Building IMF Tool under Mac OS X (also applicable to Linux)

Wolfgang Ruppel, 2017-02-27

Preparations

Create directories for asdcplib, IMF Tool and for Debugging with Eclipse:

```
drwxr-xr-x  17 wolfgangruppel  staff   578 Mar 18 13:24 asdcplib
drwxr-xr-x  10 wolfgangruppel  staff   340 Mar 18 13:39 imftool
drwxr-xr-x   5 wolfgangruppel  staff   170 Mar 18 13:39 imftool-debug
```

Prerequisites

You will need a functional build environment on your system. This chapter is for additional libraries only.

Git:
Download & install Git from for your system from https://git-scm.com/download/mac

Cmake:
Download: https://cmake.org/download/
Binaries are provided for various operating systems, e.g. MacOS

Xerces
Download source code from
http://artfiles.org/apache.org//xerces/c/3/sources/xerces-c-3.1.3.tar.gz
Unpack to temp directory, configure, build and install:

```
tar xzvf ../Downloads/xerces-c-3.1.3.tar.gz
cd xerces-c-3.1.3/
./configure
make
sudo make install
```

Qt5
Download: http://www.qt.io/download/
Choose Open Source Distribution / YES / YES
Download installer
Run installer, deselect iOS and Android (saves 10 GB of disk space!)

libxsd
Download binaries from
http://www.codesynthesis.com/products/xsd/download.xhtml

```
tar xzvf ../Downloads/xsd-4.0.0-i686-macosx.tar.bz2
```

Set LibXSD_root_DIR in Cmake to point to that directory plus libxsd/, e.g.
/Users/wolfgangruppel/src/xsd-4.0.0-i686-macosx/libxsd/

OpenJPEG
Download and build OpenJPEG from https://github.com/uclouvain/openjpeg/
Note: OpenJPEG 2.2 (with multi-threading support) is required!

