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

Wolfgang Ruppel, 2019-08-06

Preparations

Create directories for IMF Tool and its dependencies 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
drwxr-xr-x   5 wolfgangruppel  staff   170 Mar 18 13:39 openjpeg
```

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!)

OpenJPEG

Download and build OpenJPEG from https://github.com/uclouvain/openjpeg/
Note: OpenJPEG 2.2 (with multi-threading support) is required!
Use CMake for configuration.
Important: Also do a “sudo make install” to install the binaries in /usr/local
Installed openjpeg binaries are required subsequently by IMF Tool.

regxmllibc

Download tar or zip archive from
https://github.com/sandflow/regxmllib

Configure, build and install using CMake

Only if App#5 support is selected:

Download and install ilmbase from http://www.openexr.com/downloads.html
Download and install openexr from http://www.openexr.com/downloads.html

Installing and downloading asdcplib

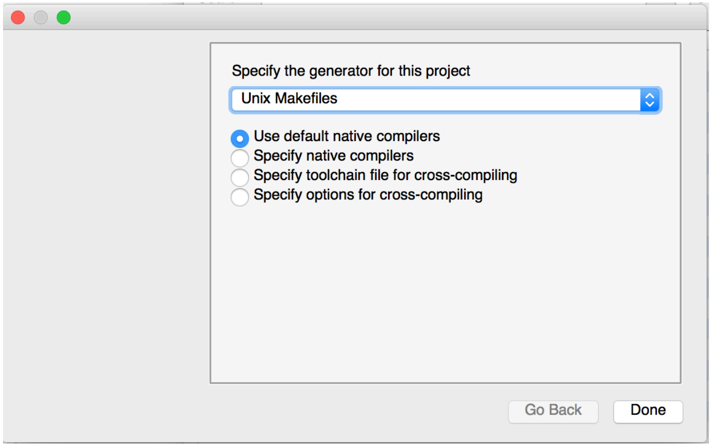
Asdcplib is an Open Source Library for reading and writing A-02 MXF files.
NEW:
For IAB support, a patched version of asdcplib is required, available here:
https://github.com/wruppelx/asdcplib/archive/master.zip

Extract archive (unzip)

Linux command line configuration and build:

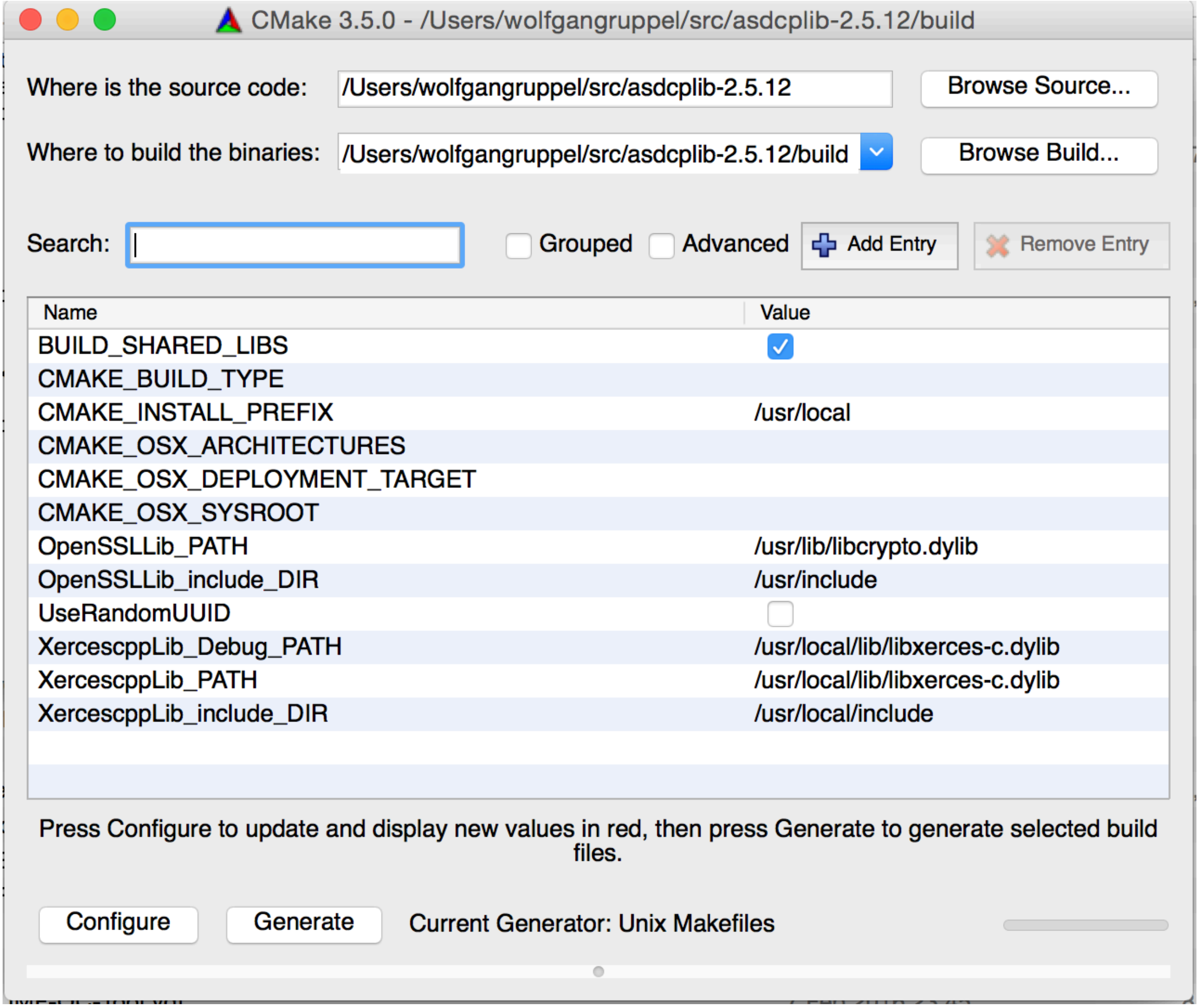
```
cd asdcplib-master/
mkdir build/
cd build/
cmake ..
make && sudo make install
```

