libLAS

LAS 1.0/1.1 ASPRS LiDAR data translation toolset

http://liblas.org

What is libLAS?

- BSD-licensed C++ library for reading ASPRS .LAS data
- Command-line utilities for inspecting and manipulating .LAS files
- Python/C/.NET APIs for manipulating .LAS files on your own

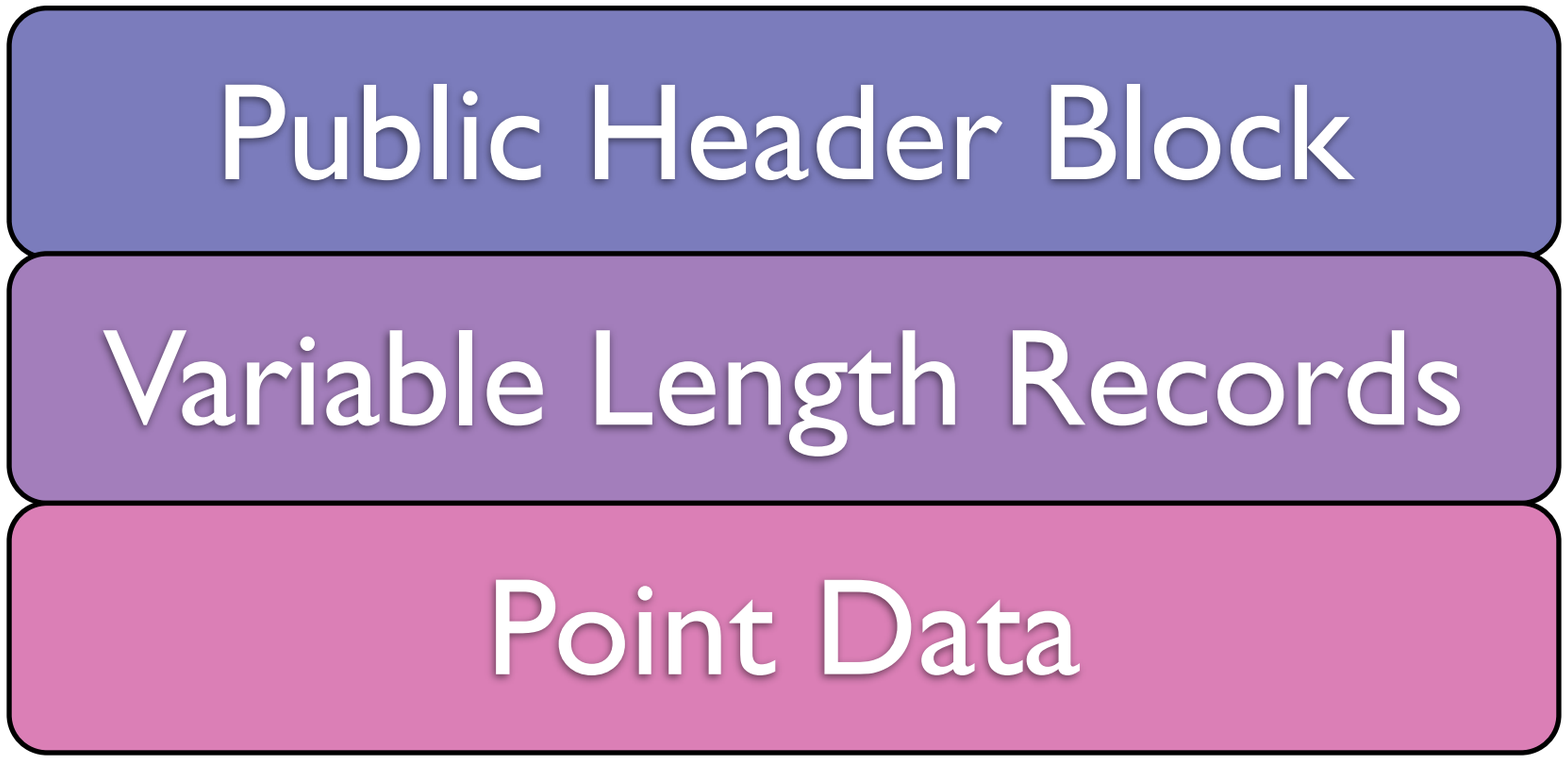
What is libLAS?

