MarlinTPC Installation

Updated slides from the MarlinTPC installation session of the April 2009 MarlinTPC tutorial

Martin Killenberg

Uni Bonn / CERN

Last update: 8. 7. 2010

Installation Session

Goal of this session:[1]

- Installing **Marlin** (incl. all dependencies) using **ilcinstall**
- Installing **MarlinTPC** manually from the **subversion** repository
  - Easier to track changes and to update
  - You can make you own processors, e. g. for analysis

We will NOT install

- Software from/for ILCSoft that is not needed for Marlin (Geant, Mokka)
- Additional processor packages for Marlin (especially those who need the CERNLIB, like MarlinReco or MarlinUtil)

[1] This document is especially interesting when you are not running on Scientific Linux. For SL4/5 there are 32 bit binary packages available at http://ilcsoft.desy.de/portal/binary_releases/

Dependencies of Marlin

You need the **development** packages of:

Package	Comes with Distribution	Can be installed by ilcinstall
ROOT	NO	NO
SUN Java	YES	NO
MySQL	YES	NO
cmake	YES	YES
CLHEP	NO	YES
gsl	YES	YES
QT4[2]	YES	YES[3]
LCIO	NO	YES
GEAR	NO	YES
RAIDA	NO	YES
LCCD	NO	YES
CondDBMySQL	NO	YES
Marlin_ILC	NO	YES

In addition you need CVS, svn, Python and a C++ compiler.

[2] Optional

[3] Takes VERY long to compile

ilcinstall

ilcinstall is a Python script that simplifies and automates the installation of ILCSoft.

- Starting point for all ILCSoft packages: ilcsoft.desy.de
- Always download the latest tag from the svn repository. Links on the ilcsoft web page might be outdated.
- Configure which packages to install (the main work)
- Run *ilcinstall*

Note:

ILCSoft is continuously evolving. You should keep track of the versions you are installing. Proposed directory structure:

```
ilcsoft
   |
   |---ilcinstall
   |        |
   |        |---v01-09
   |        |---v01-08-01
   |
   |---v01-09
   |
   |---v01-08-01
```

- ilcinstall versions have to be downloaded manually
- version directories in ilcsoft are generated by ilcinstall

The ilcinstall Config File

A few variables you need:

- `ilcsoft = ILCSoft("/usr/local/ilcsoft/v01-09")`
  Specifies the target path of the installation
- `ilcsoft.downloadType = "svn"`
  Recommended when installing HEAD versions so one can access the revision information to know exactly which version is installed
- `ilcsoft.envcmake["BUILD_32BIT_COMPATIBLE" ]="OFF"`
  When compiling on a 64 bit machine you should compile in native 64 bit.[4]

The three main commands:

- `ilcsoft.install( Marlin( "v00-12" ))`
  Install a package with the given version.
- `ilcsoft.use( ROOT( "/usr/local/root/v5.26.00b" ))`
  Use a package already installed on you system.
  - Either use absolute path
  - or give version number (package is linked manually in ILCSoft path)
- `ilcsoft.link( ROOT( "/usr/local/root/v5.26.00b" ))`
  Create symlink to the absolute path in the ILCSoft target directory.
  I never had any use for this.

[4] Now default. Turn on 32 bit compatibility for Fortran code using 32 bit CERNLIB on 64 bit mashines (mainly SL4). This is not needed for MarlinTPC

How To Configure

- Base your config on the latest release in the ilcsoft `releases` directory. This ensures that the package versions work together.
- Never use packages from DESY afs when not running Scientific Linux!
- Only use packages from DESY afs when you have a fast, permanent network connection to DESY.
- If your distribution brings a package: Use it.
- Only install what you really need. Especially packages depending on CERNLIB are known to make trouble during installation.
- If running on 64 bit Linux install in native 64 bit mode. Only use 32 bit compatibility if required (e. g. for CERNLIB). In this case you need 32 bit compatibility packages for **all** dependencies.

Using Software From Your System

ilcinstall searches for libraries and headers located in `bin`, `lib` and `include` relative to the directory you specify.

Example: `libgsl.so` is located in `/usr/lib`
⇒ `ilcsoft.use( GSL( "/usr"))`

Special Cases: QT4, Java, CMake
These packages are detected automatically if they are installed in your system. Just comment them out in the config.

Installing on 64 Bit Linux

On 64 bit Linux the libraries usually are located in `lib64`, so ilcinstall does not find them.
Trick: Create a directory with symlinks to the directories of your distribution. Name it after the version which is installed on your system and place it in the target directory and link the `lib64` directory to `lib`:

Example:

```
ilcsoft/v01-09> mkdir -p gsl/1.12
ilcsoft/v01-09/gsl/1.12> ln -s /usr/include .
ilcsoft/v01-09/gsl/1.12> ln -s /usr/lib64  lib
ilcsoft/v01-09/gsl/1.12> ln -s /usr/bin .
```

Now you can use : `ilcsoft.use( GSL( "1.12"))`

Note:
The current head of the ilcinstall trunk finds the 64 bit libraries. If you check out the trunk instead of the v01-09 tag you don't have to care about 64 bit.

Running ilcinstall

ilcinstall can be run in 3 different modes:

- `./ilcsoft-install myconfig.cfg`
  Give a summary what is going to be installed.
- `./ilcsoft-install -p myconfig.cfg`
  Make a “dryrun” and preview the installation.
- `./ilcsoft-install -i myconfig.cfg`
  Perform the installation.

Run the three commands in this order. Each step will complain if anything is missing to succeed.

Compiler Errors

Recent compiler versions have become more and more picky. This leads to the fact that code that compiled perfectly well on older compilers does not work any more.

A good place to look for solutions is the linearcollider forum:
http://forum.linearcollider.org

In ILCSoft v01-09 the the CondDBMySQL_ILC-0-8-1 tag does not compile with gcc 4.4.1. This is solved in the current trunk, so you can change

```
ilcsoft.install( CondDBMySQL( "CondDBMySQL_ILC-0-8-1" ))
```

to

```
ilcsoft.install( CondDBMySQL( "HEAD" ))
```

If you find bugs / errors please report them to the developers, so they can be fixed!

Modifying Your Shell Environment

This step is not necessary, but it's very convenient not to type the complete path all the time when calling Marlin.[5]

Example: If you are using `bash` put the following into your `~/.bashrc`:

```
# The system variable for root
export ROOTSYS=/usr/local/root/v5.26.00b

# For convenience: define ILCPATH and put Marlin and lcio to the path
export ILCPATH=/usr/local/ilcsoft/v01-09
export PATH=$PATH:$ILCPATH/Marlin/v00-12/bin:${ILCPATH}/lcio/v01-51/bin
```

Testing Marlin

```
Marlin -x
```

[5] Instead of setting the variables manually you could also source the `init_ilcsoft.sh` file in your ILCSoft target directory. But this sets environment variables, PATH and LD_LIBRARY_PATH for all packages defined in the ilcinstall config. Especially for the packages from you system (QT, MySQL, cmake, gsl) this does not make sense. The few shown here are usually sufficient.

Installing MarlinTPC

Additional Requirements:

Package	Comes with Distribution	Can be installed by ilcinstall
Minuit2[6]	NO	NO

- Download MarlinTPC from the repository (recommended: use the trunk)

```
svn checkout svn://pi.physik.uni-bonn.de/MarlinTPC/trunk MarlinTPC_trunk
```

- Create a subdirectory named `build`. Change into this directory.
- Run `cmake` to create the Makefiles

```
MarlinTPC_trunk/build> cmake -C $ILCPATH/ILCSoft.cmake ..
```

  (don't forget the two dots at the end)

- Run `make`
- Set the `MARLIN_DLL` variable. You need the Minuit2 and the MarlinTPC library. Minuit2 has to be loaded before MarlinTPC. For instance in bash:

```
export MARLIN_DLL=$ROOTSYS/lib/libMinuit2.so:$HOME/MarlinTPC/build/lib/libMarlinTPC.so
```

  For convenience you should add it to your `~/.bashrc`

- Run `Marlin -x` to see if the library is there

[6] Usually comes with ROOT

Kalman Filter

Since r2113 the `TrackFitterKalmanProcessor` and `TrackMakingKalmanFilterProcessor` are available.

They introduce new dependencies:

Package	Comes with Distribution	Can be installed by ilcinstall
KalTest	NO	NO
KalDet	NO	NO

Due to the new dependencies these processors are not build by default, so MarlinTPC still can compile without `KatTest` and `KatDet`.
To use the Kalman filters `KatTest` and `KatDet` have to be installed manually and some environment variables have to be set before compiling MarlinTPC.

Please read `reconstruction/READE` in your MarlinTPC directory for instructions.

Installing MySQL and Minuit2 Manually

MySQL:

Download the source code from

http://dev.mysql.com/get/Downloads/MySQL-5.1/mysql-5.1.31.tar.gz/from/pick#mirrors

---

Minuit2

Usually Minuit2 comes with root. But there is also a stand-alone version:
http://cern.ch/project-mathlibs/minuit/release/download.html

On this page you can also find installation instructions.