Improved VES/KiwiViewer build with Eclipse and CMake on Linux, Mac, and Windows

VES (http://ves.vtk.org) is VTK for Embedded Systems. It is a C++ rendering library for mobile devices using OpenGL ES 2.0. VES integrates with the Visualization Toolkit (VTK) to deliver:

- scientific and medical visualization capabilities
- remote rendering and/or data hosting
- modern/novel/new/collaborative interaction techniques

Recently, we made improvements to building VES (specifically KiwiViewer) for Android on Linux and Mac and Windows using CMake and Eclipse. The build procedure requires Android NDK, Android SDK, and Java JDK to be installed on the system. VES includes an example application called KiwiViewer. This blog will describe how to build VES, and then build KiwiViewer. The blog will also cover how to install and setup the correct Android development environment.

Installing the development environment

Linux/Mac

Windows

Download and install the Android SDK from here: http://developer.android.com/sdk/index.html
You should download and install the ADT Bundle for Windows. This will give you a .zip file. There is no executable installer. You will just have to unzip this into a directory. For this example I unziped this file into C:/Users/hoffman/android-windows.

Then the NDK
http://developer.android.com/tools/sdk/ndk/index.html

Need to get Java jdk as well
mine is installed here: c:/Program\ Files\ \(x86\)/Java/

Get a copy of VES

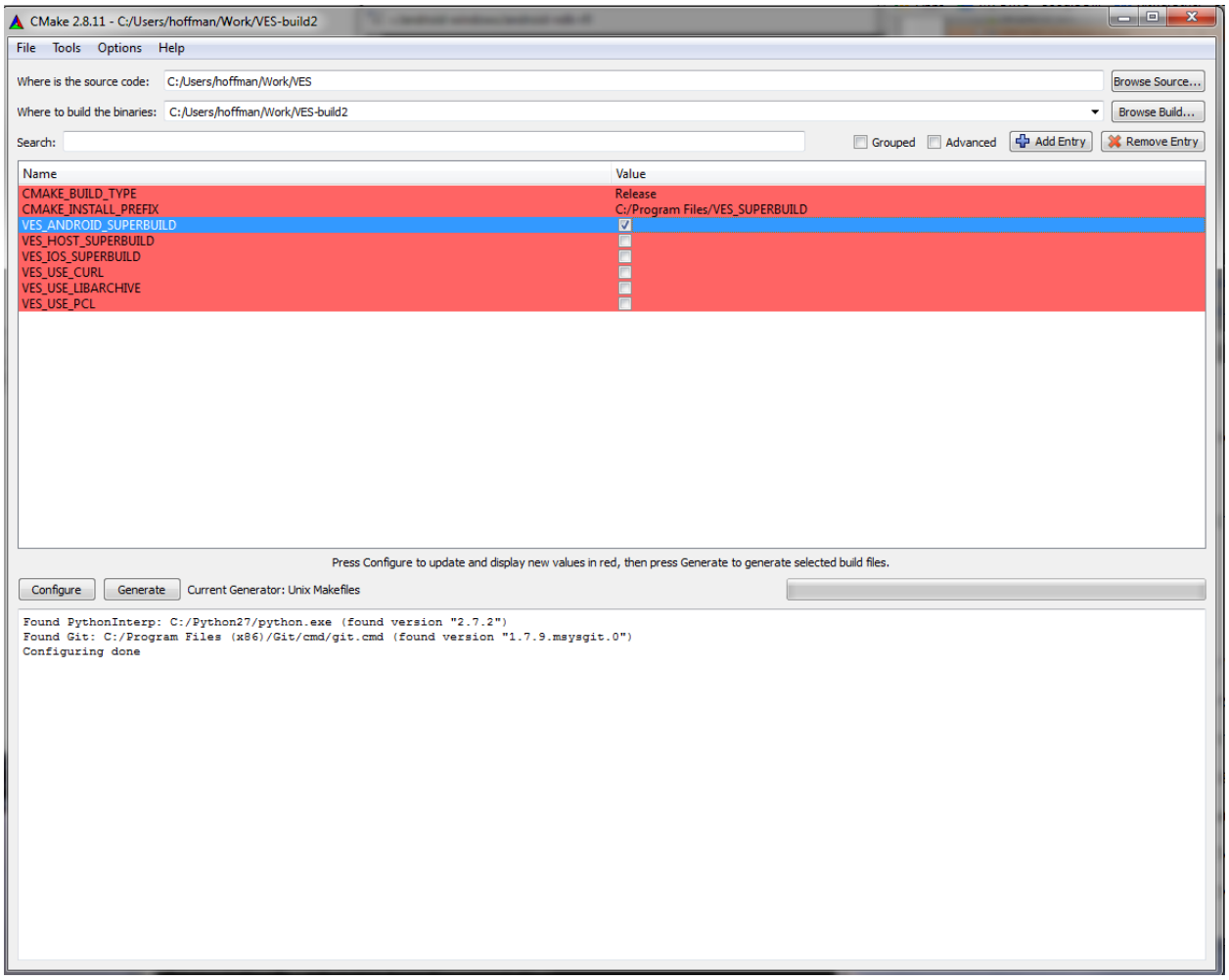
```
git clone git://vtk.org/stage/VES.git
```

```
git checkout improve_ves_android
```