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



Specify the generator for this project
Unix Makefiles

● Use default native compilers
○ Specify native compilers
○ Specify toolchain file for cross-compiling
○ Specify options for cross-compiling



CMake 3.5.0 - /Users/wolfgangruppel/src/asdcplib-2.5.12/build

Where is the source code: /Users/wolfgangruppel/src/asdcplib-2.5.12

Where to build the binaries: /Users/wolfgangruppel/src/asdcplib-2.5.12/build

Search:

□ Grouped □ Advanced

Name	Value
BUILD_SHARED_LIBS	☑
CMAKE_BUILD_TYPE	
CMAKE_INSTALL_PREFIX	/usr/local
CMAKE_OSX_ARCHITECTURES	
CMAKE_OSX_DEPLOYMENT_TARGET	
CMAKE_OSX_SYSROOT	
OpenSSLLib_PATH	/usr/lib/libcrypto.dylib
OpenSSLLib_include_DIR	/usr/include
UseRandomUUID	□
XercescppLib_Debug_PATH	/usr/local/lib/libxerces-c.dylib
XercescppLib_PATH	/usr/local/lib/libxerces-c.dylib
XercescppLib_include_DIR	/usr/local/include

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Unix Makefiles

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

Building asdcplib (from asdcplib-2.7.19/):

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

Installing and downloading IMF Tool

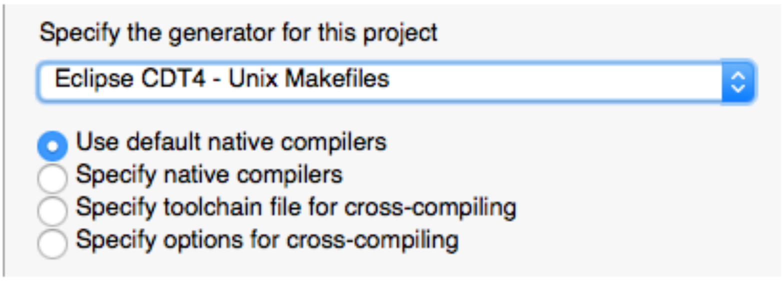
```
cd imftool/
git clone http://github.com/IMFTool/IMFTool
```

Start CMake

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

Important: If you do not deselect BUILD_APP5_SUPPORT, additional dependencies openilm and OpenEXR libraries are required.

