

Installation of an ILCSOFT release on X

Preamble

Trying to build ilcsoft on MacOS X 10.10 (Yosemite)

- You should turn on c++11 otherwise the the CMake logic in PandoraPFANew will fail. Otherwise you need a fix in the PandoraPFANew module.
 - ◆ Problem: CED doesn't build with errors like: `error: non-constant-expression cannot be narrowed from type 'double' to 'GLfloat' (aka 'float') in initializer list [-Wc++11-narrowing]`. It also says: `note: insert an explicit cast to silence this issue`
 - ◆ Solution: add `-Wno-c++11-narrowing` to `CMAKE_CXX_FLAGS` for the CED installation module in `release-base.cfg`:

```
ilcsoft.module("CED").envcmake["CMAKE_CXX_FLAGS"]='-Wall -std=c++11 -Wno-c++11-narrowing'
```

- - ◆ Problem: LCIO, MarlinReco have the same issue, so do it for that module too in `release-ilcsoft.cfg`. Probably should set it on for all packages but it is not known how it would affect them.
 - ◆ Problem: Comparison between pointer and integer for package bbq:

```
/Users/clicdp/ilcsoft/v01-17-07/bbq/v00-01-02/src/TBBQManager.cxx:717:23: error:
comparison between pointer and integer
```

- - ◆ Solution: Edit file to change `false` to `NULL`

*RunTime issues after install:

- - ◆ Problem in MarlinTPC: libraries from `tpconddb` are not copied to `MarlinTPC/lib` so when I try to run `ced2go`, it's missing those libraries.
 - ◆ Solution: symbolically link all the libraries from `MarlinTPC/tpconddb/lib` into `MarlinTPC/lib/`
- Assume to be building locally under:

```
/Users/clicdp/ilcsoft
```

We shall set this as our `ilcsoft_install_prefix` (see later). In here we will checkout `ilcinstall` which will install the base directly under `/Users/clicdp/ilcsoft`.

Dependencies like `Geant4` will be located in places like:

```
/Users/clicdp/ilcsoft/geant4
```

Then, the release will be installed in its own version directory, for example:

```
/Users/clicdp/ilcsoft/v01-17-07-pre06
```

- First of all, check out `ilcinstall` from:

<https://svnsrv.desy.de/public/ilctools/ilcinstall/trunk>

- Will not try to build `Mokka` or `slic` (`LCDD`, `GDML`, etc) therefore will not try to install `MySQL` or `Conddb`. UPDATE: If one does not want to build `MySQL`, they would have to also drop `LCCD`,

MarlinTPC and possibly others. Further, one would have to set MARLIN_LCCD=OFF in the CMake config (via the Marlin installation module). Therefore, we went back and installed MySQL using homebrew which was not so bad.

Prerequisites:

- xcode and svn: to have a working console, checkout packages, etc ..
- Homebrew: used to install wget, boost, Doxygen, CMake (this was problematic in building from scratch on 10.10 for version referenced in the config. Used brew install cmake28 which installs a mildly more recent version) and others (mysql, latex, ...)
- Qt4 framework from <https://download.qt.io/archive/qt/4.8/4.8.5/> . Use 4 to avoid issues, though Geant4 10 should support Qt5. However, to use Qt5, I think we need to move to a more recent version of CMake (it doesn't have a FindQt5.cmake config file). Note that for some reason, qmake returns the wrong include file, or at least CMake sets up the Qt include path to `/usr/include/QtCore` . Therefore a workaround is to do: `sudo ln -s /Library/Frameworks/QtCore.framework/Headers/ /usr/include/QtCore`
- CERNLIB: copy cernlib binaries from `/afs/cern.ch/project/cndoc/wwwasd/cernlib`
- Doxygen: brew install doxygen
- Java and Java SDK: used v 1.8 from oracle. For the time being, it seems like the configuration can't pick up the header files automatically, so you'll have to set it manually in `release-ilcsoft.cfg` (see below).
- x11 (XQuartz)
- mysql: brew install mysql
- latex: There has to be a better way to do this, but I tried with homebrew/cask
 - ◆ `brew install Caskroom/cask/mactex`
 - ◆ `brew cask install mactex`

