improve efficiency using C++11 default operations

Markus Frank 2016-11-10

- Simplify opaque data mappings for conditions

PandoraPFANew v03-01-02

for details and release notes see:

Subject: Re: ilcsoft release v01-19-01
Posted by [grenier](#) on Tue, 28 Feb 2017 10:58:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I've QT 5 installed on my machine. ilc software requires QT 4.7.4

Is it possible to build the software against QT 5 ?

If I have a local Geant 4 build linked against QT 5, do I need to reinstall one with QT 4 when installing ilc soft ?

Thanks for the answer.

Subject: iLCSoft release v01-19-02
Posted by [gaede](#) on Thu, 20 Apr 2017 20:31:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft developers release v01-19-02 is available.

Please note that this release requires at least gcc4.9

Please use the corresponding ilcinstall tag for installation:

svn co <https://github.com/iLCSoft/ilcinstall/tags/v01-19-02>

Reference installations are available at

/afs/desy.de/project/ilcsoft/sw/x86_64_gcc49_sl6/v01-19-02 # SL6 gcc 4.9
/cvmfs/ilc.desy.de/sw/x86_64_gcc49_sl6/v01-19-02 # SL6 gcc 4.9

There is also an ILDConfig release v01-19-02 that has steering files for running the new simulation and reconstruction for ILD with this release:

svn co <https://github.com/iLCSoft/ILDConfig/tags/v01-19-02>

For instructions see:

https://github.com/iLCSoft/ILDConfig/blob/master/StandardConfig/lcgeo_current/README.md

Please report problems and questions related to this release in this thread.

For changes wrt. v01-19-01, see:

https://github.com/iLCSoft/iLCInstall/blob/master/doc/release_notes_ilcsoft_v01-19-02.md

Frank, for the ilcsoft team.

Subject: iLCSoft release v01-19-03
Posted by [gaede](#) on Thu, 13 Jul 2017 12:25:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft developers release v01-19-03 is available.

Please note that this release requires at least gcc4.9

Please use the corresponding ilcinstall tag for installation:

svn co <https://github.com/iLCSoft/ilcinstall/tags/v01-19-03>

Reference installations are available at

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc49_sl6/v01-19-03 # SL6 gcc 4.9  
/cvmfs/ilc.desy.de/sw/x86_64_gcc49_sl6/v01-19-03 # SL6 gcc 4.9
```

There is also an ILDConfig release v01-19-03 that has steering files for running the new simulation and reconstruction for ILD with this release:

```
svn co https://github.com/iLCSoft/ILDConfig/tags/v01-19-03
```

For instructions see:

https://github.com/iLCSoft/ILDConfig/blob/master/StandardConfig/lcgeo_current/README.md

Please report problems and questions related to this release in this thread.

For changes wrt. v01-19-02, see:

https://github.com/iLCSoft/iLCInstall/blob/master/doc/release_notes_ilcsoft_v01-19-03.md

Frank, for the ilcsoft team.

Subject: iLCSoft release v01-19-04
Posted by [gaede](#) on Thu, 27 Jul 2017 09:56:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft developers release v01-19-04 is available.

This is a patch release to v01-19-03 (and originally was called v01-19-03.p01) with some fixes for the ILD simulation models.

Please note that this release requires at least gcc4.9

Please use the corresponding ilcinstall tag for installation:

```
svn co https://github.com/iLCSoft/ilcinstall/tags/v01-19-04
```

Reference installations are available at

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc49_sl6/v01-19-04 # SL6 gcc 4.9  
/cvmfs/ilc.desy.de/sw/x86_64_gcc49_sl6/v01-19-04 # SL6 gcc 4.9
```

There is also an ILDConfig release v01-19-04 that has steering files for running the new simulation and reconstruction for ILD with this release:

```
svn co https://github.com/iLCSoft/ILDConfig/tags/v01-19-04
```

For instructions see:

https://github.com/iLCSoft/ILDConfig/blob/master/StandardConfig/lcgeo_current/README.md

Please report problems and questions related to this release in this thread.

For changes wrt. v01-19-03, see:

https://github.com/iLCSoft/iLCInstall/blob/master/doc/release_notes_ilcsoft_v01-19-03.p01.md
(v01-19-04 is identical to v01-19-03.p01)

Frank, for the ilcsoft team.

Subject: iLCSoft release v01-19-05
Posted by [gaede](#) on Thu, 23 Nov 2017 15:49:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft developers release v01-19-05 is available.

