# LAS Specification Version 2.0 PROPOSAL

May 1, 2007

***Comments/Corrections – Contact:***

Lewis Graham
GeoCue Corporation
9668 Madison Blvd., Suite 101
Madison, AL 35758
01-256-461-8289 (voice)
01-256-461-8249 (FAX)
lgraham@geocue.com

TABLE OF CONTENTS

# 1 LAS FORMAT VERSION 2.0

This document reflects the first major revision of the LAS format specification since its initial version 1.0 release (Release 1.1, a minor update to LAS 1.0, is the current release level). LAS 1.1 file Input/Output (I/O) libraries will require significant modification in order to be compliant with this revision.

A detailed change document that provides both an overview of the changes in the specification as well as the motivation behind each change is available from the ASPRS website in the LIDAR committee section.

Since the change from 1.1 to 2.0 is so extensive, a modification document will not be issued. Thus this document presents the 2.0 specification without a change reference to the 1.1 specification.

The major new features of the 2.0 specification are the infinitely variable point data records and extending the format to include terrestrial scanners and, indeed, general point cloud three dimensional data.

The Point data records are now key word encoded and contain a set of mandatory fields. To these mandatory fields, users can add ASPRS defined optional fields as well as their own user defined fields. This solves the significant problem in the 1.x specification of requiring a new record type (that then became proprietary) each time a new field needed to be defined.

# 2 LAS FORMAT DEFINITION

The LAS file is intended to contain three dimensional point cloud data records. The data will generally be put into this format from software provided by sensor hardware vendors or point cloud processing software. In the case of hardware, laser pulse range data are combined with sensor orientation information to produce geocoded X, Y, and Z point data.

Version 2.0 of the LAS specification extends the idea of using the format for LIDAR data to the general case of N dimensional point cloud data. The intention of the specification is to provide an open format that allows different hardware and software tools vendors to use a common data format.

An LAS file contains binary data consisting of a file header block, a Point Record Metadata Block (which is variable length), variable length records and finally the point data.

| |
|---|
| FILE HEADER BLOCK |
| POINT RECORD METADATA BLOCK |
| VARIABLE LENGTH RECORDS |
| POINT DATA |

All data are in little-endian format. The header section consists of a standard file header block followed by variable length records. The file header block contains generic data such as point source identification and coordinate bounds. The Point Record Metadata block describes the structure of the point data. The variable length records contain variable types of data including coordinate system information, metadata and user defined auxiliary data. The point data records contain the actual LAS data points.

The actual content and format of an LAS point record is user definable. The ASPRS LAS Committee defines a set of mandatory and optional standard LAS point field records such as the X, Y, Z coordinates of the point, the reflectance classification and so forth. Users can extend the standard in any manner they choose. Regardless of the user extension schema, any LAS 2.0 compliant software will be able to read the specification-required data fields.

# 3 DATA TYPES:

The following data types are used in the LAS format definition.

**Table 3-1 Data types supported in LAS 2.0**

| **Moniker** | **Type** | **Size (in Bytes)** | **Description** |
|---|---|---|---|
| BOOL | Unsigned Char | 1 | 1 == TRUE, 0 == FALSE |
| I1 | Char | 1 | |
| UI1 | Unsigned Character | 1 | Also used for ASCII arrays |
| I2 | Signed Short | 2 | |
| UI2 | Unsigned Short | 2 | |
| I4 | Signed Long | 4 | |
| UI4 | Unsigned Long | 4 | |
| I8 | Signed 8 byte integer | 8 | |
| UI8 | Unsigned 8 byte integer | 8 | |
| R4 | IEEE Float | 4 | |
| R8 | IEEE Double | 8 | |
| R10 | Intel Extended precision Double | 10 | |
| STR | NULL (zero) terminated variable length string | VAR | Variable length string of UI1 (characters) terminated by a UI1 whose value is zero (the null terminator). Note that the length of a String is always the number of characters in the string plus 1. |
| BFx | bit field | x, where x = 1,2,4,8 | An unsigned integer (of value 1, 2, 4 or 8 bytes) encoded as a bit field. |
| **[n]** | Array | VAR | Brackets indicate a fixed size array of n elements of the specified type (e.g. UI1[4] is an array of 4 characters) |

A note on Bit Fields – The LAS storage format is "Little Endian." This means that multi-byte data fields are stored in memory from least significant byte at the low address to most significant byte at the high address. Bit fields are always interpreted as bit 0 set to 1 equals 1, bit 1 set to 1 equals 2, bit 2 set to 1 equals 4 and so forth.