Configure VES with CMake

open a VS Command Prompt
set ANDROID_NDK=c:\Users\hoffman\android-windows\android-ndk-r9
run cmake-gui from command line
"\Program Files (x86)\CMake 2.8\bin\cmake-gui.exe"

turn on VES_ANDROID_SUPERBUILD



Run jom in build directory or nmake if you want to wait. :)

cd ‘build dir from above’
\Users\hoffman\jom\jom.exe

Build Kiwi

Run cmake-gui on Kiwi, be sure to use toolchain option
CMAKE_TOOLCHAIN_FILE:FILEPATH=C:/Users/hoffman/Work/VES/CMake/toolchains/android.toolchain.cmake

Use the cache it creates:
c:\Users\hoffman\Work\VES4\VES-build\Apps\KiwiAndroid>"\Program Files (x86)\CMake 2.8\bin\cmake-gui.exe" ..\..\..\VES\Apps\Android\Kiwi

ANDROID_EXECUTABLE:FILEPATH=C:/Users/hoffman/android-windows/adt-bundle-windows-x86_64-20130917/sdk/tools/android.bat
ANT_EXECUTABLE:FILEPATH=C:/Users/hoffman/android-windows/adt-bundle-windows-x86_64-20130917/eclipse/plugins/org.apache.ant_1.8.3.v201301120609/bin/ant.bat

Put javac in the PATH and set JAVA_HOME

set PATH=%PATH%;c:\Program Files (x86)\Java\jdk1.7.0_03/bin
set JAVA_HOME=c:\Program Files (x86)\Java\jdk1.7.0_03
prevent java from running out of memory during compile
set _JAVA_OPTIONS="-Xmx256M"

run nmake or jom in build tree

run eclipse from the command line
\Users\hoffman\eclipse\eclipse\eclipse.exe

DONE

AndroidManifest.xml is only done at configure time, this needs to be moved to build time, so that clean works.

On ubuntu you will need 32 bit libs:
http://developer.android.com/sdk/installing/index.html

- If you need help installing and configuring Java on your development machine, you

might find these resources helpful:

1. https://help.ubuntu.com/community/Java
2. https://help.ubuntu.com/community/JavaInstallation

- Here are the steps to install Java and Eclipse, prior to installing the Android SDK and ADT Plugin.
    1. If you are running a 64-bit distribution on your development machine, you need to install the ia32-libs package using apt-get::
    2. apt-get install ia32-libs
    3. Next, install Java:
    4. apt-get install sun-java6-jdk
    5. The Ubuntu package manager does not currently offer an Eclipse 3.6 version for download, so we recommend that you download Eclipse from eclipse.org (http://www.eclipse.org/ downloads/). A Java or RCP version of Eclipse is recommended.
    6. Follow the steps given in previous sections to install the SDK and the ADT plugin.

