Build System – Overview

The “build system” is a collection of CMake scripts and guidelines that is intended to simplify the development of internal workgroup projects. The main goals of the build system are:

* Support cross-platform building (Windows/Linux).
* Provide automatic generation of project files for IDEs (e.g. Visual Studio)
* Keeping per-project configuration information as compact as possible (i.e. source files, target name and list of libraries).

The build system supports 2 types of projects (only C/C++ language is supported):

* Applications
* Static libraries

Each project should reside in its own subdirectory and have a CMakeLists.txt file containing one target. Libraries have the simplest CMakeLists.txt template:

**ADD\_LIBRARY(<lib\_name> STATIC <file1> <file2> <...>)**

Alternatively, a library project can include a whole directory of source files:

**AUX\_SOURCE\_DIRECTORY(<directory> <directory>\_SOURCES)**

**ADD\_LIBRARY(<lib\_name> STATIC ${<directory>\_SOURCES})**

Every application project should define a slightly more complex CMakeLists.txt file:

**cmake\_minimum\_required(VERSION 2.8.3)**

**project(<app\_name>)**

**include(<relative\_path\_to\_buildsys>/CMakeShared.txt)**

**include\_library(<path\_to\_lib1> <lib1\_name>)**

**include\_ftdi\_library() #Only if the project needs libft2xx**

**ADD\_EXECUTABLE(<app\_name> <source1> <source2> <...>)**

**TARGET\_LINK\_LIBRARIES(<app\_name> <ftd2xx> <lib1\_name> <...>)**

To build an application, create a subdirectory (**cmakebuild** is recommended) and run “**cmake ..**” from it. All generated files will be put inside the directory and can be easily removed by deleting it.

Please follow the following guidelines when creating new projects:

* 1 target per directory. This makes generated IDE project files more usable.
* Put reusable cmake code (e.g. setting SSE2 flags) in separate files and include them from your CMakeLists.txt.
* If a library contains many include files, put them into a subdirectory. E.g. **include\_library(../lib1)** will add “../lib1” to include search path. I.e. a file “../lib1/libsomething/foo.h” should be included with “#include<libsomething/foo.h>”.