Please note that this release requires at least gcc4.9

Please use the corresponding ilcinstall tag for installation:

```
svn co https://github.com/iLCSoft/ilcinstall/tags/v01-19-05
```

Reference installations are available at

```
/cvmfs/ilc.desy.de/sw/x86_64_gcc49_sl6/v01-19-05 # SL6 gcc 4.9
```

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc49_sl6/v01-19-05 # SL6 gcc 4.9
```

There is also an ILDConfig release v01-19-05 that has steering files for running the new simulation and reconstruction for ILD with this release:

```
svn co https://github.com/iLCSoft/ILDConfig/tags/v01-19-05
```

also installed at:

```
/cvmfs/ilc.desy.de/sw/ILDConfig/v01-19-05
```

```
/afs/desy.de/project/ilcsoft/sw/ILDConfig/v01-19-05
```

For short instructions on how to run the software (for ILD) see:

<https://github.com/iLCSoft/ILDConfig/blob/master/StandardConfig/production/README.md>

Note that we have created new standard steering files for all current ILD simulation models in the new location `ILDConfig/StandardConfig/production` !

Please report problems and questions related to this release in this thread.

For the many changes wrt. v01-19-04, see:

https://github.com/iLCSoft/iLCInstall/blob/master/doc/release_notes_ilcsoft_v01-19-05.md

Frank, for the ilcsoft authors.

Subject: iLCSoft production release v02-00
Posted by [gaede](#) on Thu, 19 Apr 2018 11:04:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft production release v02-00 is available.

Please note that this release requires at least gcc4.9

Please use the corresponding ilcinstall tag for installation:

svn co <https://github.com/iLCSoft/ilcinstall/tags/v02-00>

Reference installations are available at

[/cvmfs/ilc.desy.de/sw/x86_64_gcc49_sl6/v02-00 # SL6 gcc 4.9](#)

[/afs/desy.de/project/ilcsoft/sw/x86_64_gcc49_sl6/v02-00 # SL6 gcc 4.9](#)

Please use preferred the cvmfs installations as afs is about to be phased out.

There is also an ILDConfig release v02-00 that has steering files for running the new simulation and reconstruction for ILD with this release:

svn co <https://github.com/iLCSoft/ILDConfig/tags/v02-00>

also installed at:

[/cvmfs/ilc.desy.de/sw/ILDConfig/v02-00](#)

[/afs/desy.de/project/ilcsoft/sw/ILDConfig/v02-00](#)

For short instructions on how to run the software (for ILD) see:

<https://github.com/iLCSoft/ILDConfig/blob/master/StandardConfig/production/README.md>

Please report problems and questions related to this release in this thread.

For the changes wrt. v01-19-06 (and v01-19-05), see:

https://github.com/iLCSoft/iLCInstall/blob/master/doc/release_notes_ilcsoft_v02-00.md

https://github.com/iLCSoft/iLCInstall/blob/master/doc/release_notes_ilcsoft_v01-19-06.md

Frank, for the ilcsoft authors.

Subject: iLCSoft production patch release v02-00-02
Posted by [gaede](#) on Tue, 04 Sep 2018 11:40:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

a new iLCSoft patch-production release v02-00-02 is available.

Please note that this release requires at least gcc4.9

Please use the corresponding ilcinstall tag for installation:

svn co <https://github.com/iLCSoft/ilcinstall/tags/v02-00-02>

Reference installations are available at

[/cvmfs/ilc.desy.de/sw/x86_64_gcc49_sl6/v02-00-02 # SL6 gcc 4.9](#)

[/afs/desy.de/project/ilcsoft/sw/x86_64_gcc49_sl6/v02-00-02 # SL6 gcc 4.9](#)

Please use preferred the cvmfs installations as afs is about to be phased out.

There is also an ILDConfig release v02-00-02 that has steering files for running the new simulation and reconstruction for ILD with this release:

```
svn co https://github.com/iLCSoft/ILDConfig/tags/v02-00-02
```

also installed at:

```
/cvmfs/ilc.desy.de/sw/ILDConfig/v02-00-02
```

```
/afs/desy.de/project/ilcsoft/sw/ILDConfig/v02-00-02
```

For short instructions on how to run the software (for ILD) see:

<https://github.com/iLCSoft/ILDConfig/blob/master/StandardConfig/production/README.md>

Please report problems and questions related to this release in this thread.

For the changes wrt. v02-00-01 see:

https://github.com/iLCSoft/iLCInstall/blob/master/doc/release_notes_ilcsoft_v02-00-02.md

Frank, for the ilcsoft authors.
