LAS
Specification
Version 1.0
January 15, 2002

LAS FORMAT DEFINITION:

The LAS file is intended to contain LIDAR point data records. The data will generally be put into this format from software (provided by LIDAR hardware vendors) which combines GPS, IMU, and laser pulse range data to produce X, Y, and Z point data. The intention of the data format is to provide an open format which allows different LIDAR vendors to output data into a format which a variety of LIDAR software vendors can use. Software that creates the LAS file will be referred to as "generating software", and software that reads and writes to the LAS file will be referred to as "user software" within this specification.

The format contains binary data consisting of a header block, variable length records, and point data. All data is in little-endian format. The header block consists of a public block followed by variable length records. The public block contains generic data such as point numbers and coordinate bounds. The variable length records contain variable types of data including projection information, metadata, and user application data.

PUBLIC HEADER BLOCK
VARIABLE LENGTH RECORDS
POINT DATA

DATA TYPES:

char (1 byte)
unsigned char (1 byte)
short (2 bytes)
unsigned short (2 bytes)
long (4 bytes)
unsigned long (4 bytes)
double (8 byte IEEE floating point format)

PUBLIC HEADER BLOCK:

Item	Format	Size	Required
File Signature ("LASF")	char[4]	4 bytes	*
Reserved	unsigned long	4 bytes	
GUID data 1	unsigned long	4 bytes	
GUID data 2	unsigned short	2 byte	
GUID data 3	unsigned short	2 byte	
GUID data 4	unsigned char[8]	8 bytes	
Version Major	unsigned char	1 byte	*
Version Minor	unsigned char	1 byte	*
System Identifier	char[32]	32 bytes	*
Generating Software	char[32]	32 bytes	*
Flight Date Julian	unsigned short	2 bytes	
Year	unsigned short	2 bytes	
Header Size	unsigned short	2 bytes	*
Offset to data	unsigned long	4 bytes	*
Number of variable length records	unsigned long	4 bytes	*
Point Data Format ID (0-99 for spec)	unsigned char	1 byte	*
Point Data Record Length	unsigned short	2 bytes	*
Number of point records	unsigned long	4 bytes	*
Number of points by return	unsigned long[5]	20 bytes	*
X scale factor	double	8 bytes	*
Y scale factor	double	8 bytes	*

Z scale factor	double	8 bytes	*
X offset	double	8 bytes	*
Y offset	double	8 bytes	*
Z offset	double	8 bytes	*
Max X	double	8 bytes	*
Min X	double	8 bytes	*
Max Y	double	8 bytes	*
Min Y	double	8 bytes	*
Max Z	double	8 bytes	*
Min Z	double	8 bytes	*

All data that is not required and not filled with data must be set to zeros.

File Signature: The file signature must contain the four characters "LASF", and it is required by the LAS specification. These four characters should be checked by user software to determine valid LAS files.

Reserved: This data field is reserved, and must contain zeros unless otherwise specified by subsequent LAS format specifications.

GUID data: The Globally Unique Identifier (GUID) data fields are not required by the LAS specification (GUID data 1, GUID data 2, GUID data 3, and GUID data 4). The GUID data fields are intended to provide space for uniquely identifying each file created. The GUIDs provide a method of tracking individual files regardless of file names or directory structure.

Version Number: The version number is required by the LAS specification. The version number consists of a major and minor field. The major and minor fields combine to form the number that indicates the format number of the current specification itself. For example, specification number 1.0 would be 1 in the major field and 0 in the minor field.

System ID: The system identifier field is populated by the generating software application. This field provides a mechanism for specifying the actual hardware system which gathered the LIDAR data. The system identifier itself should be provided by the hardware system vendor and populated by the generating software. If the character data is less than 32 characters, the remaining data must be null.

Generating Software: The "generating software" field is also populated by the generating software. This information is ASCII data describing the generating software itself. This field provides a mechanism for specifying which generating software package and version was used during LAS file creation. If the character data is less than 16 characters, the remaining data must be null.

Flight Date Julian: The julian day of the year that the data was collected. This field should be populated by the generating software.

Year: The year in which the data was collected. This field should be populated by the generating software.

Header Size: The header size field is the actual header block itself. This number is required, and must be provided by the generating software. All software should read this header size number, and regard that number as the header size in bytes. In the event that data is added to the header, the header size field must be updated with the new header size. Generating software is

the only type of software that is allowed by the specification to add data to the general header. The variable length records should be used whenever possible to add data to the header, and not to extend the general header. However, this specification allows for a general header to be larger than the specified data structure. Therefore, the header size field must at all times contain the exact size of the header itself. In the event a generating software package adds data to the general header, this data must be placed at the end of the specified general header structure.

Offset to data: The offset to data field is the actual number of bytes from the beginning of the file to the data itself including the two bytes for the data start signature. This data offset must be updated if any software adds data from the general header or adds/removes data from the variable length records. It is recommended that additional space be allocated to the variable length header space, so that additional variable length data can be added after the initial data is written. This space prevents the need to rewrite the file every time a variable length record is added.