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

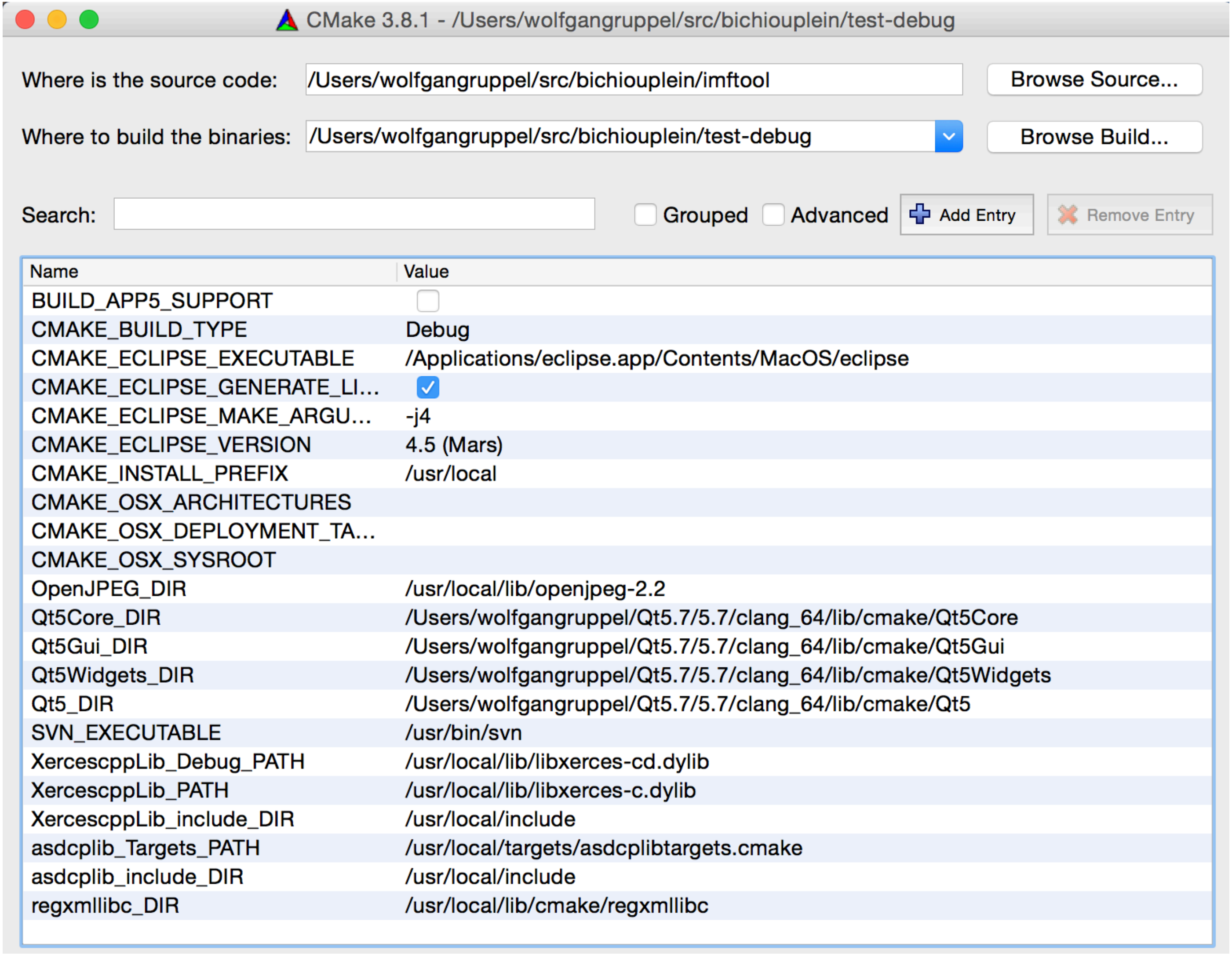


Specify the generator for this project

Eclipse CDT4 - Unix Makefiles

● Use default native compilers
○ Specify native compilers
○ Specify toolchain file for cross-compiling
○ Specify options for cross-compiling

(1) CMake configuration with BUILD_APP5_SUPPORT deselected:



CMake 3.8.1 - /Users/wolfgangruppel/src/bichiouplein/test-debug

Where is the source code: /Users/wolfgangruppel/src/bichiouplein/imftool

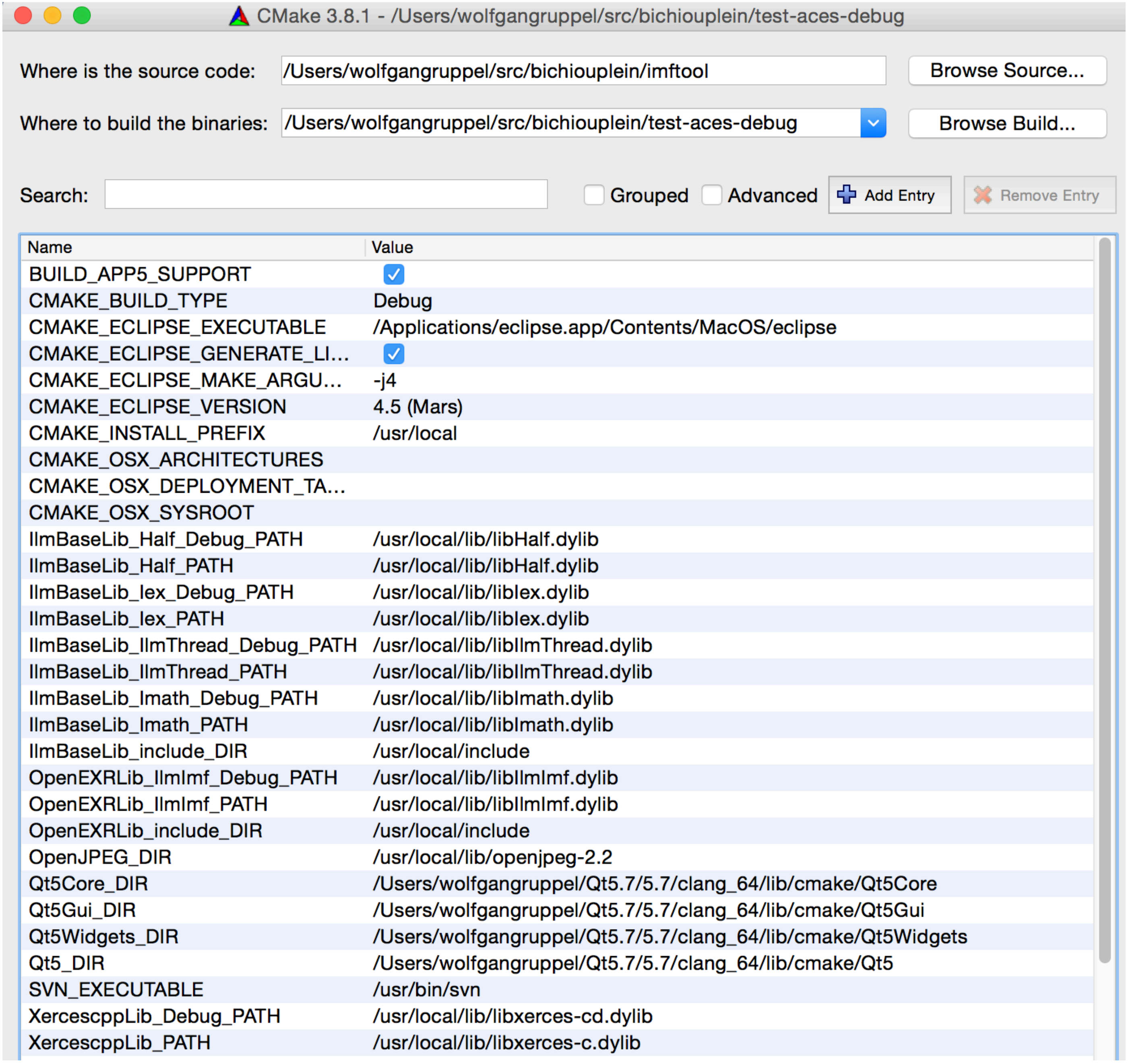
Where to build the binaries: /Users/wolfgangruppel/src/bichiouplein/test-debug

Search:

□ Grouped □ Advanced

Name	Value
BUILD_APP5_SUPPORT	□
CMAKE_BUILD_TYPE	Debug
