
GeoCue Group Support

January 28, 2015

Revision 1.0

Introduction

We added support for v1.4 of the ASPRS LAS data format to GeoCue v2014. If you are involved in LIDAR workflows in North America, you will immediately understand the value of supporting LAS Version v1.4, Point Data Record Formats (PDRFs) 6 through 8. These are the required deliverable formats for the United States Geological Survey (USGS) 3D Elevation Program and any other projects following the USGS LIDAR Base Specification¹. Of course, this will drive all standard LIDAR deliverables in North America so it will be critical to move to the new PDRFs available only in LAS version 1.4. This technical note provides details on how the GeoCue LAS v1.4 implementation addresses two challenges created by the adoption of LAS v1.4:

- Terrasolid software does not support LAS v1.4.² Organizations using these tools as their primary production software need to work in LAS v1.2 and convert to v1.4 prior to final delivery.
- Legacy data sets in v1.2 (or earlier) are still common so a conversion strategy to update these data sets to v1.4 is desirable.

We have added tools to GeoCue 2014 allowing you to continue processing data in LAS v1.2 using your existing Terrasolid workflow, but convert to LAS 1.4 for final delivery. If you are a current Terrasolid owner, you will be interested in the seamless LAS v1.4 workflows GeoCue now offers. The same conversion tools can be used to update legacy data sets. Note that LP360 users can already work natively in LAS v1.4 as LP360 fully supports v1.4 end-to-end processing so this technical note is not needed for LP360 or LP360 for GeoCue workflows. This technical note provides details on how GeoCue supports LAS v1.4 workflows.

Background

The American Society of Photogrammetry and Remote Sensing (ASPRS) LAS file format is a standard electronic data file format used for storing point cloud data. It was originally developed to support the processing and exchange of airborne LIDAR data sets, but has been extended to cover any point cloud data regardless of the originating sensor type. Data in LAS format is usually generated by pre-processing software provided by hardware vendors, or by converting from other file formats in post-processing software. LAS is an open format that allows different hardware and software tools to output and exchange point cloud data in a common format. Technical specifications and full details on the LAS format are available from asprs.org. The latest version of LAS as of January 2015 is v1.4.

¹ Heidemann, Hans Karl, 2014, Lidar base specification (ver. 1.2, November 2014): U.S. Geological Survey Techniques and Methods, book 11, chap. B4, 67 p. with appendixes, <http://dx.doi.org/10.3133/tm11B4>

² As of January 2015; we expect support for v1.4 will be added at a future date.

To correctly implement LAS version conversions in your workflow, it is important to have a solid technical understanding of the LAS format and the differences between versions. A LAS formatted file contains binary data consisting of a public header block, any number of optional Variable Length Records (VLRs), the Point Data Records (PDRs) and any number of optional Extended Variable Length Records (EVLRS). The public header block contains summary data such as point numbers and point data bounds while the point data records contain the actual point information such as the XYZ values for each point. Point Data Records are further differentiated by assigning them a specific Point Data Record Format (PDRF) depending on the specific extent of the data being stored for each point. For example, if the Point Data Record contains RGB values for each point or not.

In LAS v1.2 point data records consist of a core 20 bytes of information (PDRF 0) and additional information defined in PDRFs 1-5. In LAS v1.4 additional PDRF definitions are added with an additional 10 bytes included in the core group for PDRFs 6-10. This additional information covers more bits for return numbers (up to 15), point classifications (up to 255), higher precision scan angles (16 bits instead of 8) and mandatory GPS time. The addition of this extra information in PDRFs 6-10 has implications for conversion strategies between formats, in particular for backwards conversion to earlier formats. Table 1 below summarizes the PDRFs defined in LAS v1.4.