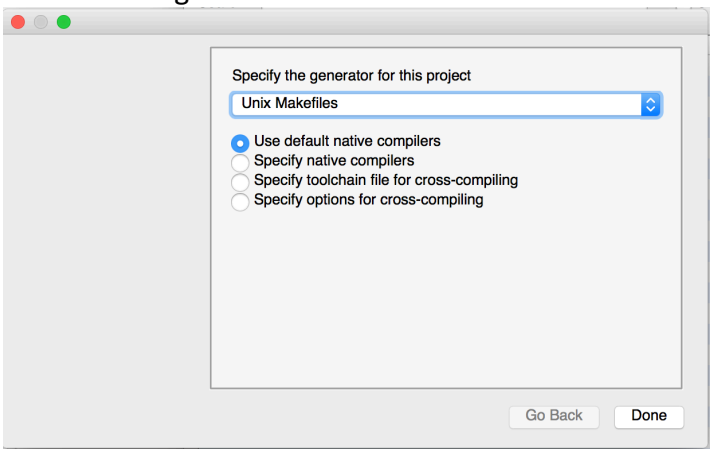
Installing and downloading asdcplib

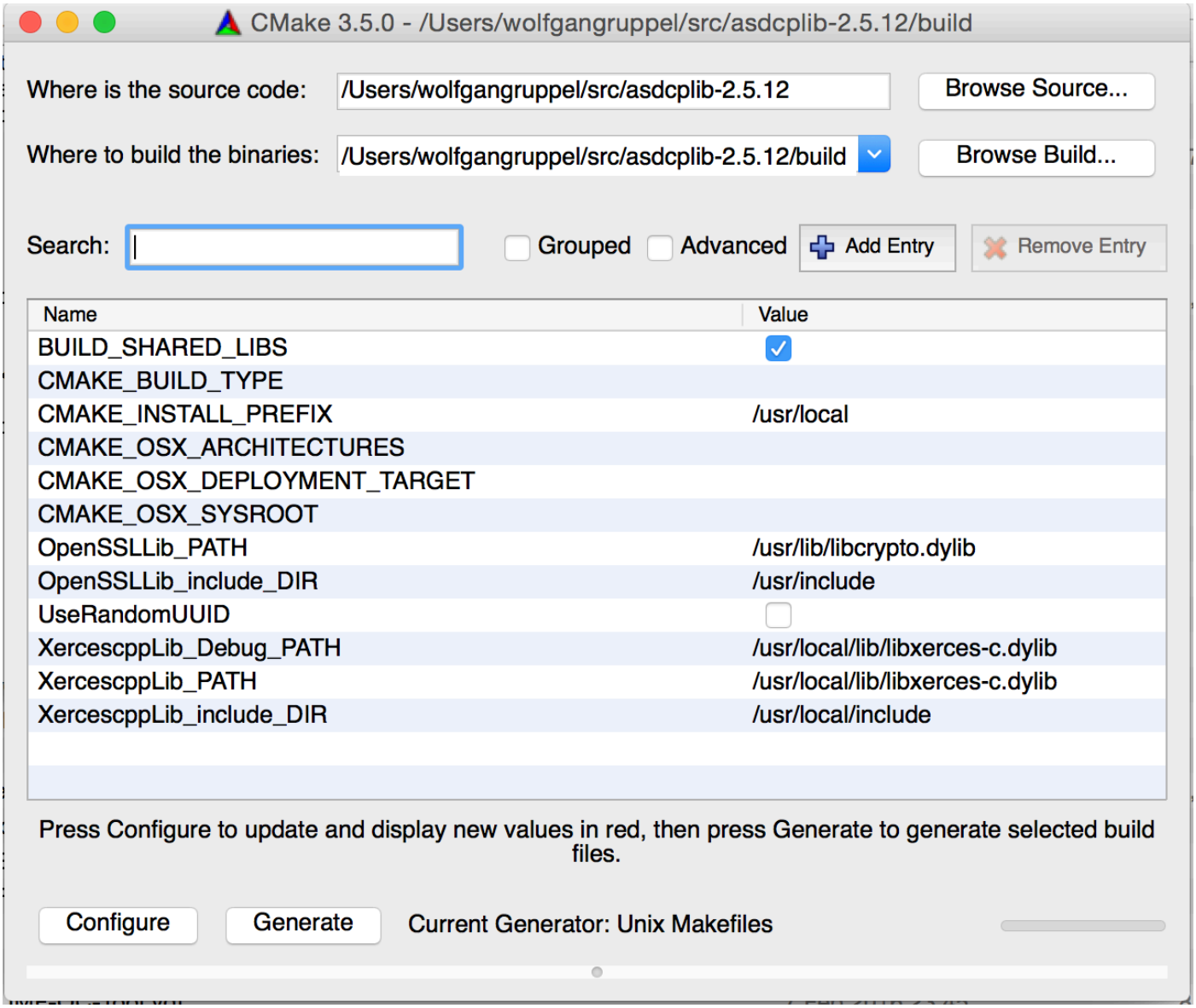
Asdcplib is an Open Source Library for reading and writing A-02 MXF files.
Download asdcplib-2.7.19 from
http://download.cinecert.com/asdcplib/asdcplib-2.7.19.tar.gz
Extract archive (tar –xvzf)

Linux command line configuration and build:

```
cd asdcplib2.9.17/
mkdir build/
cd build/
cmake ..
make && sudo make install
```

Configuration using CMake (Mac) / cmake-gui (Linux):
Start CMake from the Applications menu (Mac) or start cmake-gui (Linux)
CMake settings:





Press “Configure” (twice) , choose “UNIX Makefile” and “Generate”.