# 4 FILE HEADER BLOCK:

The first section of the file is a "File Header Block" (note that *all* fields are required)

**Table 4-1 File Header Block**

| Item | Format | Size |
|---|---|---|
| File Signature ("LASF") | UI1[4] | 4 bytes |
| File ID | UI4 | 4 bytes |
| Project ID - GUID data 1 | UI4 | 4 bytes |
| Project ID - GUID data 2 | UI2 | 2 bytes |
| Project ID - GUID data 3 | UI2 | 2 bytes |
| Project ID - GUID data 4 | UI1[8] | 8 bytes |
| Version Major | UI1 | 1 byte |
| Version Minor | UI1 | 1 byte |
| System Identifier | UI1[32] | 32 bytes |
| Generating Software | UI1[32] | 32 bytes |
| File Creation Date | UI4 | 4 bytes |
| File Creation Time | UI4 | 4 bytes |
| File Header Size | UI2 | 2 bytes |
| Source ID Packet | BF2 | 2 bytes |
| Offset to Point Record Metadata Block | UI2 | 2 bytes |
| Size of Point Record Metadata Block | UI2 | 2 bytes |
| Point Data Record Length | UI2 | 2 bytes |
| Offset to point data | UI8 | 8 bytes |
| Number of point records | UI8 | 8 bytes |
| Offset to first variable length record | UI4 | 4 bytes |
| Number of variable length records | UI4 | 4 bytes |
| Number of points by return | UI4[16] | 64 bytes |
| Point Record Compatibility | UI1 | 1 byte |
| Coordinate System Type | UI1 | 1 byte |
| Metric Units | UI1 | 1 byte |
| Vertical Metric Units | UI1 | 1 byte |
| X Origin | R8 | 8 bytes |
| Y Origin | R8 | 8 bytes |
| Z Origin | R8 | 8 bytes |
| Apply Scaling | BOOL | 1 byte |
| X scale factor | R8 | 8 bytes |
| Y scale factor | R8 | 8 bytes |
| Z scale factor | R8 | 8 bytes |
| Apply Offsets | BOOL | 1 byte |
| X offset | R8 | 8 bytes |
| Y offset | R8 | 8 bytes |
| Z offset | R8 | 8 bytes |
| Max X | R8 | 8 bytes |
| Min X | R8 | 8 bytes |
| Max Y | R8 | 8 bytes |
| Min Y | R8 | 8 bytes |
| Max Z | R8 | 8 bytes |
| Min Z | R8 | 8 bytes |

## 4.1 Field Descriptions

The following subparagraphs describe the content of each field of the File Header Record.

### 4.1.1 FILE SIGNATURE

The file signature must contain the four characters "LASF", and it is required by the LAS specification. These four characters can be checked by user software as a quick-look initial determination of file type.

### 4.1.2 FILE ID

This field should be set to a value between 1 and $2^{32}$ -1, inclusive. A value of zero (0) is interpreted to mean that an ID has not been assigned (this condition is considered a non-fatal error). In this case, processing software is free to assign any valid number as the File ID and will assume that Point Source IDs have not been set (see Point Record information). A source can be considered an original data capturer identification ID or it can be the result of merge and/or extract operations.

For an airborne system, the File ID is equivalent to the Flight Line Number if this file was derived from an original airborne LIDAR flight line.

For a terrestrial scanner, the File ID uniquely identifies a single scanning operation. Note that in terrestrial scanning operations, two distinct File IDs do not necessarily imply two physically separate scanning stations.

The File ID is assigned when a file is first introduced to a Project (see Project ID). For example, if a USGS DEM were converted to LAS format and introduced to a project, it would be assigned a unique ID.

File IDs must be unique across a project. That is, any two files with the same Project ID must have different File IDs.

### 4.1.3 PROJECT ID (GUID DATA)

The Project Identifier (Project ID) is a Globally Unique Identifier (GUID) that is used to uniquely identify a collection of LAS files as being members of the project identified by the Project ID. The Project ID should be the same for all files that are associated with a unique project. By assigning a Project ID and using a File ID every file within a project and every point within a file can be uniquely identified in all time and space.

A Project ID consisting of all zero values implies that the Project ID scheme is not in use.

### 4.1.4 VERSION NUMBER (MAJOR, MINOR)

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 2.0 (this version) would contain 2 in the major field and 0 in the minor field.

### 4.1.5 SYSTEM IDENTIFIER

Files often result from extraction, merging or modifying existing data files. The System ID is used to identify the source of the data in *this* file. Recommended Identifiers are listed in Table 4-2.

**Table 4-2 System Identification**

| **Generating Agent** | **System Identifier** |
|---|---|
| Hardware system | String identifying hardware (e.g. "ALTM 1210" or "ALS-50" |
| Merge of one or more files | "MERGE" |
| Modification of a single file | "MODIFICATION" |
| Extraction from one or more files | "EXTRACTION" |
| Reprojection, rescaling, warping, etc. | "TRANSFORMATION" |
| Some other operation | "OTHER" or a string up to 32 characters identifying the operation |

If the character data is less than 32 characters, the remaining data must be set to zero.