Information for other platforms

---- OLD STUFF MOST LIKELY can delete soon

Tons of trouble getting java to work from eclipse…

BUILD FAILED
C:\Users\hoffman\android-windows\adt-bundle-windows-x86_64-20130917\sdk\tools\ant\build.xml:720: The following error occurred while executing this line:
C:\Users\hoffman\android-windows\adt-bundle-windows-x86_64-20130917\sdk\tools\ant\build.xml:734: Unable to find a javac compiler;
com.sun.tools.javac.Main is not on the classpath.
Perhaps JAVA_HOME does not point to the JDK.
It is currently set to "C:\Program Files (x86)\Java\jre7"

http://stackoverflow.com/questions/4613046/setting-java-home-at-android-sdk

- VTK needs to be fixed

$ git diff

```
diff --git a/CMake/GenerateExportHeader.cmake b/CMake/GenerateExportHeader.cmake
index e0897d4..3390887 100644
--- a/CMake/GenerateExportHeader.cmake
+++ b/CMake/GenerateExportHeader.cmake
@@ -170,7 +170,7 @@ macro(_test_compiler_hidden_visibility)
    # gcc on mac just reports: "gcc (GCC) 3.3 20030304 ..." without the
    # patch level, handle this here:
diff --git a/CMake/GenerateExportHeader.cmake b/CMake/GenerateExportHeader.cmake
index e0897d4..3390887 100644
--- a/CMake/GenerateExportHeader.cmake
+++ b/CMake/GenerateExportHeader.cmake
@@ -170,7 +170,7 @@ macro(_test_compiler_hidden_visibility)
    # gcc on mac just reports: "gcc (GCC) 3.3 20030304 ..." without the
    # patch level, handle this here:
    if(NOT _gcc_version)
-     string(REGEX REPLACE ".*\\(GCC\\).* ([34]\\.[0-9]) .*" "\\1.0"
+     string(REGEX REPLACE ".*\\(GCC\\).*([34]\\.[0-9]).*" "\\1.0"
        _gcc_version "${_gcc_version_info}")
    endif()

diff --git a/CMake/vtkCompilerExtras.cmake b/CMake/vtkCompilerExtras.cmake
index 0b57437..c9a59b4 100644
--- a/CMake/vtkCompilerExtras.cmake
+++ b/CMake/vtkCompilerExtras.cmake
@@ -27,16 +27,19 @@ if(CMAKE_COMPILER_IS_GNUCXX)
    _gcc_version_info)
  string (REGEX MATCH "[345]\\.[0-9]\\.[0-9]*"
    _gcc_version "${_gcc_version_info}")
+ message("[${_gcc_version}] = _gcc_version")
  if(NOT _gcc_version)
-   string (REGEX REPLACE ".*\\(GCC\\).* ([34]\\.[0-9]) .*" "\\1.0"
+   string (REGEX REPLACE ".*\\(GCC\\).*([34]\\.[0-9]).*" "\\1.0"
      _gcc_version "${_gcc_version_info}")
  endif()
+ message("[${_gcc_version_info}] = _gcc_version_info")
+ message("[${_gcc_version}] = _gcc_version")

  # GCC visibility support, on by default and in testing.
  check_cxx_compiler_flag(-fvisibility=hidden HAVE_GCC_VISIBILITY)
```

Tons of warnings

NEW tool chain fixes this.

https://raw.github.com/Itseez/opencv/master/platforms/android/android.toolchain.cmake

CMake Warning at C:/Users/hoffman/Work/VES/CMake/toolchains/android.toolchain.cmake:318 (message):
  Could not uniquely determine machine name for compiler from

c:/Users/hoffman/android-windows/android-ndk-r9/toolchains/x86-clang3.3/prebuilt/windows-x86_64.
Call Stack (most recent call first):
  C:/Users/hoffman/Work/VES/CMake/toolchains/android.toolchain.cmake:481 (__DETECT_TOOLCHAIN_MACHINE_NAME)
  C:/Users/hoffman/Work/VES/Apps/Android/Kiwi-b3/CMakeFiles/2.8.11/CMakeSystem.cmake:1 (include)
  CMakeLists.txt:2 (project)