Number of variable length records: This field contains the current number of variable length records. This number must be updated if the number of variable length records changes at any time.

Point Data Format ID: The point data format ID corresponds to the point data record format type.

Number of point records: This field contains the total number of point records within the file.

Number of points by return: This field contains an array of the total point records per return. The first unsigned long value will be the total number of records from the first return, and the second contains the total number for return two, and so forth up to five returns.

X, Y, and Z scale factors: The scale factor fields contain a double floating point value that is used to scale the corresponding X, Y, and Z long values within the point records. The corresponding X, Y, and Z scale factor must be multiplied by the X, Y, or Z point record value to get the actual X, Y, or Z coordinate. For example, if the X, Y, and Z coordinates are intended to have two decimal point values, then each scale factor will contain the number 0.01.

X, Y, and Z offset: The offset fields should be used to set the overall offset for the point records. In general these numbers will be zero, but for certain cases the resolution of the point data may not be large enough for a given projection system. However, it should always be assumed that these numbers are used. So to scale a given X from the point record, take the point record X multiplied by the X scale factor, and then add the X offset.

$X_{coordinate} = (X_{record} * X_{scale)} + X_{offset}$
$Y_{coordinate} = (Y_{record} * Y_{scale)} + Y_{offset}$
$Z_{coordinate} = (Z_{record} * Z_{scale)} + Z_{offset}$

Max and Min X, Y, Z: The max and min data fields are the actual file coordinate extents of the LAS point file data.

The projection information for the point data is required for all data. The projection information will be placed in the variable length records. Placing the projection information within the variable length records allows for any projection to be defined including custom projections. The GeoTiff specification http://www.remotesensing.org/geotiff/geotiff.html is the model for representing the projection information, and the format will be explicitly defined by this specification.

VARIABLE LENGTH RECORDS:

Item	Format	Size	Required
Record Signature (0xAABB)	unsigned short	2 bytes	*
User ID	char[16]	16 bytes	*
Record ID	unsigned short	2 bytes	*
Record Length After Header	unsigned short	2 bytes	*
Description	char[32]	32 bytes	

Record Signature: The record signature is a two byte data field that must contain 0xAABB. Note that there may be other data within the file that has 0xAABB as its data.

User ID: The user ID field is ASCII character data that identifies the user which created the variable length record. It is possible to many variable length records from different sources with different user Ids. If the character data is less than 16 characters, the remaining data must be null. The user ID must be registered with the LAS specification managing body. The management of these Ids insures that no two individuals accidentally use the same ID. The specification will initially use two Ids. One for globally specified records (LASF_Spec), and another for projection types (LASF_Projection).

Record ID: The record ID is dependent upon the User ID. There can be 0 to 65535 record Ids for every User ID. The LAS specification will manage its own record Ids (User Ids owned by the specification), otherwise record Ids will be managed by the owner of the given User ID. So each User ID is allowed to assign 0 to 65535 record Ids as they wish. Publicizing the meaning of a given record ID will be left to the owner of the given User ID. Unknown User ID/Record ID combinations should be ignored.

Record Length after Header: The record length is the number of bytes for the record after the end of the standard part of the header.

Description: Optional null terminated text description of the data. Any remaining characters not used must be null.

Point Data Start Signature:
Two bytes after the last variable length record, and before the point data
0xCCDD

POINT DATA RECORD FORMAT 0:

Item	Format	Size	Required
X	long	4 bytes	*
Y	long	4 bytes	*
Z	long	4 bytes	*
Intensity	unsigned short	2 bytes	
Return Number	3 bits	3 bits	*
Number of Returns (given pulse)	3 bits	3 bits	*
Scan Direction Flag	1 bit	1 bit	*
Edge of Flight Line	1 bit	1 bit	*
Classification	unsigned char	1 byte	
Scan Angle Rank (-90 to +90) – Left side	char	1 byte	*
File Marker	unsigned char	1 byte	
User Bit Field	unsigned short	2 bytes	

X, Y, and Z: The X, Y, and Z values are stored as long integers. The corresponding X scale, Y scale, and Z scale values from the public header block change these long integers to their true floating point values. The corresponding offset values can also be used for projections with very large numbers.

Intensity: The intensity value is the integer representation of the pulse return magnitude. This value is optional and system specific.

Return Number: The return number is the pulse return number for a given output pulse. A given output laser pulse can have many returns, and they must be marked in sequence of return. The first return will have a return number of one, the second a return number of two, and so on up to five returns.

Number of Returns (given pulse): The number of returns is the total number of returns for a given pulse. So a laser data point may be return two (return number) with a total number of five returns.

Scan Direction Flag: The scan direction flag denotes the direction at which the scanner mirror was traveling at the time of the output pulse. A bit value of 1 is a positive scan direction, and a bit value of 0 is a negative scan direction.

Edge of Flight Line: The edge of flight line data bit has a value of 1 only when the point is at the end of a scan. It is the last point on a given scan line before it changes direction.

Classification: The classification field is a number to signify a given classification during filter processing. The LAS specification has a public list of classifications which shall be used when mixing vendor specific user software.