### 4.1.6 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 (e.g. "TerraScan V-10.8", "REALM V-4.2", "GeoCue 4.0" and etc.). If the character data are less than 32 characters, the remaining data must be set to zero.

### 4.1.7 FILE CREATION DATE

Modified Julian Day (MJD). This is the integer number of days that have elapsed since midnight, Universal Time, at the beginning of Wednesday November 17, 1858.

### 4.1.8 FILE CREATION TIME

The number of seconds into the Julian day with midnight being zero

### 4.1.9 FILE HEADER SIZE

The size, in bytes, of the file header block (this block) itself. Unlike previous versions of the LAS specification, extension of the File Header Block by users is not permitted. This field has a value of **322** for Version 2.0.

### 4.1.10 SOURCE ID PACKET

The Source ID Packet is a bit field that is used to rapidly identify the data type(s) contained in the file. If all bits are zero, software can safely assume that the Source has not been specified.

**Table 4-3 Source ID Packet**

| ***Bit(s)*** | ***Field Name*** | ***Notes*** |
|---|---|---|
| 0 | Airborne Sensor only | If set, data was solely collected from an airborne sensor |
| 1 | Terrestrial Sensor only | If set, data originally was solely collected from a terrestrial sensor |
| 2 | Original Flight Line | Data in this file came from a single flight line (implies that bit 0 should also be set) |
| 3 | Original Terrestrial Scan | Data in this file came from a single terrestrial scan (implies that bit 1 should also be set) |
| 4 | Mixed Flight Line file | This file contains data from multiple flight lines (implies bit 0 should also be set) |
| 5 | Mixed Terrestrial Scans | This file contains data from multiple terrestrial scans (implies bit 1 should also be set) |
| 6 | Hybrid Airborne, Terrestrial data | Data in this file resulted from merging both airborne and terrestrial scan data – implies both bits 0 and 1 should be reset (zero) |
| 7 | Stereo Image automatic correlation | Data in this file were derived from stereo image extraction using an automatic correlation technique (Automatic Terrain Extraction, ATE) |
| 8 | Manual Stereo Collection | Data in this file were collected using interactive (manual) stereo collection from visible imagery |
| 9 | SAR | Data in this file were derived from synthetic aperture radar |
| 10 | Elevation Import | Data in this file were created by importing an elevation file product (such as a USGS DEM or an NGA DTED) |
| 11:15 | RESERVED | |

### 4.1.11 OFFSET (IN BYTES) TO POINT RECORD METADATA BLOCK

The number of bytes from the beginning of the LAS file to the first byte of the Point Record Metadata Block. This value should be the same as the value contained in the File Header Size field (i.e. there can be no additional fields inserted between the File Header Block and the start of the Point Record Metadata Block). The value is **322** for LAS Version 2.0.

### 4.1.12 SIZE (IN BYTES) OF THE POINT RECORD METADATA BLOCK

The size, in bytes, of the Point Record Metadata Block. Note that the Point Record Metadata block is a variable size record and thus this value can vary from file to file.

### 4.1.13 POINT DATA RECORD LENGTH

The size of a point data record, in bytes. Note that content and hence the size of a point data record is not fixed in the LAS 2.0 specification. Thus applications must be certain to update this field. Note that all point data records in a single file must be the same length.

### 4.1.14 OFFSET TO POINT DATA

The actual number of bytes from the beginning of the file to the first byte of the first point record data field. This data offset must be updated if any software adds/removes data from the variable length records.

### 4.1.15 NUMBER OF POINT RECORDS

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

### 4.1.16 OFFSET TO FIRST VARIABLE LENGTH RECORD

The actual number of bytes from the beginning of the file to the first byte of the first variable length record data field.

### 4.1.17 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.

### 4.1.18 NUMBER OF POINTS BY RETURN

This field contains an array of the total point records per return. The first UI4 value (array index 0) will be the total number of records for NULL returns (see Point Record description for the definition of a null return), the second UI4 value will contain the total count for the first returns, the third contains the total number for return two, and so forth up to 15 returns. If the collection system does not support multiple returns or the source was from a non-sensor then the second entry in this array should equal the total number of point records and the remaining entries should be set to zero.