PDRF	Description
0	Core 20 bytes shared by PDRF's 0-5
1	Adds GPS time information.
2	Adds three color channels for storing RGB values (colorized points).
3	Adds GPS time and RGB channels.
4	Adds GPS time and wave packets (for waveform data)
5	Adds GPS time, RGB channels and wave packets.
6	Core 30 bytes shared by PDRFs 6-10 including mandatory GPS time.
7	Adds three color channels for storing RGB values (colorized points).
8	Adds RGB channels and a fourth NIR channel.
9	Adds wave packets.
10	Adds wave packets and RGB channels.

Table 1 - Basic PDRF Descriptions

For purposes of this technical note we will discuss methods for converting between LAS v1.2 and LAS v1.4 excluding those PDRFs with wave packet information. Full technical details on the differences between public header blocks and how to work with VLRs and EVLRs are available in the public LAS definition documents available at asprs.org.

PDRF Conversions

To convert between LAS versions and different PDRFs it is important to understand the actual structural differences between each PDRF. We will focus our discussion on converting between LAS v1.2 PDRF 1 and LAS v1.4 PDRF 6. This is a very common conversion, particularly for Terrasolid users delivering to USGS guidelines. There are 28 bytes defining 13 items in PDRF 1 and 30 bytes defining 15 items in PDRF 6. A detailed breakdown of the differences between these two versions and PDRFs is included for convenience in Appendix A. We will refer to converting from v1.2 to v1.4 as up-conversion and from v1.4 to v1.2 as down-conversion.

Items with Straightforward (Trivial) Conversions

There are nine identical items shared between the two PDRFs that have identical definition, data type format and size. They are:

- X
- Y
- Z
- Intensity
- Scan Direction Flag
- Edge of Flight Line
- User Data
- Point Source ID
- GPS Time

Conversion between versions for these items is straightforward. Note that the User Data byte swaps position with Scan Angle Rank/Scan Angle in the data record, but this is easily handled in any conversion. The remaining six items are not identical in one or more of definition, data type format or size. Each requires a specific conversion strategy. As noted below, typically these remaining items will not support both up and down conversion without loss of information, or the introduction of ambiguity. For this reason GeoCue currently does not support down conversion from v1.4 to earlier versions. The remaining six items are special cases as follows:

Return Number/Number of Returns

The Return Number and the Number of Returns are related. Return Number is defined as 3 bits in PDRF 1, and 4 bits in PDRF 6. PDRF 1 technically supports return numbers up to seven, however, the LAS specification stipulates a maximum of five returns; PDRF 6 supports up to fifteen returns. This means up-conversion is trivial. Down-conversion of data sets from sensors known to have seven, or fewer, returns is also trivial. For data sets from sensors with more than seven returns, down-conversion is not well-defined and will result in a loss of information, so is not recommended. The same conversion logic applies to the Number of Returns (3 bits vs. 4 bits).

Classification & Classification Flags

The Classification byte requires special consideration. In PDRF 1 this byte is defined with the lower bits 0-4 assigned for 32 point classes (0-31) and the upper three bits 5-7 assigned for the Synthetic, Key-Point and Withheld flags. In PDRF 6 the full Classification byte is used for 256 point classes (0-255). The Synthetic, Key-Point and Withheld flags are moved to separate Classification Flag bits and a fourth flag, Overlap, is added. This creates several conversion cases that need to be handled appropriately:

Case A – From PDRF 1 and Classification Byte Contains Flags and Classes 0-31

In this case up-conversion is straightforward. Classes convert directly and the flags can be set appropriately in the PDRF 6 record. Note that the Overlap flag in PDRF 6 cannot be set directly from the PDRF 1 Classification byte flags as it is not defined in PDRF 1. A possible strategy to address this limitation is to use the overlap class (Class 12) to set the Overlap flag, however this method means you cannot up-convert and preserve both class and overlap together (i.e. you can't convert a ground class point flagged as overlap). We refer to this method as "class-to-flag" conversion. Implementing such class-to-flag conversions should be discussed with the end user of the data prior to implementation so they understand the trade-offs. GeoCue does not currently implement any forward, or reverse, class-to-flag conversions, but we are reviewing the need for this method in consultation with our users.

