

Building 3rd Party Libraries on VS2015

Assimp

Version: 3.3.1

Website: <http://www.assimp.org/>

Source: <https://github.com/assimp/assimp/releases/tag/v3.2/>

Build Method: CMake + VS2015

- 1) Assimp must be configured with CMake's GUI. Do not use Cygwin's packaged version of CMake.
- 2) Point the CMake GUI at the root source directory of Assimp. Configure and generate a project for Visual Studio 14 2015 Win64
- 3) Open the project in VS2015. Change the build type from Debug to Release and build the solution.
- 4) Copy {build dir}/bin/Release/assimp-vc140-mt.dll into vaporwin64deps/vapordlls
- 5) Copy the directory {build dir}/include/assimp into Vapor3rdParty2015/include

GRIB API

Version: 1.19.0

Website: <https://software.ecmwf.int/wiki/display/GRIB/Home>

Download: <https://software.ecmwf.int/wiki/display/GRIB/Releases>

- 1) Download and extract grib_api package into target build directory
- 2) Within the build directory, there are two directories called 'definitions' and 'samples'
- 3) Set environment variables as follows:
GRIB_DEFINITION_PATH={build dir}/definitions
GRIB_SAMPLES_PATH={build dir}/samples
- 4) Open the Visual Studio project in {build dir}/windows/msvc/grib_api.lib
- 5) Set the build to be 'Release' on 'x64' and build the project grib_api.lib (other projects may be build optionally)
- 6) Copy the library {build dir}/windows/msvc/grib_api.lib/x64/Release/grib_api.lib into Vapor3rdParty/releaseLib64
- 7) Copy only the header files from {build dir}/src into Vapor3rdParty/include/grib_api

To build with libjasper for jpeg2000 support:

Download cg-toolkit: <https://developer.nvidia.com/cg-toolkit-download>

Download libjasper: <https://software.ecmwf.int/wiki/download/attachments/48109524/jasper-1.900.1.zip?version=1&modificationDate=1438768621027&api=v2>

- 1) Install the cg-toolkit to install glut
- 2) Download the Jasper-1.900 zip file linked above and extract
- 3) Open the visual studio solution in the extracted directory and allow VS2015 to perform its conversion on the contained projects
- 4) Build the libjasper project, ignoring the others. You may close the jasper visual studio solution.
- 5) Open the grib_api solution in VS2015
- 6) Add the jasper library project to the grib_api solution by right clicking the grib_api solution (in the Solution Explorer) and then click Add->Add Existing Project
- 7) Select the libjasper project located at {jasper build dir}/src/msvc/libjasper.vcproj
- 8) Now open the grib_api.lib project and right-click on the References tree. Click 'Add Reference' and check the libjasper checkbox
- 9) Right click the grib_api.lib project and click 'Properties'.
- 10) Under Configuration Properties->C/C++->Preprocessor->Preprocessor Definitions, add the following text:
HAVE_JPEG;USE_JPEG2000;JAS_WIN_MSVC_BUILD;HAVE_LIBJASPER;
- 11) Build the grib_api.lib project with Release/x64 settings and continue at step 6 for building the grib_api.lib above

Freetype

Version: 2.7.1

Website: <https://www.freetype.org/>

Download: <http://download.savannah.gnu.org/releases/freetype/>

Build Method: Cygwin + VS2015

- 1) Download source and navigate to the freetype-2.6.5 directory in Cygwin
- 2) Type make, this will initialize a configuration routine
- 3) Type make again. This will build a Visual Studio project in freetype-2.7.1/builds/windows/vc2010
- 4) Open the Visual Studio project and allow VS2015 to upgrade the 2010 project to 2015 settings.
- 5) Build the solution configured for Release, on x64
- 6) Copy the directory freetype-2.7.1/include/freetype into Vapor3rdParty2015/include
- 7) Copy the file freetype-2.7.1/include/ft2build.h to Vapor3rdParty2015/include
- 8) Copy the library freetype-2.7.1/objs/vc2010/x64/freetype265 into Vapor3rdParty2015/releaseLib64