### 4.1.19 POINT RECORD COMPATIBILITY MODE

This flag indicates if the point records are compatible with LAS 1.1 record formats. If set to 0 or 1, the point records in this file are identical to the corresponding point record types in the prior formats. This enables quick conversion of existing files. Note that software, upon noting the compatibility flag, can safely skip the point record descriptors.

| ***Compatibility Mode*** | ***Definition*** |
|---|---|
| 0 | Point records are compatible with LAS 1.1 point record type 0 |
| 1 | Point records are compatible with LAS 1.1 point record type 1 |
| 2-255 | Reserved |

### 4.1.20 COORDINATE SYSTEM TYPE

Coordinate System types are defined as:

| ***Type*** | ***Definition*** |
|---|---|
| 1 | Geographic or projected coordinate system as defined by Well Known Text (WKT) definitions in variable length records |
| 2 | Cartesian, in which case X, Y and Z have their standard Cartesian meaning |
| 3 | Positive Spherical (defined in the positive spherical definition table) |
| 4 | Negative Spherical (defined in the negative spherical definition table) |

The spherical coordinate systems are described in the following two tables:

**Table 4 Positive Spherical**

| ***Value*** | ***Meaning*** |
|---|---|
| X = *r* | the radial distance of the point from the origin |
| Y = θ | the azimuthal angle in the X, Y plane from the *x*-axis with $0 \le \theta \le 2\pi$, positive direction counter-clockwise |
| Z = φ | the polar angle from the positive Z axis, $0 \le \varphi \le \pi$ |

**Table 5 Negative Spherical**

| ***Value*** | ***Meaning*** |
|---|---|
| X = *r* | the radial distance of the point from the origin |
| Y = θ | the azimuthal angle in the X, Y plane from the *x*-axis with $0 \le \theta \le \pi$, positive direction counter-clockwise |
| Z = φ | the polar angle from the negative Z axis, $0 \le \varphi \le 2\pi$ |

Angles in spherical coordinates are always expressed in radians.

### 4.1.21 METRIC UNITS

This field specifies the units of the data (applies only to the *r* parameter for spherical coordinate systems; angles are always in radians) when the coordinate system type is 2 or 3 (units for type 1 are specified in the WKT record in which case this field should be set to 0). The valid values and their meanings are listed in Table 4-6.

**Table 4-6 Metric Unit Definitions**

| ***Value*** | ***Meaning*** |
|---|---|
| 0 | Units are defined in the WKT file |
| 1 | Meters |
| 2 | International Feet |
| 3 | U. S. Survey Feet |
| 4 | Arc Seconds (Cartesian coordinate systems only) |

### 4.1.22 VERTICAL METRIC UNITS OVERRIDE

(ONLY VALID FOR CARTESIAN COORDINATE SYSTEMS)

This field specifies the vertical units of the data when the coordinate system is Cartesian and the units are different from the horizontal units. This field can be set to zero when it is not applicable.

**Table 4-7 Vertical Metric Unit Definitions**

| ***Value*** | ***Meaning*** |
|---|---|
| 0 | Use units defined in the Metric Units field |
| 1 | Meters |
| 2 | International Feet |
| 3 | U. S. Survey Feet |

### 4.1.23 X, Y, Z ORIGIN

The sensor origin (within the specified global coordinate system) if spherical coordinates are used. These fields must be set to 0.0 for Cartesian coordinates (they should be set to 0.0 when not applicable).

### 4.1.24 APPLY SCALING

This Boolean indicates if scale factors should be applied to coordinates (next three fields). If TRUE, the scale factors must be applied to <u>all</u> point coordinates.

### 4.1.25 X, Y, AND Z SCALE FACTORS

The scale factor fields contain an R8 floating point value that is used to scale the corresponding X, Y, and Z 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. Scale must be applied if the "Apply Scale" flag is set to TRUE.

### 4.1.26 APPLY OFFSETS

This Boolean indicates if offset factors should be applied to coordinates (next three fields). If TRUE, the offset factors must be applied to <u>all</u> point coordinates.

### 4.1.27 X, Y, AND Z OFFSET

The offset fields should be used to set the overall offset for the point records. Offsets must be applied if the "Apply Offset" flag is set to TRUE. 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.

If both scale and offset have been applied then the final coordinates are determined via:

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

#### 4.1.28 MAX AND MIN X, Y, Z

The maximum and minimum data fields are the actual file coordinate extents of the LAS point file data. Note that Origin, Scale and Offset must be applied, where appropriate, to these values.

# 5 Point Record Metadata Block

## 5.1 Metadata Block Description

The first variable length record in the file is the **Point Record Metadata Block**. This block specifies the content and format of the point data records (the heart of the file). Note that in LAS Version 2.0, the format has been generalized such that a user can construct any conceivable point record structure.

The structure of this record is a consecutive set of zero terminated string pairs that comprise field name and data type. All fields defined by this specification begin with "LASF_". Field Name and Format strings are *not* case sensitive. Thus the pairs:

"lasf_x", "i4"
"Lasf_x", I4"
"laSf_X", "i4"

all have the same meaning.

Users can extend the point record specification by defining their own data types in an analogous manner. For example, if atmospheric pressure were desired by the XYZCorp on a point by point basis, the following fields could be added to the header:

"XYZCorp_Atm_Press", "R4"

Note that entities that want to extend the point record definitions should obtain a registration key from the LAS Standards Group (lasformat.org) to ensure that their identifying string is unique (e.g. "XYZCorp" in the above example).