- GeoTIFF SRS support with libgeotiff
- Variable Length Record (VLR) abstraction
- Cross-platform (Windows/OS X/Linux/ Solaris)

What is LAS?

- A binary LiDAR data exchange format specification managed by ASPRS
- Little-endian
- 1.0, 1.1, 1.2, and 2.0 versions

LAS File Organization



Public Header Block

- Version information
- File identification and description
- Point layout description
- Summary information

Variable Length Records

- Any old blob of binary stuff you want
- 54 byte header for each VLR that describes what might be there
- Application must know how to interpret a VLR

Variable Length Records

- Uses for VLRs:
  - Application-specific spatial indexing
  - Classification information (pre-1.1 LAS data)
  - Spatial referencing (GeoTIFF)

Point Data

- Coordinates
- Pulse return information
- Laser angle and direction
- Classification (1.1)
- epoch time (depends on format type)

Library Design

- Application of the “Bridge Pattern” - PIMPL
- Heavy use of STL
- RAII - Resource Acquisition is Initialization

Bridge Pattern

- Separate the interface from the implementation
    - Allows extension of libLAS to new versions of the format without materially touching the interface
- Reduces recompilation

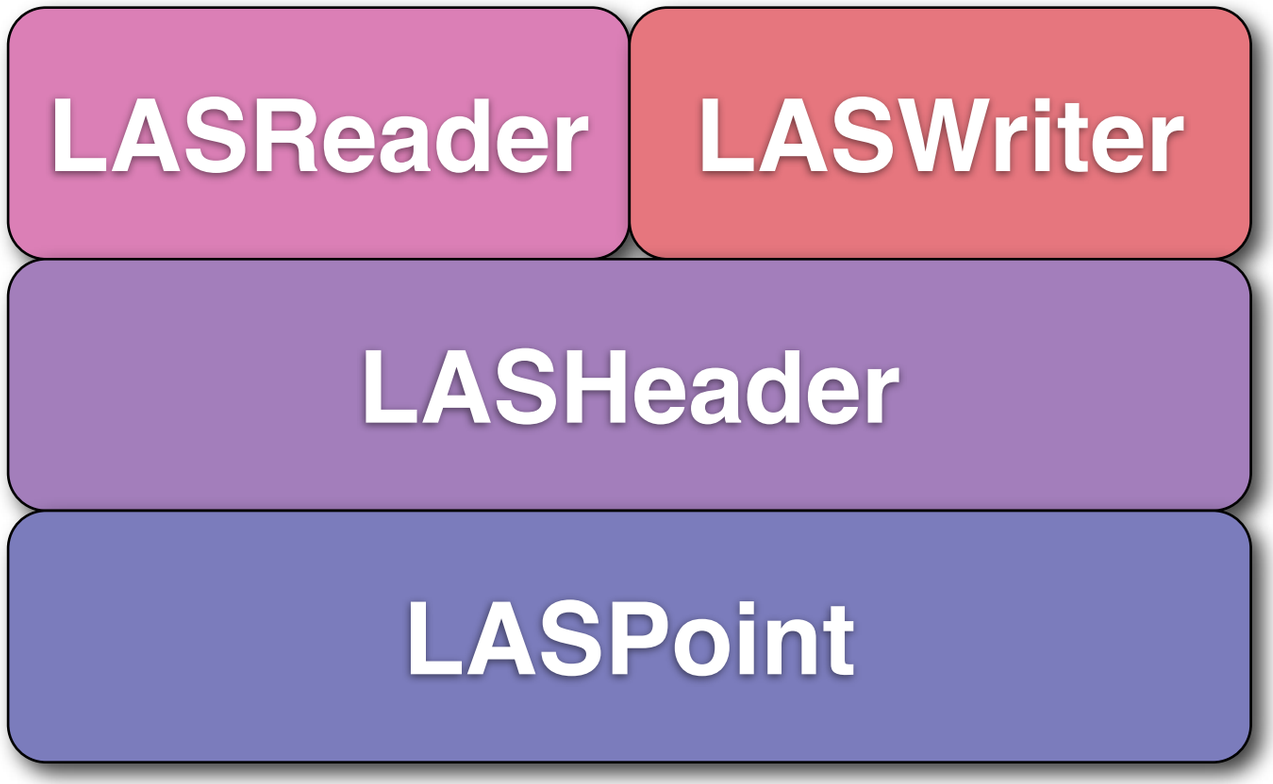
Healthy Dose of STL

- STL-compatible input/output iterators
- Standard elements like buffered I/O streams, strings, and containers
- Usability with standard algorithms like std::copy