CMAKE_ECLIPSE_EXECUTABLE	/Applications/eclipse.app/Contents/MacOS/eclipse
CMAKE_ECLIPSE_GENERATE_LI...	☑
CMAKE_ECLIPSE_MAKE_ARGU...	-j4
CMAKE_ECLIPSE_VERSION	4.5 (Mars)
CMAKE_INSTALL_PREFIX	/usr/local
CMAKE_OSX_ARCHITECTURES	
CMAKE_OSX_DEPLOYMENT_TA...	
CMAKE_OSX_SYSROOT	
OpenJPEG_DIR	/usr/local/lib/openjpeg-2.2
Qt5Core_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5Core
Qt5Gui_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5Gui
Qt5Widgets_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5Widgets
Qt5_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5
SVN_EXECUTABLE	/usr/bin/svn
XercescppLib_Debug_PATH	/usr/local/lib/libxerces-cd.dylib
XercescppLib_PATH	/usr/local/lib/libxerces-c.dylib
XercescppLib_include_DIR	/usr/local/include
asdcplib_Targets_PATH	/usr/local/targets/asdcplibtargets.cmake
asdcplib_include_DIR	/usr/local/include
regxmllibc_DIR	/usr/local/lib/cmake/regxmllibc

(2) CMake configuration with BUILD_APP5_SUPPORT selected:



Where is the source code: /Users/wolfgangruppel/src/bichiouplein/imftool

Where to build the binaries: /Users/wolfgangruppel/src/bichiouplein/test-aces-debug

Name	Value
BUILD_APP5_SUPPORT	☑
CMAKE_BUILD_TYPE	Debug
CMAKE_ECLIPSE_EXECUTABLE	/Applications/eclipse.app/Contents/MacOS/eclipse
CMAKE_ECLIPSE_GENERATE_LI...	☑
CMAKE_ECLIPSE_MAKE_ARGU...	-j4
CMAKE_ECLIPSE_VERSION	4.5 (Mars)
CMAKE_INSTALL_PREFIX	/usr/local
CMAKE_OSX_ARCHITECTURES	
CMAKE_OSX_DEPLOYMENT_TA...	