Case B – From PDRF 1 and Classification Byte Contains No Flags and Classes 0-255

In this case up-conversion is straightforward. Classes convert directly, however, the flags cannot be set appropriately in the PDRF 6 record since they are not defined in the PDRF 1 record. Again a class-to-flag strategy is a possibility, but should be discussed with the end user prior to implementation due to the limitations discussed above. For deliveries that need to comply with the USGS guidelines, this case is not supported since the Withheld and Overlap flags must be set correctly for USGS deliveries.

Case C – From PDRF 6 and Classification Byte Contains Classes 0-31

In this case down-conversion of the classes and flags is straightforward. Classes convert directly and any Synthetic, Key-Point or Withheld flags can be down-converted by setting the appropriate upper bit in the classification byte. The Overlap flag cannot be down-converted directly (there is no Overlap flag in PDRF 1), although a reverse class-to-flag method is an option. This would move Overlap flagged point records to the Overlap class. This will result in a loss of information if the point already has an assigned class. Again, any class-to-flag strategy should be discussed with the end user prior to implementation.

Case D – From PDRF 6 and Classification Byte Contains Classes 0-255

In this case down-conversion of the classes and flags is not straightforward. Similar to Case C, classes less than 32 would convert directly and any Synthetic, Key-Point or Withheld flags could be down-converted by setting the appropriate upper bit in the classification byte. However, classes greater than 31 will not down-convert correctly since they are not supported in PDRF 1. For this reason this down-conversion is not recommended for data sets with more than 31 classes.

Scan Angle Rank/Scan Angle

This item was renamed and redefined between v1.2 and v1.4. Scan Angle Rank in v1.2 PDRF 1 corresponds to Scan Angle in v1.4 PDRF 6, but with a different definition of scanner angle, and different resolutions. In PDRF 1, Scan Angle Rank is the scan angle rounded to the nearest integer from +90° to -90°, with -ve being left of 0° (nadir) looking forward. In PDRF 6, Scan Angle is a rotational position in 0.006° increments with values from +30,000 to -30,000 covering +180° to -180° (0° is at nadir/down, -ve is left/clockwise of nadir looking forward). In addition PDRF 1 is a single byte while PDRF 6 is two bytes. The PDRF 1 definition was implemented primarily to support airborne scanners, where the scanner is constrained to views either side of nadir and below the horizontal plane of the sensor. The updated PDRF 6 definition supports scanners with a full 360 degree field of view, such as mobile scanners that may scan above the horizontal plane of the sensor. See Figure 1 for a diagram of the two cases.