Building asdcplib (from asdcplib-2.5.14/):

```
cd build/
make && sudo make install
```

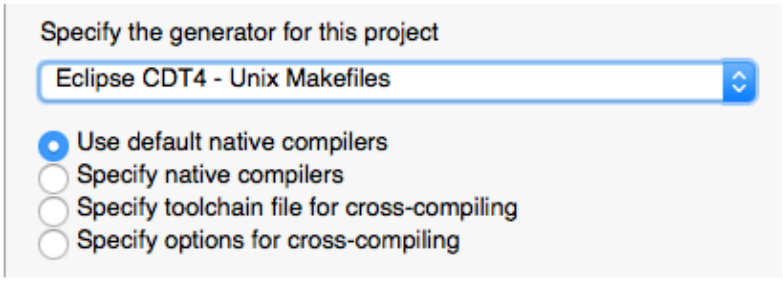
Installing and downloading IMF Tool

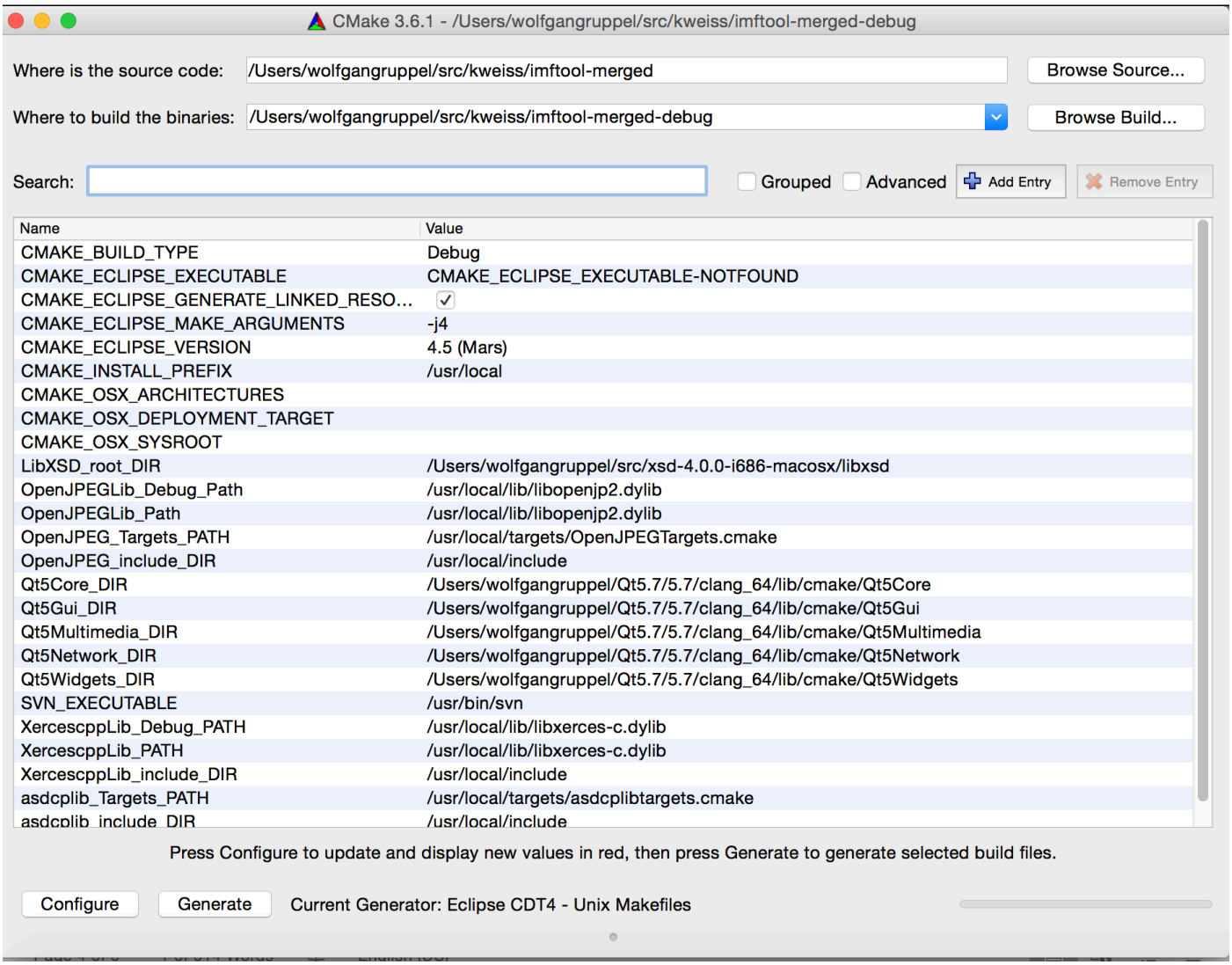
```
cd imftool/
git clone http://github.com/IMFTool/IMFTool
```

Start CMake

Carefully fill in all PATHs as shown above!
..and set CMAKE_BUILD_TYPE to “Debug”

Press “Configure” and “Generate”, choose “Eclipse CDT4 – Unix Makefiles” as generator.





