

---

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

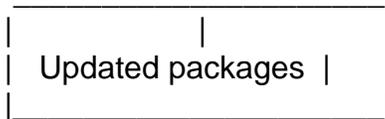
=====

MarlinDD4hep v00-01

=====

by F. Gaede

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



=====

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 = \*lcdd.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 `gdm/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 inside 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 DDRec::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 Icio 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)

=====  
MarlinKinfitt 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 where some dependencies introduced from TPCCondData on MarlinTPC (Exceptions). These where 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 verser.

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 410 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 07:17: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.

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 cvmfs", 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: 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

---