RAII

- RAII is an important part of ensuring exception safety
- Helps prevent leaks
- Reduces worries about variable scope

Base C++ Classes

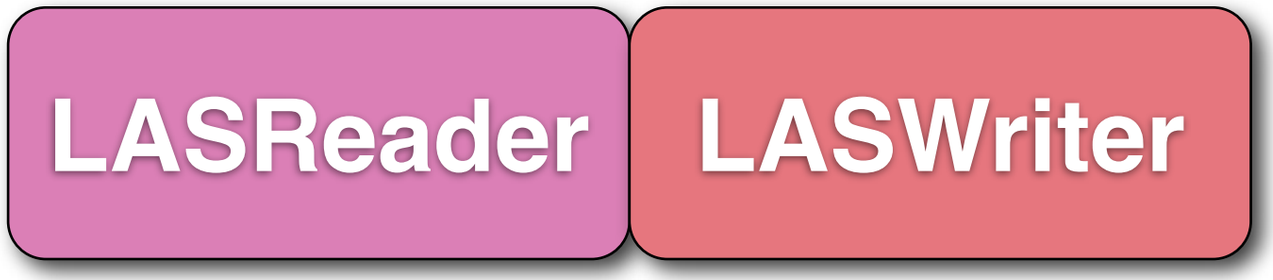


LASPoint

- Provides some convenience operators ([], ==)
- Provides some validation (Validate() and IsValid() methods)
- Some complain that it is too heavy and a naked struct would be sufficient
- Implements Point Data

LASHeader

- Provides context for the concept of a LAS *file* -- LASPoints can stand on their own, but they are not read/written to a file without LASHeader context
- Specification provides for scaling/offsetting coordinates and multiple point record types
- Implements the Public Header Block



- Connect LASHeader and LASPoint to a physical file (actually a stream)
- Apply the scaling when reading/writing points as described in the LASHeader
- Hide reading/writing details

Other APIs

- C
- Python
- .NET

libLAS C API

- Similar to GDAL/OGR C API
- Allows for binary compatibility and helps prevent cross-dll troubles
- Tedious and verbose, but it provides some extra logic for working with libLAS C++ classes

libLAS Python API

- Uses ctypes and libLAS C API
- Provides even more sugar
- Good doctest coverage demonstrating usage
- Installable via PyPI

libLAS .NET API

- Contributed effort by Martin Vales
- Built atop libLAS C API using Pinvoke
- Hope to get it in libLAS 1.0 release
- Building/deployment needs some work

libLAS Command-line Utilities

- Port of Martin Isenburg's LASTools to use libLAS
- Provide inspection, conversion, pruning, processing, and merging

lasinfo

- Similar to gdalinfo, ogrinfo, tiffinfo
- Provides header inspection and point summary

las2las

- Simple LAS processing
- Clip by bounding box, cull returns of a certain classification, cull invalid points, convert between LAS versions (1.0 <-> 1.1)

las2txt

- Dump LAS file to a character-separated file with a given format

txt2las

- Take in a text file with a given format and convert to a LAS file

lasmerge

- Merge multiple LAS files into a single file

liblas.org

- Howard Butler
- Mateusz Loskot
- Martin Vales
- Phillip Vachon
- Frank Warmerdam

Future

- Looking for support to implement LAS 1.2
- Maybe do LAS 2.0 support if a fire hose of money shows up :)
- Optional Boost types integration (array, iterators)
- Policy-based reader/writer for C++

Future, cont.

- Hope to have a GDAL/OGR driver someday
- GRASS could use libLAS for multiplatform LAS format support
- More language bindings

Thanks

- Iowa Department of Natural Resources
- Martin Isenburg of UNC