Scan Angle Rank: The scan angle rank is a signed one-byte number. The scan angle rank is the angle at which the laser point was output from the laser system including the roll of the aircraft. The scan angle is within 1 degree of accuracy from +90 to –90 degrees. The scan angle is an angle based on 0 degrees being NADIR, and –90 degrees to the left side of the aircraft in the direction of flight.

File Marker: The file marker is an optional field that should be used in conjunction with the variable length records. The file marker allows for the LAS flight-line based files to be combined into single files with more than one flight-line. The actual original flight-line should be tracked in the variable length records, so that individual flight lines can be separated from a given combined file.

User Bit Field: A bit field that is to be used at the users discretion.

Other point data formats must be derived from the "Point Data 0" structure with the additional data added thereafter.

POINT DATA RECORD FORMAT 1:

Item	Format	Size	Required
X	long	4 bytes	*
Y	long	4 bytes	*
Z	long	4 bytes	*
Intensity	unsigned short	2 bytes	
Return Number	3 bits	3 bits	*
Number of Returns (given pulse)	3 bits	3 bits	*

Scan Direction Flag	1 bit	1 bit	*
Edge of Flight Line	1 bit	1 bit	*
Classification	unsigned char	1 byte	
Scan Angle Rank (-90 to +90) – Left side	unsigned char	1 byte	*
File Marker	unsigned char	1 byte	
User Bit Field	unsigned short	2 bytes	
GPS Time	double	8 bytes	*

GPS Time: The GPS time is the double floating point time tag value at which the point was acquired.

DEFINED VARIABLE LENGTH RECORDS: Georeferencing Information

Georeferencing for the LAS format will use the same robust mechanism that was developed for the GeoTIFF standard. The variable length header records section will contain the same data that would be contained in the GeoTIFF key tags of a TIFF file. With this approach, any vendor that has existing code to interpret the coordinate system information from GeoTIFF tags can simply feed the software with the information taken from the LAS file header. Since LAS is not a raster format and each point contains its own absolute location information, only 3 of the 6 GeoTIFF tags are necessary. The ModelTiePointTag (33922), ModelPixelScaleTag (33550), and ModelTransformationTag (34264) records can be excluded. The GeoKeyDirectoryTag (34735), GeoDoubleParamsTag (34736), and GeoASCIIParamsTag (34737) records will be used.

Only the GeoKeyDirectoryTag record is required. The GeoDoubleParamsTag and GeoASCIIParamsTag records may or may not be present, depending on the content of the GeoKeyDirectoryTag record.

GeoKeyDirectoryTag Record
User ID: LASF_Projection
Record ID: 34735

This record contains the key values that define the coordinate system. A complete description can be found in the GeoTIFF format specification. Here is a summary from a programmatic point of view for someone interested in implemention.

The GeoKeyDirectoryTag is defined as just an array of unsigned short values. But, programmatically, the data can be seen as something like this:

```
struct sGeoKeys
{
   unsigned short wKeyDirectoryVersion;
   unsigned short wKeyRevision;
   unsigned short wMinorRevision;
   unsigned short wNumberOfKeys;
   struct sKeyEntry
   {
      unsigned short wKeyID;
      unsigned short wTIFFTagLocation;
      unsigned short wCount;
      unsigned short wValue_Offset;
   } pKey[1];
```

};

Where:
wKeyDirectoryVersion = 1; // Always
wKeyRevision = 1; // Always
wMinorRevision = 0; // Always
wNumberOfKeys // Number of sets of 4 unsigned shorts to follow

For each set of 4 unsigned shorts:

wKeyID	Defined key ID for each piece of GeoTIFF data. IDs contained in the GeoTIFF specification.
wTIFFTagLocation	Indicates where the data for this key is located: 0 means data is in the wValue_Offset field as an unsigned short 34736 means the data is located at index wValue_Offset of the GeoDoubleParamsTag record. 34767 means the data is located at index wValue_Offset of the GeoAsciiParamsTag record.
wCount	Number of characters in string for values of GeoAsciiParamsTag , otherwise is 1
wValue_Offset	Contents vary depending on value for wTIFFTagLocation above

GeoDoubleParamsTag Record
User ID: LASF_Projection
Record ID: 34736
This record is simply an array of doubles that contain values referenced by tag sets in the GeoKeyDirectoryTag record.

GeoASCIIParamsTag Record
User ID: LASF_Projection
Record ID: 34737
This record is simply an array of ascii data. It contains many strings separated by null terminator characters which are referenced by position from data in the GeoKeyDirectoryTag record.

Classification lookup
User ID: LASF_Spec
Record ID: 0
Length: 256 recs X 16 byte struct len
struct CLASSIFICATION
{
    unsigned char ClassNumber;
    char Description[15];
};

Header lookup for flight-lines
User ID: LASF_Spec
Record ID: 1
Length: 256 recs X struct len

```
Struct FLIGHTLINE
{
        unsigned char FileMarkerNumber;
        char Filename[256];
};
```

Histogram
User ID: LASF_Spec
Record ID: 2

Text area description
User ID: LASF_Spec
Record ID: 3