PointoolsIO Supported File Formats

# List

## Open source libraries

* [LAS](#_LAS_1)
* [LAZ](#_LAZ)
* [ASTM E57](#_ASTM_E57)
* [Terrascan BIN](#_File_format_importers)
* [Leica PTG](#_LAS)
* [Leica PTS](#_LAS)
* [Leica PTX](#_LAS)
* [Riegl RDB](#_LAS)
* [Ascii XYZ](#_LAS)
* [Pointools POD](#_LAS)

## Libraries we can’t rebuild

* Topcon CL3 (TopconCodec.dll)
* Faro FLS (COM dll; ???)
* Faro FWS (COM dll; ???)
* Leica PTZ (ptreader80.dll)
* Riegl RSP (???)
* Riegl RXP (scanifc-mt.dll, rivlib)
* Riegl 3DD (COM dll; scannermod.dll, riscanlib)
* Z+F ZFS (zfs.dll)

# How to build third party libraries for Vortex

## File format importers

Vortex already contains c++ importers to support these file formats.

## LAS

### Prerequisites

Building LAS requires the [boost](http://www.boost.org/) libraries, more specifically, you will need to build the following boost libraries:

* Program\_options
* Thread
* System
* Iostreams
* Filesystem

Vortex also requires the built LAS to have GDAL, [LASZip](https://github.com/LASzip/LASzip) and [GEOTiff](http://trac.osgeo.org/geotiff/) support. Image++ has a version of GDAL which can be used to build LAS. Finally, to build LAS with GEOTiff support, you will also need the [TIFF](http://www.libtiff.org/) library.