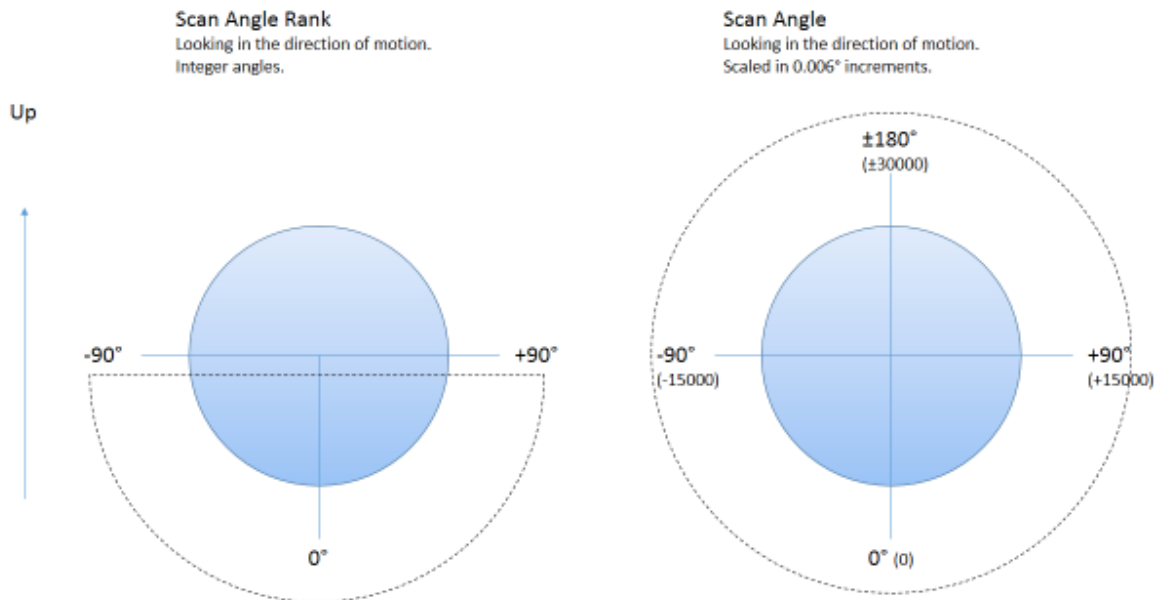


Figure 1 - Scan Angle Rank vs. Scan Angle

For up-conversion, Scan Angle Rank conversion to Scan Angle is straightforward and can be converted using:

$$\text{Scan Angle} = \text{Scan Angle Rank} / 90 * 15,000$$

For down-conversion, Scan Angle may be converted to Scan Angle Rank for sensors that maintain a field of view between +90° to -90° (typically airborne platforms), but with a loss of precision due to rounding from 0.006° resolution to integer angles. For this reason down-conversion is not recommended.

Scanner Channel

Scanner Channel consists of 2 bits in PDRF 6. There is no corresponding scanner information in the PDRF 1 definition. Up-conversion would only be possible by implanting a class-to-scanner channel conversion where specific classes were assigned to specific scanner channels and then used to set the scanner channel bits in the PDRF 6 record. This has the previously discussed class-to-flag limitation that the source data could not otherwise be classified or flagged prior to up-conversion. For this reason it is not recommended and typically during up-conversion of PDRF 1 files the Scanner Channel is set to '0' by default, corresponding to a single scanner system such as most airborne sensors. Down-conversion is also not recommended for the same reason; a scanner channel-to-class method would have to be implemented which would lose any classification information.

GeoCue Implementation of v1.4 Workflow

GeoCue is fully LAS v1.4 compliant for point data without waveform data. You can import LAS data in v1.4, process that data through your workflow, and export it for final delivery. However, TerraSolid's TerraScan does not support v1.4. Processing data through TerraScan requires you to work in v1.2, or earlier, and then convert to v1.4 prior to final delivery. This workflow is supported in GeoCue as described below.

Managing LAS File Versions

To manage different LAS versions within a GeoCue project, each LIDAR source entity and LAS Working Segment now includes a FormatID attribute that stores the LAS version and point data record format (PDRF) for the associated LAS data file. This attribute is automatically populated when entities are created, or populated from source data, or as format updates are applied. The attribute is located on the LAS Working Segment tab of the Properties pane. GeoCue supports PDRF's 0-8. PDRF's 9-10 cover the addition of wave packets for waveform data and are not currently supported in GeoCue. The reading of LAS PDRF's with waveform data is enabled, but the waveform information is removed during the population of Working Segments.

Properties			
General	Points		Process
System	Name	Value	Description
Files	NumberOfPoints	369,383	Number of points in the file.
	AverageElevation	930.613211057358	Average elevation of the points in the file.
DPMS	TimeStamps	True	Flag indicating whether the file contains GPS time stamps.
LAS Working...	MaxPointDensity	0.119510196510145	Maximum point density of the data.
	LASFileSourceID	-1	The source file ID for the data.
LAS_TSR	RGB	False	Flag indicating LAS point data has associated RGB values.
	FormatID	1.4 PF1	LAS specification / Point format ID
User Defined...	ExtraBytes	0	Number and description of extra bytes per point record

Figure 2 - FormatID Attribute on LAS Working Segment

GeoCue currently supports up-converting earlier LAS versions to LAS v1.4, via the Update LAS File utility (as discussed below), but does not support backward conversions of LAS v1.4 files to earlier versions. Note that up-conversion of earlier formats is not technically the same as generating the initial LAS files from the raw LIDAR data directly in LAS v1.4 PDRFs 6-10. For example, the greater scan angle precision available in PDRFs 6-10 cannot be recreated from an earlier LAS version file. Care should be taken to verify up-conversion is an acceptable method of creating LAS v1.4 PDRF 6-10 files for your end-user.