(Note: I had to manually configure the paths to the QT directories and the libXSD directory.)

```
cd imftool-debug/
make
```

Run IMF-Tool:

```
nohup ./src/IMF-Tool&
```

Debugging using Eclipse MARS on MacOS Yosemite

Install Homebrew: http://brew.sh

Install gdb:

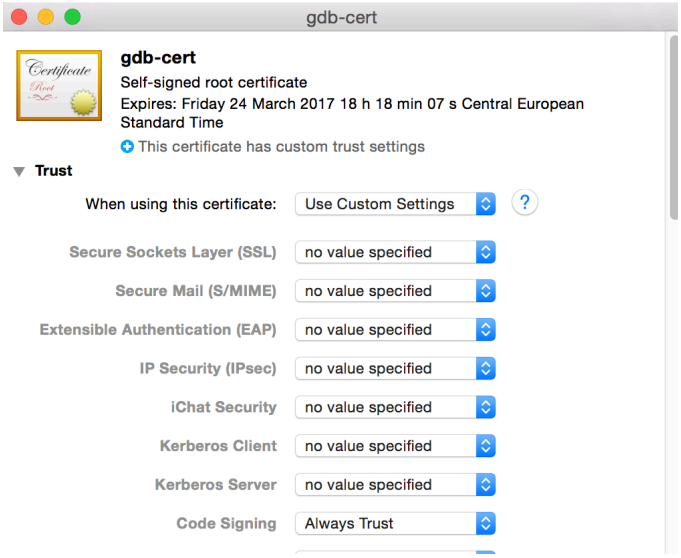
```
brew update
brew tap homebrew/dupes
brew install gdb
```

Signing the gdb binary (from http://andresabino.com/2015/04/14/codesign-gdb-on-mac-os-x-yosemite-10-10-2/)

To enable gdb access to other processes, we must first code sign the binary. This signature depends on a particular certificate, which the user must create and register with the system.
To create a code signing certificate, open the Keychain Access application. Choose menu Keychain Access -> Certificate Assistant -> Create a Certificate…
Choose a name for the certificate (e.g., gdb-cert), set Identity Type to Self Signed Root, set Certificate Type to Code Signing and select the Let me override defaults. Click several times on Continue until you get to the Specify a Location For The Certificate screen, then set Keychain to System.
Double click on the certificate, open Trust section, and set Code Signing to Always Trust. Exit Keychain Access application.
Restart the taskgated service, and sign the binary.