### Get LAS sources

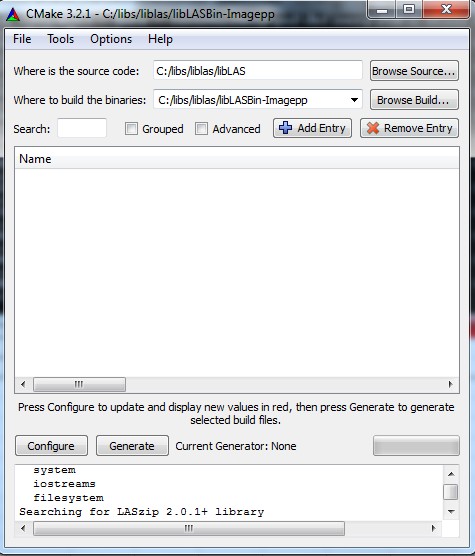
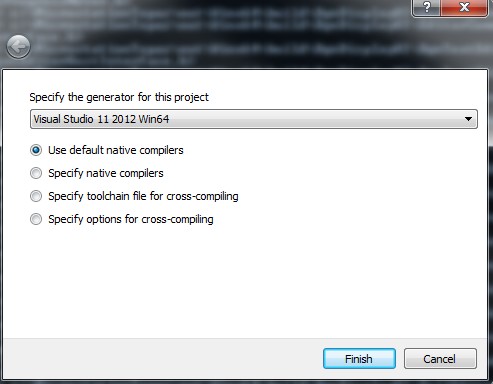
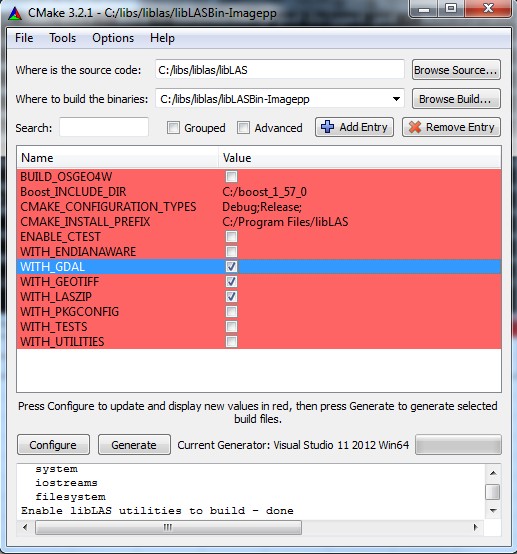
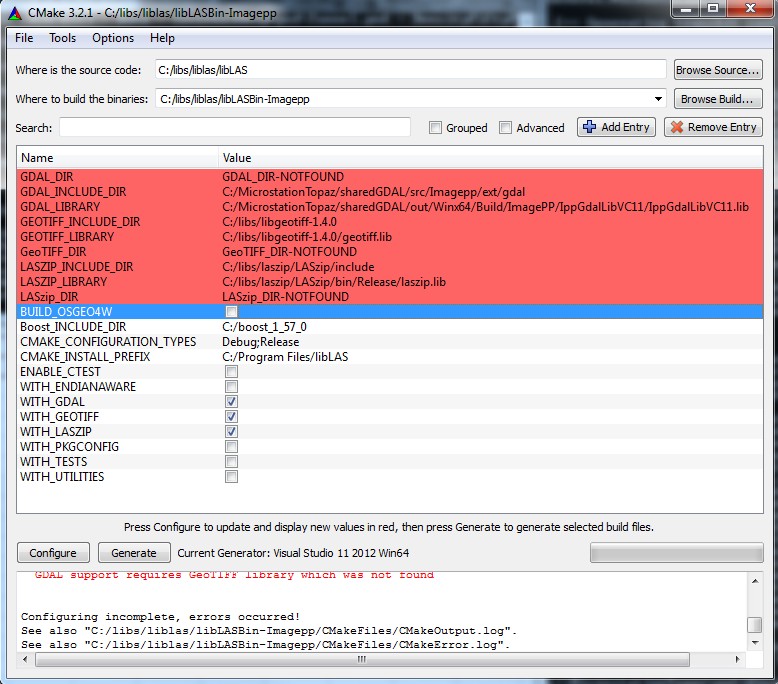
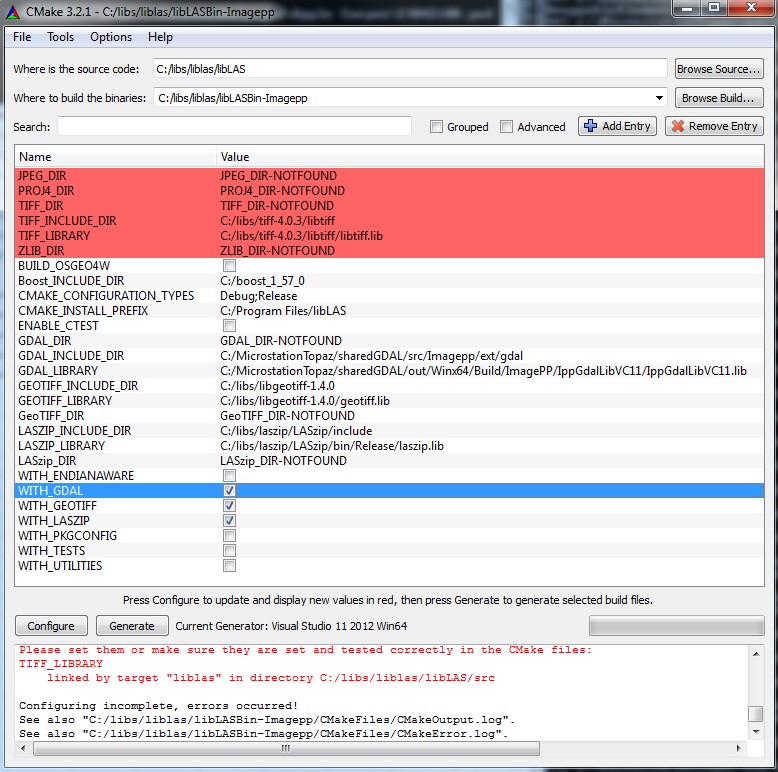
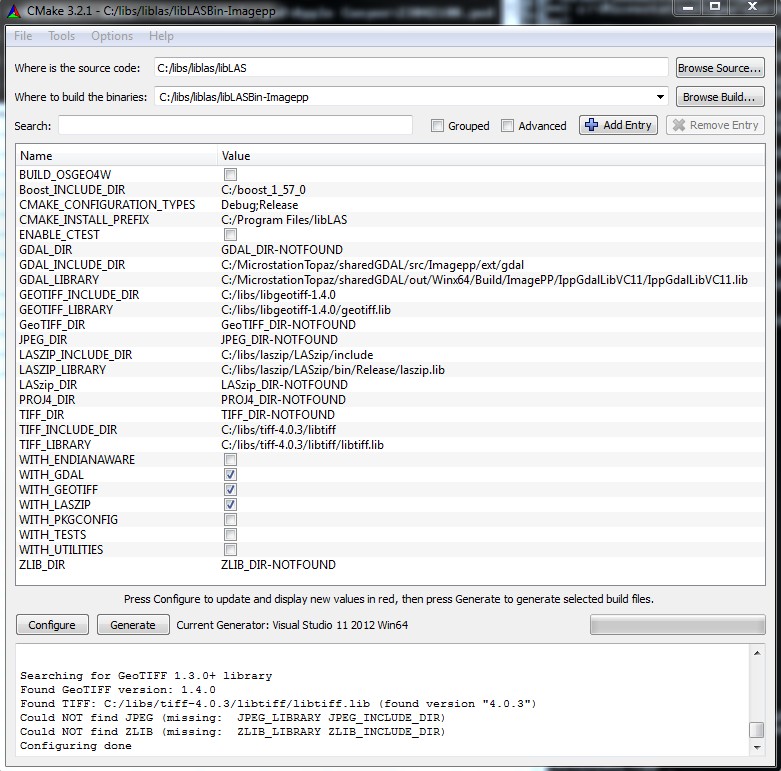
Currently, this library is statically built. In order to do so, you must download the source code here: <http://www.liblas.org/download.html>