## 5.2 Point Record LAS Defined Content

The following table lists all of the ASPRS Version 2.0 recognized fields and whether or not the field is mandatory. Note that only "LASF_END" is required to be in a specific position (the end) within this Metadata description. The order of data within the point records is defined by the order of the field descriptors in the Point Record Metadata Block. Note that if this file is to be compatible with LAS 1.1, the required fields must be in the same order as in the LAS 1.1 specification (they are in correct order in Table 5-1)

The field "LASF_END" is only included in the Metadata Block. The point data records do not contain an end of record indicator.

**Table 5-1 Point Data Record Standard Fields**

| **Field Name** | **Valid Formats** | **Required[1]** |
|---|---|---|
| LASF_X | I4, I8, R4, R8, R10 | M, 1 |
| LASF_Y | I4, I8, R4, R8, R10 | M, 1 |
| LASF_Z | I4, I8, R4, R8, R10 | M, 1 |
| LASF_Intensity | UI1, UI2, UI4, R4, R8 | O, 1 |

[1] M = Mandatory, C = Conditional (as determined by Sensor Field in Main Header), O = Optional, , 1 = Mandatory if impersonating LAS Release 1.1 Record Type 1 and must be of the same data type as in LAS 1.1.

| Field Name | Valid Formats | Required[1] |
|---|---|---|
| LASF_Return_Packet | BF1 | C[2], 1 |
| LASF_Classification | UI1 | M, 1 |
| LASF_Airborne_Scanner_Packet | UI1 | O, 1 |
| LASF_User_Data | UI1 | O, 1 |
| LASF_Point_Source_ID | UI2, UI4, UI8 | M, 1 |
| LASF_GPS_Week_Time | R8 | O, 1 |
| LASF_GPS_Week | UI2 | O |
| LASF_Sigma_X | R4, R8 | O |
| LASF_Sigma_Y | R4, R8 | O |
| LASF_Sigma_Z | R4, R8 | O |
| LASF_Pan | UI1, UI2, UI4, R4, R8 | O |
| LASF_Red | UI1, UI2, UI4, R4, R8 | O |
| LASF_Green | UI1, UI2, UI4, R4, R8 | O |
| LASF_Blue | UI1, UI2, UI4, R4, R8 | O |
| LASF_NIR | UI1, UI2, UI4, R4, R8 | O |
| LASF_Attributes | BF1 | M |
| LASF_Class_Confidence | UI1 | O |
| LASF_Group_ID | UI1, UI2, UI4, UI8 | O |
| LASF_POSIX_Time | R8 | O |
| LASF_PAD | UI1, UI2, UI4, UI8 | O |
| *User Defined Fields* | User Defined | O |
| LASF_END | N/A | M |

## 5.3 Field Definitions

The definition of each LAS defined field is contained in the following subsections.

### 5.3.1 X, Y, AND Z

The X, Y, and Z values as defined by Coordinate Type in the Main Header. The corresponding X scale, Y scale, and Z scale and offset values from the file header block must be applied if the corresponding "Apply Scale" and/or "Apply Offset" flags are true. The user should apply the usual precautions when converting between data types.

### 5.3.2 INTENSITY

The intensity value is the representation of the pulse return integrated energy magnitude. This value is system specific. Note that this field should never be used for encoding an auxiliary value such as intensity provided by an associated camera.

### 5.3.3 RETURN PACKET

The Return Packet is a one byte bit field. The lower 4 bits represent the number of this return pulse (return number) and the upper four bits represent the total number of returns that were detected for this pulse.

[2] Mandatory if these data were produced by an airborne sensor

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 fifteen possible returns.

Number of Returns (for this emitted pulse): The number of returns is the total number of detected returns for a given pulse. For example, a laser data point may be return two (return number) with a total number of five detected returns.

Zero is reserved for the special case of a pulse being emitted by the sensor but no return pulse was detected (a null pulse). For this case, the number of returns field will also be set to zero. The X, Y, Z coordinates are invalid but the GPS Time, if included in the data format, represents the time the pulse was fired. Not all sensors are capable of detecting the occurrence of these null pulses.

> NOTE: Multi-return pulses *must* be sequentially placed in the LAS file. That is, for a single fired pulse with 3 detected returns, the data records must be in the sequence return 1 followed by return 2 followed by return 3 with no other intervening data.
>
> Software applications must be extremely careful regarding this rule since it is easy to mix data pulses when merging or extracting files.