Interaction with TerraScan

Importing v1.4 files into TerraScan can result in the loss of information and potential file corruption. To prevent this, GeoCue enforces strict compliance with the LAS data file format in TerraScan workflows; you will not be able to load entities with LAS v1.4 data files into TerraScan, nor to assign and run TerraScan macros against them in GeoCue. A suitable warning message (Figure 3) will be displayed to the user should this case be detected. This restriction does not apply for editing tools that do support LAS v1.4, such as LP360, LP360 for GeoCue, or LP360 for ArcGIS.

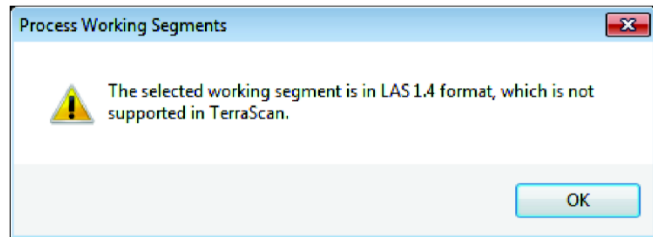


Figure 3 - Warning for LAS formats not supported in TerraScan

Update LAS File Utility

We have modified the Update LAS File utility step to allow you to convert LAS data files to v1.4. The step can be found on the Utilities tab of the standard LAS Working Segment checklist, or you can add it to the appropriate portion of your custom checklist using Environment Builder. The tool can be run against multiple LAS Working Segments at once, as in standard GeoCue multi-entity processing, and can also be dispatched, or distributed for processing using GeoCue's Command Dispatch System. When invoked the user will be presented with the dialog in Figure 4 - Set LAS File Version dialog in GeoCue.

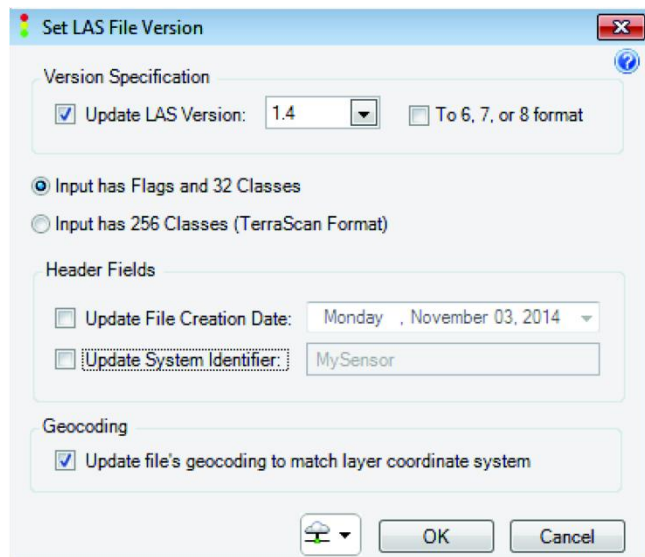


Figure 4 - Set LAS File Version dialog in GeoCue

To update a LAS file, specify the version you want to convert to in the first section. Typically this will be v1.4. Also, specify if you want to update the PDRF version to add the additional core bytes (PDRF 6, 7 or 8). If you select this option, GeoCue will automatically determine if your data is best suited for PDRF 6, 7, or 8 based on the data file itself, you do not need to specify. Note you can generate v1.4 files preserving PDRF versions of 0-5, LAS v1.4 does not require PDRF 6 or higher. However, most v1.4

deliverables, such as to the USGS, will likely be in LAS v1.4 PDRF 6-8. Keep in mind GeoCue does not support down-conversion so you cannot use this utility to rollback to an earlier LAS version.