Or clone the source trunk from github (you can use [Git Bash](https://msysgit.github.io/) for this):

git clone <https://github.com/libLAS/libLAS.git>

### Configure LAS

Once you have obtained the code, you need to configure the build using [cmake](http://www.cmake.org/download/). This is how you setup LAS to build with GDAL, GEOTiff and LASZip support.

1. Open cmake and set source code path and the path to build binaries.  
     
   
2. Click on the Configure button.
3. Select your compiler version with corresponding cpu architecture. As of writing this document only Visual Studio 2005 and Visual Studio 2012 can be used to build Vortex.   
     
   ––––
4. Set the configuration variables and carefully set the boost include directory.   
     
   
5. Click on the Configure button again.
6. New variables appear in red; set the variables according to your paths.   
     
   
7. Click on the Configure button again.
8. New variables appear in red; set the variables according to your paths. JPEG, PROJ4 and ZLIB can be ignored.   
     
   
9. Click on the Configure button one final time.
10. Configure is complete.   
      
    

## LAZ

## ASTM E57

To build the libraries for the E57 file format, you will need to download and build the [Apache Xerces C++ 3.1.1 library](http://archive.apache.org/dist/xerces/c/3/sources/xerces-c-3.1.1.zip) (note that this is not the latest version which as of writing this document is 3.1.2 but the latest version has not been tested with Vortex). Building Xerces should be straightforward, simply open the VC11 solution, set the desired platform and build.

The other dependencies are the Boost libraries. The following boost libraries will need to be built:

* filesystem
* program\_options
* system
* thread

In order for CMake to find these libraries, they must all be in a special directory structure. Here is the recommended directory structure for boost libraries:

BOOST\_ROOT

* boost
* lib

where boost directory contains the header files and lib directory contains \*.lib files.

Next we will use CMake again to build ASTM E57. Simply open CMake, indicate the path to E57 source directory and set the desired output directory (it is recommended to set this directory to a different folder than where the source code has been saved).

Then click on the Configure button. CMake will now ask you to set BOOST\_ROOT which should point to a boost folder containing the lib directory where all the libraries are saved and the boost directory where all the headers are saved. Click on the Configure button again and CMake will ask you to set the XERCES\_ROOT variable which points to where Xerces was built. This also needs to be in a special tree structure otherwise CMake will not be able to find Xerces:

XERCES\_ROOT:

* include\xercesc
* lib

where include\xercesc directory contains the Xerces header files and lib directory contains \*.lib files.

Click on the Configure button one last time should succeed. Finally, click on the Generate button to generate the solution projects which can be used to build E57RefImpl and time\_conversion necessary to build Vortex.