CMAKE_OSX_SYSROOT	
IlmBaseLib_Half_Debug_PATH	/usr/local/lib/libHalf.dylib
IlmBaseLib_Half_PATH	/usr/local/lib/libHalf.dylib
IlmBaseLib_Iex_Debug_PATH	/usr/local/lib/libIex.dylib
IlmBaseLib_Iex_PATH	/usr/local/lib/libIex.dylib
IlmBaseLib_IlmThread_Debug_PATH	/usr/local/lib/libIlmThread.dylib
IlmBaseLib_IlmThread_PATH	/usr/local/lib/libIlmThread.dylib
IlmBaseLib_Imath_Debug_PATH	/usr/local/lib/libImath.dylib
IlmBaseLib_Imath_PATH	/usr/local/lib/libImath.dylib
IlmBaseLib_include_DIR	/usr/local/include
OpenEXRLib_IlmImf_Debug_PATH	/usr/local/lib/libIlmImf.dylib
OpenEXRLib_IlmImf_PATH	/usr/local/lib/libIlmImf.dylib
OpenEXRLib_include_DIR	/usr/local/include
OpenJPEG_DIR	/usr/local/lib/openjpeg-2.2
Qt5Core_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5Core
Qt5Gui_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5Gui
Qt5Widgets_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5Widgets
Qt5_DIR	/Users/wolfgangruppel/Qt5.7/5.7/clang_64/lib/cmake/Qt5
SVN_EXECUTABLE	/usr/bin/svn
XercescppLib_Debug_PATH	/usr/local/lib/libxerces-cd.dylib
XercescppLib_PATH	/usr/local/lib/libxerces-c.dylib

(Note: For finding Qt5, only Qt5_DIR needs to be filled in.)

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

Run IMF-Tool:

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

Annex: Debugging using Eclipse MARS on MacOS Yosemite (outdated)

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…

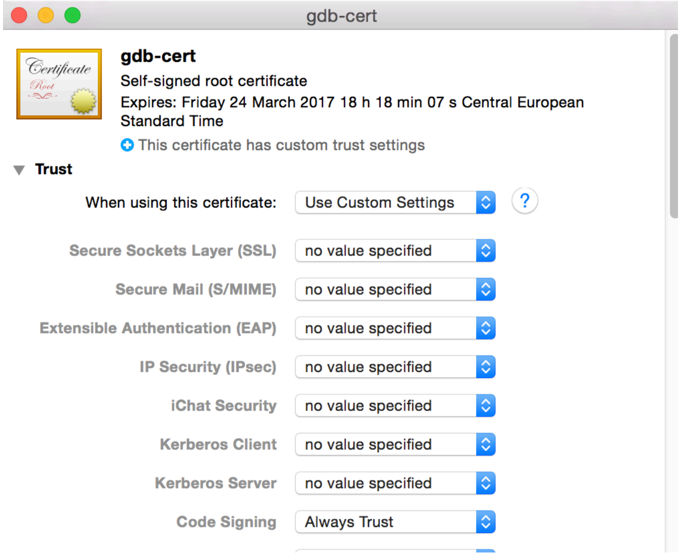
Choose a name for the certificate (e.g., gdb-cert), set Identity Type to Self Signed Root, set Certificate Type to Code Signing and select the Let me override defaults. Click several times on Continue until you get to the Specify a Location For The Certificate screen, then set Keychain to System.