The second section of the dialog specifies how to handle the classification byte when up-converting to LAS v1.4. This byte can either follow the official LAS standard classification byte definition (LAS v1.1-1.3) consisting of three flags (synthetic, key-point and withheld) and 32 standard classes (0-31), or as commonly used in TerraScan (but not actually valid by the official LAS v1.1-1.3 standard) 256 classes (the full byte). In LAS v1.4, this single byte gets expanded into a full byte for the classification, and 4 flag bits for synthetic, key-point, withheld and overlap. If you have flags set in your v1.2 PDRF 1 data, and you want to preserve these during conversion, you will specify the *Input has Flags and 32 Classes* option in this section. If you are not using flags, and have an extended class table (> 32 classes), and want to preserve the classifications during the conversion you will specify the *Input has 256 Classes* option. GeoCue will then handle the upper bits of the classification byte appropriately as flags, or classes, during the conversion.

In the third section of the dialog you can chose to update the header information for file creation date and system identifier.

Finally, if you chose to update the geocoding in the last section of the dialog, the LAS file coordinate reference system (CRS) information in the public header block will be updated to match the GeoCue layer coordinate system of your Working Segments. This simply updates the information, it doesn't implement any reprojection of the point data. When GeoCue writes the LAS file it will set the "WKT" bit to "True", and format the CRS information as Open Geospatial Consortium (OGC) well known text (WKT). The actual WKT information is written to the specified VLRs in the public header block as per the LAS specification.

Recommended TerraScan Workflow for LAS v1.4

Because of the current LAS version restrictions within the TerraScan software, and the limitations on backward conversions from LAS v1.4 in GeoCue, we recommend you implement the following workflow in GeoCue when needing to deliver LAS v1.4 files. Please note it is important to verify with the end user if partial processing in earlier versions of LAS is permitted prior to delivery in v1.4. If processing in v1.4 needs to be strictly enforced throughout the processing, we recommend moving to an LP360 workflow. GeoCue and LP360 are fully LAS 1.4 compliant. The following instructions are only necessary for processing in Terrasolid products since they do not yet support LAS 1.4.

1. Generate your source data in LAS v1.2. Typically this will be v1.2 PDRF 1 (core 20 bytes plus GPS time), although PDRF 3 is also a possibility if RGB values are generated from the source data. GPS time should be Adjusted Standard GPS Time, as required by the USGS, although GPS Week Time is still supported by the LAS format, but not recommended. If the GPS time needs to be converted, for example if you are not the original generator of the source data and it is in GPS week time, create your GeoCue project in adjusted standard GPS time and GeoCue will convert for you on import (you will need to know the date of collection). Also note that for USGS deliveries, the source LAS strips (unclassified flight line data) is a required