Changes to configuration:

in `release-versions.py` :

- Change `ilcsoft_install_prefix` to point to your installation (for base and then for the releases):
 - ◆ `ilcsoft_install_prefix = "/Users/clidp/ilcsoft/"`
- Change CLHEP version to 2.1.4.2 from 2.1.4.1 to avoid having to modify `Vectors.cc` (see <https://its.cern.ch/jira/browse/CLHEP-103>)
 - ◆ `CLHEP_version = "2.1.4.2"`
- If you have MySQL installed (I had 5.6.25), change the relevant variables around line 80:

```
# ----- mysql -----
MySQL_version = "5.6.25"
#MySQL_path = ilcPath + "/mysql/" + MySQL_version
MySQL_path = "/usr/local/Cellar/mysql/" + MySQL_version

#if( ilcsoft_afs_path[ arch ] == '/afs/desy.de/project/ilcsoft/sw/x86_64_gcc46_ub1204' ):
#   MySQL_path = "/usr"
```

notice that I commented out the if block starting with `if(ilcsoft_afs_path[arch]...` since it leads to "key error". **If you don't have MySQL installed** and you want to proceed like that, comment out the whole block above.

- Change boost location:

```
#----- boost headers files -----
#Boost_path = "/afs/desy.de/project/ilcsoft/sw/boost/1.58.0"
Boost_path = "/usr/local/Cellar/boost/1.58.0"
```

- Change CERNLIB location:

```
# ----- CERNLIB -----
CERNLIB_version = "2006"
#CERNLIB_path = "/afs/desy.de/project/ilcsoft/sw/x86_64_gcc44_sl6/cernlib/" + CERNLIB_version
CERNLIB_path = "/Users/clicdp/ilcsoft/CERNLIB_2006_MacIntel_gcc4"
```

- Set fortran library path to "" even if c++11 is enabled:

```
Fortran_lib_path = ""
# ----- when using gcc48 we need to give a hint where to find the library:
#if( use_cpp11 ):
#   Fortran_lib_path = "/afs/cern.ch/sw/lcg/contrib/gcc/4.8.1/x86_64-slc6-gcc48-opt/lib64"
```

in release-base.cfg:

- comment out installation of Qt and CMake. Also, point to the qmake executable:

```
geant4.envcmake["QT_QMAKE_EXECUTABLE"]='/usr/bin/qmake'
```

in release-ilcsoft.cfg:

- Comment out use of CMake (will be picked up from system, since installed by brew)
- Comment out linking to Qt, CondDBMySQL
- Comment out linking to HepPDT, GDML, LCDD, SLIC, SlicPandora
- Comment out installation of Mokka and configuration of Mokka installation module
- Manually define a variable for `Java_path` and fill it appropriately. In my case, I had to add the following around line 303 (just before the java check):

```
Java_path="/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home"
```

- Fortran should not be needed. Set `fortran_lib_path=""` in `release-versions.py` and in `release-ilcsoft.cfg` do:

```
ilcsoft.module("MarlinReco").envcmake["MARLINRECO_FORTRAN"]='OFF'
```

- Try with setting c++11 on just for PandoraPFANew and MarlinPandora:

```
◆ ilcsoft.module("PandoraPFANew").envcmake["CMAKE_CXX_FLAGS"]='-std=c++11 -Wall'
◆ ilcsoft.module("MarlinPandora").envcmake["CMAKE_CXX_FLAGS"]='-std=c++11 -Wall'
```

---+++ Changes to code:

- Needed to modify PandoraAnalysis to add `#include <unistd.h>` for all following files in `/Users/clicdp/ilcsoft/v01-17-07-pre06/PandoraAnalysis/v01-00-01/calibration/`
 - ◆ `ECalDigitisation_ContainedEvents.cc`
 - ◆ `HCalDigitisation_ContainedEvents.cc`

in `release-versions.py` :

ILCSOFTMacOSX < CLIC < TWiki

- ◆ HCalDigitisation_DirectionCorrectionDistribution.cc
- ◆ HCalDigitisation_Ring.cc
- ◆ PandoraPFACalibrate_EMScale.cc
- ◆ PandoraPFACalibrate_HadronicEnergyGaussianFit.cc
- ◆ PandoraPFACalibrate_HadronicScale_ChiSquareMethod.cc
- ◆ PandoraPFACalibrate_HadronicScale_TotalEnergyMethod.cc
- ◆ PandoraPFACalibrate_MipResponse.cc
- ◆ RescaleEnergies.cc

probably we will have to commit changes to the packages to handle this - maybe a preprocessor directive to make the change unique to mac.