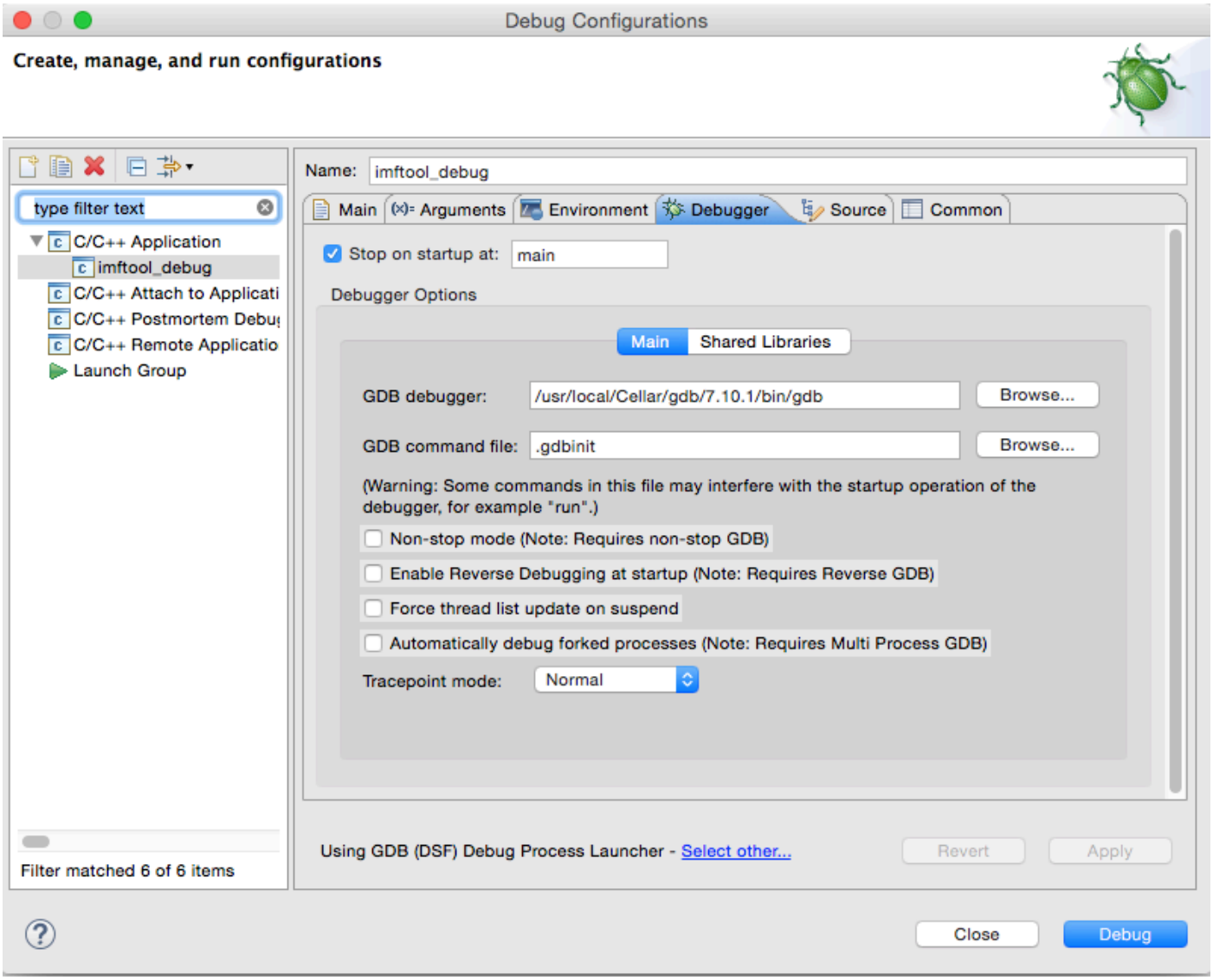
```
$ sudo killall taskgated
$ codesign -fs gdb-cert /usr/local/bin/gdb
```

Certificate Settings



Finally, point eclipse to the gdb binary:

(Input in field “GDB Debugger” was /usr/local/bin/gdb, eclipse has changed the path afterwards)


Debug Configurations
Create, manage, and run configurations
type filter text
C/C++ Application
imftool_debug
C/C++ Attach to Applicati
C/C++ Postmortem Debu
C/C++ Remote Applicatio
Launch Group
Filter matched 6 of 6 items
Name: imftool_debug
Main
Arguments
Environment
Debugger
Source
Common
Stop on startup at: main
Debugger Options
Main
Shared Libraries
GDB debugger: /usr/local/Cellar/gdb/7.10.1/bin/gdb
Browse...
GDB command file: .gdbinit
Browse...
(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)
Non-stop mode (Note: Requires non-stop GDB)
Enable Reverse Debugging at startup (Note: Requires Reverse GDB)
Force thread list update on suspend
Automatically debug forked processes (Note: Requires Multi Process GDB)
Tracepoint mode: Normal
Using GDB (DSF) Debug Process Launcher - Select other...
Revert
Apply
Close
Debug