### 5.3.4 CLASSIFICATION

This field is used to specify the type of object that the point represents. If a point has never been classified, this byte must be set to zero. There are no user defined classes available. If user defined classes are needed, the user should define a new user defined field for this purpose (e.g. XYZCORP_User_Classification). If a user classification field is added, the standard field must also be populated. For example, if a user needed to define a number of different building classes, then the Building (Generic) field should be set in the standard class and the specific type of building in the user defined classification. Setting the high bit of the Classification byte indicates that the user has supplied a custom classification field that contains additional classification information. In the above example, the user could set a value of 6 for building, set the high bit of the classification byte and indicate specific building types in a custom classification field.

Classification must adhere to the following LAS 2.0 standard (we expect to assign the ASPRS Reserved values as LAS Version 2.0a, 2.0b, 2.0c and etc.):

**Table 5-2 ASPRS Standard LIDAR Point Classes**

| ***Classification Value (b:0 –b:6)*** | ***Meaning*** |
|---|---|
| 0 | Created, never classified |
| 1 | Unclassified[3] |

[3] We are using both 0 and 1 as ***Unclassified*** to maintain compatibility with current popular classification software such as TerraScan. We extend the idea of classification value 1 to include cases in which data have been subjected to a classification algorithm but emerged in an undefined state. For example, data with

| ***Classification Value (b:0 –b:6)*** | ***Meaning*** |
|---|---|
| 2 | Ground (bare earth surface) |
| 3 | Low Vegetation |
| 4 | Medium Vegetation |
| 5 | High Vegetation |
| 6 | Building (Generic) |
| 7 | Low Point |
| 8 | *High Point* |
| 9 | Water |
| 10 | *Bridge* |
| 11 | *Road* |
| 12 | *Miscellaneous Structure* |
| 13 | Railroad |
| 14 | Stream |
| 15 | Power Line |
| 16 | Power Line Tower |
| 17 | Power Line Tower to wire connection |
| 18-127 | *Reserved for ASPRS Definition* |
| b7 (high bit set) | *Farther classification provided in a custom classification field* |

### 5.3.5 AIRBORNE SCANNER PACKET

Some airborne LIDAR post-processing software packages are able to encode information about the scanner mirror (or equivalent fiber-optic data). The Airborne Scanner Packet is an encoded field containing the following fields:

**Table 5-3 Airborne Scanner Packet**

| ***Bit(s)*** | ***Name*** | ***Description*** |
|---|---|---|
| 0 | 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. The positive direction is taken to be from left side of flight line to right side of flight line when viewed in the direction of travel of the platform. |
| 1 | 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 the scanner mirror changes direction. |
| 2 | Mirror Centered | This point was acquired with the scan mirror at its most central position (This would be nadir if the aircraft roll were zero) |

class 0 is sent through an algorithm to detect man-made structures – points that emerge without having been assigned as belonging to structures could be remapped from class 0 to class 1.

| Bit(s) | Name | Description |
|---|---|---|
| 3:7 | RESERVED | |
| 8:15 | Scan Angle Rank | The scan angle rank is a signed 8 bit number with a valid range from -90.00 to +90.00. The scan angle rank is the angle (rounded to the nearest integer in the absolute value sense) at which the laser point was output from the laser system excluding[4] 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. |

Note that this packet is included only for backward compatibility with LAS 1.0 and LAS 1.1. It is strongly recommended that a custom field be added for rich data content, when available.

### 5.3.6 USER DATA

This field is carried over from LAS 1.1 to allow point record compatibility. It is optional except when impersonating LAS 1.1 Point Record types.

### 5.3.7 POINT SOURCE ID

This value indicates the initial file from which this point originated. This value is used as a trace back to original point sources. For example, if the original files were from airborne flight lines (one file per line) and a composite file were created by merging a number of these flight line files, then the Point Source ID would be equivalent to the flight line number.

Once assigned, a Point Source ID should never change. This means that the Point Source ID combined with the Project ID (see File Header Block) can be used to trace the original source of an LAS point regardless of the number of files through which the point has propagated.

Valid values for this field are 1 to the maximum count that can be contained in the data type selected. Zero is used for a special case discussed below. The numerical value corresponds to the File ID from which this point originated.

Zero is reserved as a convenience to system implementers. A Point Source ID of zero implies that this point originated in this file. This implies that processing software must set the Point Source ID equal to the File ID of the file containing this point during a merge or extract processing operation.

### 5.3.8 GPS WEEK TIME

The GPS week time, in seconds, associated with this point. This time must match the time used in an associated trajectory file. The time starts at zero on midnight of Saturday and repeats each week. The typical resolution is 200 Hz. GPS Week Time has been carried over to LAS 2.0 to allow point record compatibility with LAS 1.1 Point Record type 1. Users are strongly discouraged

[4] Note: This is a change from version 1.1 where aircraft roll was included in scan angle rank.

from using GPS Week Time since merging files from different GPS Weeks results in ambiguous times.