- Needed to modify CED/v01-09-01/src/server/glced.cc:4667:56:

```
std::cout << "Sorry 100x100 are the max value" << std::endl;
```

instead of

```
std::cout << "Sorry 100x100 are the max value" << std::cout;
```

- Needed to modify BBQ (see above) to eliminate pointer comparison with "false". Edit TBBQManager.cxx lines 717 and 726

Installation

Having fulfilled the above requirements and applied the suggested changes, you can proceed with the installation of the base first and the ilcsoft release later.

To install the base, it is advised not to do it under the ilcsoft release tag, but rather on the ilcsoft top area. This is now done by default.

First check the configuration by doing:

```
./ilcsoft-install releases/v01-17-07/release-base.cfg
```

If all is OK, proceed with the installation:

```
./ilcsoft-install releases/v01-17-07/release-base.cfg -i
```

Then, first check the ilcsoft configuration by doing:

```
./ilcsoft-install releases/v01-17-07/release-ilcsoft.cfg
```

and finally launch the installation by doing:

```
./ilcsoft-install releases/v01-17-07/release-ilcsoft.cfg -i
```

You may have some warnings from the documentation build or even be asked for input when some command is not defined (press enter to continue or q to exit latex). Either handle it manually, or turn the documentation building off.

In building MarlinTPC I got a problem during the linking:

```
ld: unknown option: --no-as-needed clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

in release-ilcsoft.cfg:

I resolved it temporarily by going to `MarlinTPC//build` and issuing `make VERBOSE=1`. At the failing command, I copy the complete compilation-linking command with all the arguments which I edit and remove `-Wl,--no-as-needed`. I executed the modified command and compiled, after which I could go back to the `ilcinstall-trunk` directory and reissue the `ilcsoft` installation command, which resumes (since it finds the objects already). UPDATE: CMake Configuration in MarlinTPC was fixed in HEAD to avoid setting this option for APPLE. You may need to manually change it in your release. Instead of the hack above, just edit `CMakeLists.txt` and surround line 245 with:

```
if (NOT APPLE)
  SET_TARGET_PROPERTIES( generatePedestalRawData PROPERTIES LINK_FLAGS -Wl,--no-as-needed )
endif()
```

Problems/Things to improve

- Make CMake configuration more robust in applying c++11 for modern compilers that support it. Advice to actually use C++11 throughout the installation.
- The documentation does not appear to build properly. Probably a more standard latex installation is needed, but one should consider turning the documentation building off.
- In Building MarlinTPC I get the error:

ld: unknown option: --no-as-needed clang: error: linker command failed with exit code 1 (use -v to see invocation)

UPDATE: CMake Configuration in MarlinTPC was fixed in HEAD to avoid setting this option for APPLE. You may need to manually change it in your release.

- DD4hepExamples: problem with naming of the `libDD4hepSimpleDetector.dylib` (should be `libSimpleDetector.dylib` ?) UPDATE: DD4hepExamples installation module updated to fix this bug. Should be OK
- Bug in `CED/v01-09-01/src/server/glced.cc` line 4667 `std::cout` instead of `std::endl`
- Bug in MarlinTPC: libraries from `tpconddb` are not copied to `MarlinTPC/lib` so when I try to run `ced2go`, it's missing those libraries. Solution: symbolically link all the libraries from `MarlinTPC/tpconddb/lib` into `MarlinTPC/lib/`

-- NikiforosNikiforou - 2015-06-25

This topic: CLIC > ILCSoftMacOSX

Topic revision: r6 - 2015-06-26 - NikiforosNikiforou



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback