MarlinTPC Tutorial
Installation Session

Martin Killenberg

University of Bonn

11. February 2009

Installation Session

Goal of this session:

- 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)

universitätbonn

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
QT4[1]	YES	YES[2]
LCIO	NO	YES
GEAR	NO	YES
RAIDA	NO	YES
LCCD	NO	YES
CondDBMySQL	NO	YES
Marlin_ILC	NO	YES

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

[1] Optional

[2] 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 CVS tag from the 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 Links on the ilcsoft web page might be outdated.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 structure:

```
ilcsoft
  |
  |---ilcinstall
  |        |
  |        |---v01-06
  |        |---v01-07
  |
  |---v01-06
  |
  |---v01-07
```

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

universitätbonn

The ilcinstall Config File

A few variables you need:

- `ilcsoft = ILCSoft("/usr/local/ilcsoft/v01-06")`
  Specifies the target path of the installation
- `ilcsoft.useCMake = True`
  This always has to be on, the old *GNUmakefile* based installation is not supported any more
- `ilcsoft.envcmake["BUILD_32BIT_COMPATIBLE" ]="OFF"`
  When compiling on a 64 bit machine you should compile in native 64 bit.[3]

The three main commands:

- `ilcsoft.install( Marlin( "v00-10-04" ))`
  Install a package with the given version.
- `ilcsoft.use( ROOT( "/usr/local/root/v5.22.00" ))`
  Use a package already installed on you system.
  - Either use absolute path
  - or give version number (package is linked in ILCSoft path)
- `ilcsoft.link( ROOT( "/usr/local/root/v5.22.00" ))`
  Create symlink to the absolute path in the ILCSoft target directory.
  I never had any use for this.

[3] Per default 32 bit compatibility is on so the old Fortran CERNLIB can be used. This is not needed for MarlinTPC

universitätbonn

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.

ilcinstall searches for libraries and header 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 Case: QT4

QT4 needs an additional directory `mkspecs`. When it comes with you distribution this is not necessarily in the same place as the `bin`, `lib` and `include`.

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:

```
ilcsoft/v01-06> mkdir -p QT/4.4.3
ilcsoft/v01-06/QT/4.4.3> ln -s /usr/include .
ilcsoft/v01-06/QT/4.4.3> ln -s /usr/lib .
ilcsoft/v01-06/QT/4.4.3> ln -s /usr/bin .
ilcsoft/v01-06/QT/4.4.3> ln -s /usr/share/qt4/mkspecs .
```

Now you can use `ilcsoft.use( QT( "4.4.3"))`

universitätbonn

Installing on 64 Bit Linux

On 64 bit Linux the libraries usually are located in `lib64`, so ilcinstall does not find them. We use the same trick as for QT4:

Example:

```
ilcsoft/v01-06> mkdir -p gsl/1.11
ilcsoft/v01-06/gsl/1.11> ln -s /usr/include .
ilcsoft/v01-06/gsl/1.11> ln -s /usr/lib64  lib
ilcsoft/v01-06/gsl/1.11> ln -s /usr/bin .
```

```
ilcsoft.use( GSL( "1.11"))
```

Running ilcinstall

ilcinstall can be run in 3 different modes:

- `./ilcsoft-install -s 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.

Currently the latest GEAR and LCCD tags have problems with gcc 4.3. A good place to look for solutions is the linearcollider forum:
http://forum.linearcollider.org

A solution to the gcc 4.3 problem can be found there:
http://forum.linearcollider.org/index.php?t=tree&goto=1658

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

universitätbonn

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.

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

```
# The system variable for root
export ROOTSYS=/usr/local/root/v5.22.00

# For convenience: define ILCPATH and put Marlin and lcio to the path
export ILCPATH=/usr/local/ilcsoft/v01-06
export PATH=$PATH:$ILCPATH/Marlin/v00-10-04/bin:${ICLPATH}/lcio/v01-11/bin
```

Testing Marlin

```
Marlin -x
```

universitätbonn

Installing MarlinTPC

Additional Requirements:

Package	Comes with Distribution	Can be installed by ilcinstall
Minuit2[4]	NO	NO
subversion	YES	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

[4] Usually comes with ROOT

universitätbonn

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.