- deliverable and has to be in LAS v1.4. This will typically require a second set of source data to be generated from your processing software (LMS, CloudPro, RiProcess) in v1.4.
2. Import your LAS v1.2 source data into your GeoCue project.
 3. Create and populate your working segments/blocks as per normal. These blocks will be in LAS v1.2, as per the source data.
 4. Follow the normal TerraScan workflow for running classification macros, interactive editing and QA/QC review.
 5. To set the “Withheld Flag” correctly, prior to converting to v1.4 do a final reclassification in TerraScan. Add a class offset of 128 for any “Withheld” points. So withheld ground points in class 2 are moved to class 130. The same method can be used if you need to set the “Synthetic” flag (offset by 32), or “Key-point” flag (offset by 64). Note this class-offset method only works if your points are restricted to the base 32 classes (0-31) as specified in the ASPRS LAS standard class table. If your point class table uses classes greater than 31, you will not be able to use this method to generate v1.4 files. This is a limitation of the LAS-to-LAS conversion process.
 6. Apply any final reprojections or datum conversions as per a normal GeoCue workflow.
 7. Use the GeoCue “Set LAS File Version” checklist step utility to up-convert the LAS files to the desired LAS format. Typically this will be converting to LAS v1.4 PDRF 6 (core 30 bytes that includes GPS time), or PDRF 7 if RGB values are being included. This new GeoCue utility step needs to be added to custom checklists via Environment Builder. Note that down-conversion to earlier LAS versions is not supported in GeoCue so you always want to do this up conversion only after you are sure all work in Terrasolid has been completed. Typically you will want to set the options on the conversion dialog to “1.4” update “To 6, 7, or 8 format” and “Input has Flags and 32 Classes”. We strongly recommend turning on GeoCue’s automatic file versioning for the Update LAS File step. This will create a prior LAS version backup of the data file that you can rollback to should there be any problems.
 8. To set the “Overlap Flag” correctly, open each block in LP360 for GeoCue (or full LP360) and use the LP360 Set Flag tool to set the overlap flag on the appropriate points. Typically this will be setting the overlap flag on all points in the overlap class.
 9. Export the LAS v1.4 files for delivery to your client using the Export Files checklist step.
 10. For rework, or additional editing of the final tiles, add an LP360 for GeoCue step to your workflow. LP360 for GeoCue handles LAS v1.4 directly, and can be used for all interactive rework, or additional editing without requiring any format conversions.

Tip: If at any point you are uncertain which blocks/tile have been converted to LAS v1.4 in your project, and which have not, you can run a quick query in Entity Manager against the FormatID attribute to isolate the tiles with a specific LAS format and PDRF.

Appendix A – PDRF 1 – PDRF 6 Comparison Table

Item	Format	Size (PDRF 1)	Size (PDRF 6)	Format	Item	Notes
X*	long	4 bytes	4 bytes	long	X*	
Y*	long	4 bytes	4 bytes	long	Y*	
Z*	long	4 bytes	4 bytes	long	Z*	
Intensity	unsigned short	2 bytes	2 bytes	unsigned short	Intensity	
Return Number*	3 bits (0-2)	3 bits	4 bits	4 bits (0-3)	Return Number*	
Number of Returns*	3 bits (3-5)	3 bits	4 bits	4 bits (4-7)	Number of Returns*	
			4 bits	4 bits (0-3)	Classification Flags	
			2 bits	2 bits (4-5)	Scanner Channel*	
Scan Direction Flag*	1 bit (6)	1 bit	1 bit	1 bit (6)	Scan Direction Flag*	
Edge of Flight Line*	1 bit (7)	1 bit	1 bit	1 bit (7)	Edge of Flight Line*	
Classification*	unsigned char	1 byte	1 byte	unsigned char	Classification*	1
Scan Angle Rank*	char	1 byte	1 byte	unsigned char	User Data	2
User Data	unsigned char	1 byte	2 bytes	short	Scan Angle*	3
Point Source ID*	unsigned short	2 bytes	2 bytes	unsigned short	Point Source ID*	
GPS Time*	double	8 bytes	8 bytes	double	GPS Time*	4

Notes:

1. In PDRF 1 bits 0-4 of the classification byte are used for class codes 0-31; bits 5-7 are flags for Synthetic, Key-Point and Withheld respectively.
2. Scan Angle Rank is the scan angle rounded to the nearest integer from +90 to -90 with - being left of 0 (nadir) looking forward.
3. Scan Angle is a rotational position in 0.006 degree increments with values from +30,000 to -30,000 covering +180 degrees to - 180 degrees (0 is at nadir/down, - is left/clockwise of nadir looking forward).
4. Note that PDRF 1 simply adds GPS time to PDRF 0.
5. * Required

[Notes]

For information on this CueTip, contact:

GeoCue Group Support
GeoCue Group, Inc.
9668 Madison Blvd., Suite 202
Madison, AL 35758
support@geocue.com
+1-256-461-8289

Find additional information and
participate in our GeoCue Group user
forums.

<http://support.geocue.com>