### 5.3.9 GPS WEEK

Users are strongly encouraged to use GPS Week Number if they are using GPS Week Time (and not trying to clone LAS 1.1 record structures). The GPS Week Number count began at approximately midnight on the evening of 05 January 1980 / morning of 06 January 1980. Since that time, the count has been incremented by 1 each week, and broadcast as part of the GPS message. The GPS Week Number field is modulo 1024. This means that at the completion of week 1023, the GPS week number rolled over to 0 on midnight GPS Time of the evening of 21 August 1999 / morning of 22 August 1999. Note that this corresponded to 23:59:47 UTC on 21 August 1999.

### 5.3.10 SIGMA_X, SIGMA_Y, SIGMA_Z

The standard deviation of X, Y and Z respectively. These values should be set to 0 for absolute known values and less than zero for completely unknown values. The units are the same as for the parameter being characterized.

### 5.3.11 PAN, RED, GREEN, BLUE, NIR

These fields represent potential Panchromatic (PAN), Red, Green, Blue and/or Near-infrared values that have been associated with the point (usually from an auxiliary camera in the case of a sensor such as LIDAR or from image draping in the case of generalized elevation data).

Users should only store data that actually originated from the specified channel type in these parameters. For example, using the common practice of storing panchromatic data in the green channel will cause data from different files to be improperly merged.

### 5.3.12 ATTRIBUTES

Attributes are bit flags that signify general conditions of the individual points. The format for attributes is a bit encoded unsigned integer. The bit definitions are:

**Table 5-4 Attribute Bit Field Encoding**

| ***Bits*** | ***Field Name*** | ***Description*** |
|---|---|---|
| 0 | Class Verified | This point's class is considered known (e.g. if the class is Ground and this bit is set, the point is known to be ground to a "very high" degree of certainty). Note that if this bit is not set, the meaning is that no statement is being made regarding confidence of classification. This field is useful when the optional "LASF_Class_Confidence" byte is not in use. |

| Bits | Field Name | Description |
|---|---|---|
| 1 | Control Point | This point has been inserted or marked for control purposes. |
| 2 | Breakline[5] | This point forms a vertex of a breakline (It is recommended that the Group ID field be used in conjunction with this bit to define specific breaklines) |
| 3 | Noise | This point has been identified as noise |
| 4 | Overlap | If set, this point has been marked as being in the overlap region of two or more scans (NOTE – It is not mandatory that this bit be set and thus one should not conclude that a Zero in this position implies that the point is not in an overlap region) |
| 5 | Synthetic | If set then this point was created by a technique other than direct sensor collection such as digitized from a photogrammetric stereo model. |
| 6 | Key | If set, this point is considered to be a model key-point and thus generally should not be withheld in a thinning algorithm. |
| 7 | Withheld | If set, this point should not be included in processing (synonymous with Deleted). |

Note that bits are treated as flags and can be set or clear in any combination. For example, a point with bits 1, 5 and 6 all set to one and other bits set to 0 would be a *control* point that had been *Synthetically* collected and marked as a *model key-point*.

### 5.3.13 CLASS CONFIDENCE

This optional value is a measure of the confidence of the classification of the point. 0 means no confidence and 255 means absolute confidence.

Note that the LASF_Attributes bit field contains a bit that can be used to signify absolute confidence in a classification. This is useful when the confidence only needs to be known for the absolute condition.

### 5.3.14 GROUP ID

The Group ID is an optional field that can be used for *associativity.* For example, it could be used as the index (key value) in an external database used for feature coding (e.g. Group ID 1124 == First State Bank implying all points encoded with a Group ID of 1124 are reflecting from this particular structure).

Group ID used in conjunction with Attributes and Class allow the construction of context. For example, points with the breakline attribute set, class set to Stream and Group ID set to 117

[5] Breakline topology can be determined by using Group number (identifies membership in a specific breakline) and Time_Stamp (identifies the order of vertices within the group).

would be known to be forming a linear breakline for hydrography. The Time Stamps would be used to determine vertex order.

#### 5.3.15 POSIX TIME

The time, in seconds, since January 1, 1970 in Universal Coordinated Time (UTC), associated with this point. This time must match the time used in an associated trajectory file. The typical resolution is microseconds. Users are strongly encouraged to use POSIX time rather than GPS Week Time since POSIX time will be unique across all data files.

#### 5.3.16 PAD

This is an optional field used to "pad" a point data record to a particular length. It is typically used to address machine specific address alignment issues. The content of the PAD field is undefined. PAD fields should never be used to carry information because they are not required to be included during file merge and extract operations.