Qt

Version: 4.8.7

Website: <https://www.qt.io/>

Download: https://download.qt.io/official_releases/qt/4.8/4.8.7/

Build method: VS2015

Note: qt 4.8 and up requires perl

1) Set the following Environment Variables:

- QTDIR={build dir}/src
- QMAKESPEC=win32-msvc2015
- PATH=[PATH];qt_build/bin

2) Download source and extract into {build dir}

3) Apply the patch file (PatchForVS2015_Qt4.8.7) located at the bottom of this wiki page or in /glade/p/DASG/VAPOR/third-party/. This may need to be done manually due to differences in line endings on Unix and Windows.

4) Visual Studio comes packaged with special command line interfaces that can only seem to be launched from the start menu for some ungodly reason.

Launch the "VS2015 x64 Native Tools Command Prompt" as found in the screenshot below.

5) Navigate to {build dir} and enter the following:

configure -make nmake -platform win32-msvc2015 -prefix c:\Qt-2015\4.8.6\msvc2015 -opensource -opengl desktop -nomake examples -nomake tests

6) Then launch the command 'nmake'

jpeg

Version: 9b

Website: <http://www.ijg.org/>

Download: <http://www.ijg.org/files/jpegsr9b.zip>

Build Method:

1) cd to the location of the extracted zip file

2) Open a "Developer Command Prompt for VS2015" with admin privileges (note that this is different from a normal dos command prompt)

3) Run the following command: NMAKE /f [makefile.vc](#) setup-v10

4) If an error message comes up that states 'win32.mak' could not be found, do a search for that file on your local machine and copy it into the current directory and rerun the command in step #3.

5) Open the file jpeg.sln in VS2015 and build it.

6) If you get an error indicating that jpeg.sln could not be loaded, make sure that you've downloaded the jpeg9b.zip file and not the .tar.gz file. This error went away for me when following these instructions regarding the .zip file.

CMake

Version: 3.6.1

Website: <https://cmake.org/>

Binary Distribution: <https://cmake.org/download/>

Notes:

Install this first as it will be used for building many of the libraries.

When building with CMake, unless otherwise specified:

- Use CMake-Gui (Do not use cygwin cmake)
- Use the same source and build directory
- Select "Visual Studio 14 2015 Win64" as the generator for this project
- After pressing [Generate] once, build options will be generated and can be configured
- Press [Generate] again after setting options
- Open the build folder and open <Library Name>.sln
- Make sure the build target is Release x64
- Right click on the solution and [Build Solution].

HDF5

- Requires zlib support
- While HDF5 builds properly, the output cmake configuration files do not. These files are required for building NetCDF with cmake.
- HDF5 provides an installer for the library built with VS 2015 that includes the correct cmake files in [install_dir]/cmake

NetCDF

- HDF5 cmake directory needs to be manually specified ([hdf5_install_dir]/cmake)
- zlib library and include dir needs to be manually specified. This is under the hdf5 install directory
- The configuration options for these two libraries can be found under Advanced Cmake Variables
- Disable building tests (missing files causes build to fail)
- Use latest release version from GitHub. Release version from website is missing files (known bug) and the latest github pull is not building properly either.

Other Notes

- expat needs to be added as an official dependency
- python2.7.lib was copied from binary install however the configuration currently used by vapor is odd so I don't know if this is necessary.
- proj_i.lib was renamed to proj.lib under target lib dependencies.
- different libraries are either built as dynamic or static so I just followed the current setup but this should be looked at.
- some of the libraries build multiple targets.
- libgeotiff, for example, outputs a .lib in bin/release/libgeo.lib as well as lib/release/libgeo.lib. Only the library in the bin subdirectory works as the other one is missing 2 symbols.



PatchForVS2015_Qt4.8.7.diff