Esri's Optimized LAS I/O Library and API

Clayton Crawford

Contents

EzLasLib - What It Is

- C++ library for reading and writing Esri Optimized LAS (zLAS)
  - Convert between zLAS and LAS
  - Optimized queries on zLAS
- Easily incorporated into existing systems that read/write LAS
- Includes support through LAS 1.4, all point record formats
- Compatible with Visual Studio 2008 and higher
- Compatible with Windows XP sp3 and higher
- 32 and 64 bit versions
- Linux release planned for later in the year

zLAS Recap

- ASPRS LAS is the standard
- Esri fully supports the standard
- The standard was designed for data exchange, not direct use
- zLAS allows us to turn LAS into something easy and efficient to use
  - Compression
  - Indexing
  - Statistics
  - Lossless
- Free EzLas Optimizer utility app available to convert between zLAS/LAS

Library Constructs

Interfaces

iEzLasObject
iEzLasReader
iEzLasWriter
iEzLasQueryFilter
iEzLasLongArray
iEzLasLong64Array
iEzLasMemoryBuffer
iEzLasEnumPoint
iEzLasFeedback

Classes/Objects

EzLasObjectFactory
-> Reader
-> Filter
-> Writer

Structs

EzLasHeader
EzLasPointInfo
EzLasRGB
EzLasPoint2D
EzLasPoint3D
EzLasExtent3D
EzLasPointSpacingOption

Enums

EzLasPointPropertyType
EzLasColorChannelType
EzLasPointFlagType
EzLasPointReturnType
EzLasErrorCodes

The EzLAS Reader

- Read as LAS
  - Records loaded into memory buffers in standard LAS format
- Read into structs/arrays as an alternative to LAS format buffers
- Supports query filter
  - AOI envelopes
  - Returns, class codes, flags
  - Enumerators return query results
- Supports point data mask
  - Decompress only what’s needed
- Includes statistics that go beyond standard LAS

The EzLAS Writer

- Optimize/compress LAS file into zLAS
- Decompress zLAS file into LAS
- File level granularity, operates on entire files

Parallel Processing

- Both Reader and Writer Objects are multi-threaded
- Significant performance gain utilizing multiple cores
- Can control amount of CPU resources allocated

Error Handling and Reporting

- All methods return success or failure codes
- Primary objects (reader/writer) provide access to descriptive error string (english)
  - Non-english apps should use error codes to map to their own strings
- A validator method is included for checking integrity of input zLAS

Progress Reporting and Canceling

- Implement iEzLasFeedback
- Sets up callbacks:
  - ‘SetProcessCompletionPercentage’
    - for progress reporting
  - ‘Continue’
    - to support cancel in the middle of a process

Spatial Reference

- Coordinate system records are retrieved from zLAS via memory buffers
- Buffers are written to in LAS format so any existing app code used to read this info from LAS can be used to read from zLAS
- Does not provide means of interpreting these records

Extra Bytes

- LAS 1.4 construct for extensibility
  - E.g., Topo-bathy domain profile
- Can contain different payloads
- Accessed via memory buffers
- Your code has to interpret

Using the Library in a Project

- Include reference to EzLasAPIs.h
- Methods/properties documented in .h
- Add EzLasLib folder to VS project library paths
- Include EzLasLib dll in exe folder

Demo

Availability

- Library is final and available for download
- GitHub -> https://github.com/Esri/esri-zlas-io-library
- Apache 2.0 license (allows for use and redistribution without cost)
- Sample source code for this demo
  - Request via email: ccrawford@esri.com

Questions?

Rate This Session

www.esri.com/RateMyDevSummitSession