#### 5.3.17 USER DEFINED FIELDS

Users can extend the data packet definitions to any extent desired. The extension is made by creating a unique field name, data type pair. The unique field name must be prefixed by the user's LAS Key. Vendors obtain the LAS Key from the ASPRS LIDAR Committee (*see www.lasformat.org).*

For example if the XXZ Corporation obtained the registered LAS key of "XYZCORP" then they could add a new point data field by simply adding their new field name and data type strings to the Point Data Record Description:

"XYZCORP_Energy", "R4"

Note that the only field that must be placed in a specific location in the Point Data Record Description is the terminal field, "LASF_END"

#### 5.3.18 LASF_END

The end of the Point Data Record Description is indicated by the string "LASF_END". This string must be the last field in the Point Data Record Description. It is immediately followed by the string null terminator which is then immediately followed by the first byte of the first variable length record.

## 5.4 An Example Point Record Metadata Block

An example of a typical Point Record Metadata Block for an airborne sensor is:

“LASF_POSIX_Time”, “R8”,“LASF_X”, “R4”, “LASF_Y”, “R4”, LASF_Z”, “I4”, “LASF_Intensity”,”UI1”“LASF_Classification”, “UI1”, “LASF_ATTIBUTES”, “UI1”, “LASF_Return_PACKET”, “UI1”, “LASF_Point_Source_ID”,”UI4”, “LASF_END”

Note that in this example we mixed the data type for horizontal and vertical coordinates and also the order (with respect to the LAS specification) in which data fields occur in the point data structures. Also notice that we used a mixed set of upper and lower case characters.

Note that the size of each point record for this particular example is 28 bytes.

# 6 Variable Length Records

The Point Record Metadata Block is followed by one or more Variable Length Records. There are two mandatory Variable Length Records – **LASF_WKT_Primary, LASF_WKT_Secondary**. The number of variable length records is specified in the "Number of variable length records" field in the File Header Block. The variable length records must be accessed sequentially since the size of each variable length record is contained in the Variable Length Record Header. Each Variable Length Record Header is 58 bytes in length.

## 6.1 Variable Length Record Header

Each variable length record begins with a fixed size header:

**VARIABLE LENGTH RECORD HEADER**

| **Item** | **Format** | **Size** | **Required** |
|---|---|---|---|
| User ID | UI1[16] | 16 bytes | * |
| Record ID | UI2 | 2 bytes | * |
| Description | UI1[32] | 32 bytes | |
| Record Length After Header | UI8 | 8 bytes | * |

### 6.1.1 USER ID

The user ID field is ASCII character data that identifies the user which created the variable length record. It is possible to have 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 set to null. The user ID must be registered with the LAS specification managing body. The management of these IDs ensures that no two individuals accidentally use the same ID. "LASF" is reserved for use by the specification itself.

### 6.1.2 RECORD ID

The record ID is dependent upon the User ID. There can be 0 to 65,535 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. Thus each User ID is allowed to assign 0 to 65535 record IDs in any manner they desire. 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.

### 6.1.3 DESCRIPTION

Optional 32 byte text description of the data. Any remaining characters not used must be null.

### 6.1.4 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. Thus the entire record length is 58 bytes (the header size) plus the number of bytes in the variable length portion of the record.

## 6.2 Georeferencing Information

Georeferencing for the LAS format uses the Well Known Text (WKT) mechanism promulgated by the Open Geospatial Consortium (OGC) and supported by a number of vendors. The WKT convention used in the LAS Specification is the ESRI[6] compatible version (that is, the LAS WKT can be directly used in ArcGIS products).

### 6.2.1 PRIMARY WKT RECORD (MANDATORY IF COORDAINTE SYSTEM TYPE == 1)

User ID: LASF
Record ID: 0
Description: Primary_WKT
Record Length after Header: Variable

This record contains the literal ESRI compatible Well Known Text (WKT) definition of the coordinate system. It should encode both the horizontal and vertical systems. The WKT is a null terminated string. Note that this string, if literally written to disk as a text file with the extension ".prj" can be read in as the coordinate system definition to any ESRI coordinate system compatible software.

Record Content:
STR – ESRI compliant WKT coordinate system null terminated string

### 6.2.2 SECONDARY WKT RECORD (OPTIONAL)

User ID: LASF
Record ID: 1
Description: Secondary_WKT
Record Length after Header: Variable

This record contains the literal OGC compatible Well Known Text (WKT) definition of the coordinate system. It should encode both the horizontal and vertical systems. The WKT is a null terminated string.

Record Content:
STR – OGC compliant WKT coordinate system null terminated string

Note: This secondary record is used for two purposes. The first is that standard coordinate systems passed around by processing software often do not encode the vertical coordinate system. In these cases, users can encode the vertical system in the secondary key without rendering the primary key incompatible with processing software that misinterprets the vertical tags. The second use of this record is to support cases in which processing software can interpret OCS WKS encodings but not ESRI WKT variations.

# END OF SPECIFICATION

[6] ESRI and ArcGIS are registered trademarks of Environmental Sciences Research Institute, Inc.