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



gdb-cert

Self-signed root certificate

Expires: Friday 24 March 2017 18 h 18 min 07 s Central European Standard Time

This certificate has custom trust settings

Trust

When using this certificate: Use Custom Settings

Secure Sockets Layer (SSL) no value specified

Secure Mail (S/MIME) no value specified

Extensible Authentication (EAP) no value specified

IP Security (IPsec) no value specified

iChat Security no value specified

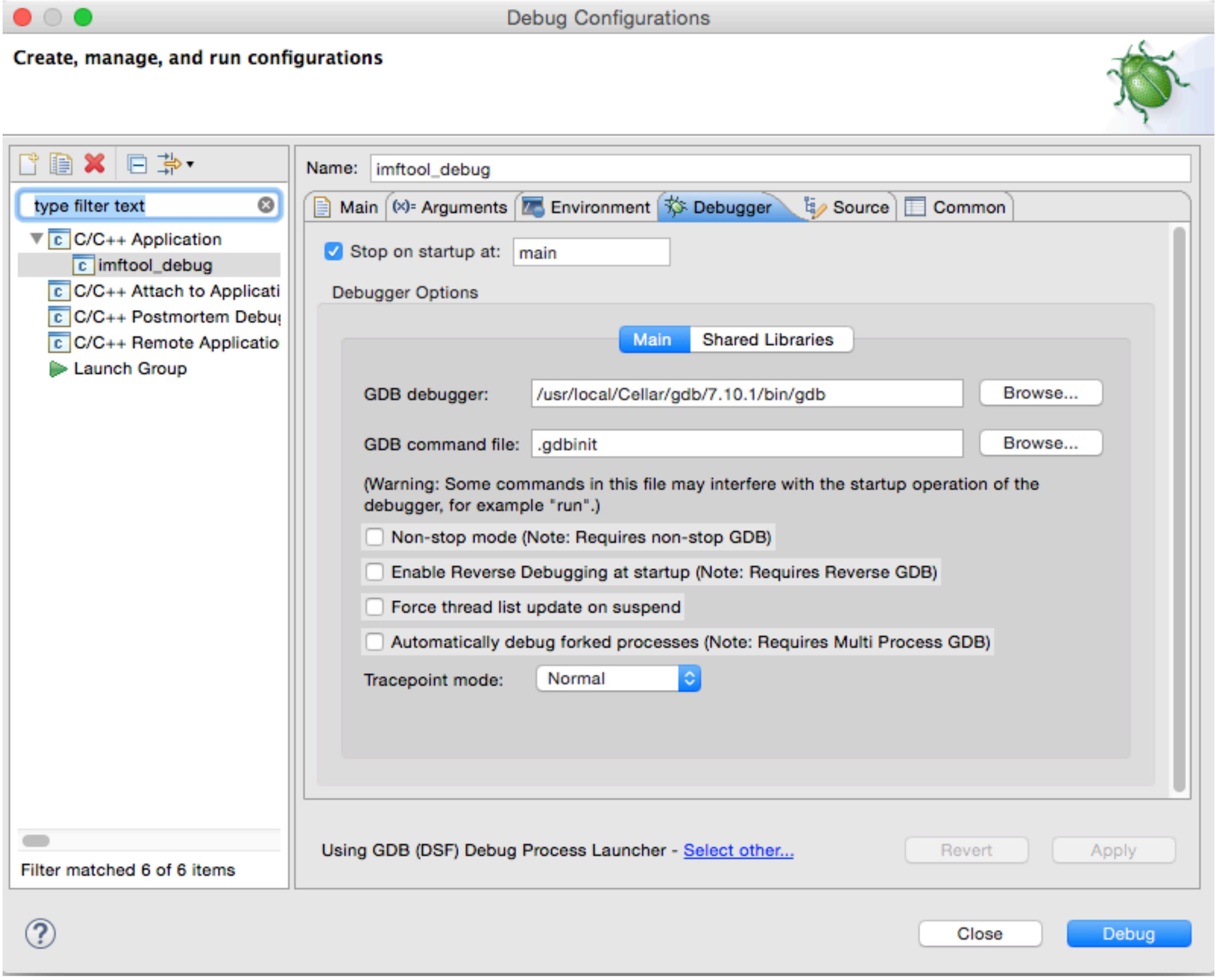
Kerberos Client no value specified

Kerberos Server no value specified

Code Signing Always Trust

Finally, point eclipse to the gdb binary:

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



Name: imftool_debug

☑ Stop on startup at: main

GDB debugger: /usr/local/Cellar/gdb/7.10.1/bin/gdb

GDB command file: .gdbinit

(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...