Software Specification

SL3 File Format Specification

Rev 01

Project Code	Project Name	Author	Doc. Status
1988	NOS	Steve Parker	DRAFT

Table of Contents

1. Document History

Date	Rev	Author	Reason
22nd Aug 2018	01	S Parker	Initial revision

2. Introduction

The Sonar Log File (SLG) file format was created 10 years ago, had a limited amount of data for debugging, and was restricted to a single channel of data. The data records needed to be fixed length, which enabled random access as required by the tools.

A new format was needed to support multiple channels of data (for sidescan information) which had different data rates and packet sizes. The second version of the Sonar Log File Format (SL2) file was created to fill this need. This format needed to be able to be record sequentially and allow the recording to be interrupted at anytime without corrupting the file. The SL2 file format also added features that enabled random access to the data packets whilst maintaining the variable packet size.

The SL2 format achieved random access support by aligning all packets to a 4 byte boundary and including a header for which a program scans. In addition, each packet contains a sequence number for the data in that channel so that a reader can essentially binary search a file for a desired record.

The third version of the Sonar Log File (SL3) was created to support the Structure Scan 3D, and Forward scan channels.

2.1. Purpose

The purpose of this document is to describe the SL3 file format structure in a programming language independent manner.

2.2. Scope

This document describes the SL3 file format structure including the minor revisions 3.0, 3.1 and 3.2.

2.3. Abbreviations and Acronyms

SLG	**S**onar **L**o**G;** sonar log file format versions 1.xx
SL2	**S**onar **L**og v**2;** sonar log file format versions 2.xx
SL3	**S**onar **L**og v**3;** sonar log file format versions 3.xx
SS3D	**S**tructure **S**can **3D**

2.4. References

[1] A.Coleman, *Design Report: SL2 file format,* Rev 01, Navico, 2 Nov 2009

[2] A.Coleman, *Design Report: SLG File Format*, Rev 01, Navico, 25 Jan 2010

3. SL3 File Format Structure

3.1. File naming conventions

The Sonar Log Files use the filename extension to identify the file major version. The file extension convention used is given in Table 1. The filename extension should NOT be relied upon for decoding the file contents, as the full version number (MAJOR.MINOR) is available the file header.

Table 1 - Sonar Log Filename Extension Convention

Major Version	File Extension
1	.slg or .SLG
2	.sl2 or .SL2
3	.sl3 or .SL3

3.2. Top Level File Structure

The SL3 file consists of a **SLGFileHeader** structure followed by a sequence of SL3**DataPacket** structures as show below in Table 2.

Table 2 - Top Level SL3 File Structure

Field Name	Field Type	Description
FileHeader	**SLGFileHeader**	Top level file header
DataPacket #1	**Sl3DataPacket**	Data packet for a single column of a single channel
DataPacket #2	**SL3DataPacket**	Data packet for a single column of a single channel
…	**…**	
DataPacket #N	**SL3DataPacket**	Data packet for a single column of a single channel

The **SLGFileHeader** identifies the file as a Sonar Log and contains the version information as show below in Table 3.

Table 3 - SLGFileHeader structure

Field Name	Field Type	Description
Major	**INT16**	Version major number. Set to 3 for SL3 files
Minor	**INT16**	Version minor number. 0, 1, or 2
BytesPerSounding	**INT16**	Maximum number of range cells per packet. Typ value 3200
Flags	**UINT16**	Bit 0: Debug Enabled. **DIGITAL_DEPTH** and **NOISE_WINDOW** channels are enabled.

3.3. SL3DataPacket Structure

The **SL3DataPacket** contains a single column of range cell data for a single sonar channel. This corresponds to a single ping of data.

The **SL3DataPacket** consists of a **SL3PacketHeader**, optional **LastChannelAddress** data followed by the **ChannelData** structure as show in Table 4.

Table 4 - SL3DataPacket Structure

Field Name	Field Type	Description
PacketHeader	**SL3PacketHeader**	Description of the contents of this packet
LastChannelAddress	Array of **UINT32**	Array length determined by *NumChannels* field in the *PacketHeader*.[1] Offset in bytes from the start of the file of the last packet for each channel. Used for backwards scanning through the file.
Data	**ChannelData**	Sonar data.
Padding	**N x UINT8**	Padding to align the next **SL3DataPacket** to the next 4 byte boundary.

3.3.1. Sl3PacketHeader

The **SL3PacketHeader** contains the fields given in Table 5. The content of each of the fields is described in the subsequent Tables.

Table 5 - SL3PacketHeader Structure

Field Name	Field Type	Description
PacketSubHeader	**SL3PacketSubHeader**	See Table 6
SoundingSetup	**SL3SoundingSetup**	See Table 8
ColumnInformation	**SL3ColumnInformation**	See Table 9
FishID	**SL3FishID**	See Table 10
DigitalInformation	**SL3DigitalInformation**	See Table 11
SounderSetup	**SL3SounderSetup**	See Table 12

Table 6 - SL3PacketSubHeader Structure

Field Name	Field Type	Description
ThisFileAddress	**UINT32**	Offset in bytes from the start of the file of this data packet
NumChannels	**UNIT32**	Number of supported channels,and the length of the *LastChannelAddress* array. 9 for V3.0 10 for V3.1, and V3.2
PacketSize	**UINT16**	Size of this packet in bytes
PreviousPacketSize	**UINT16**	Size of the previous packet in bytes
Channel	**UINT16**	Channel type. See Table 7 for a list of the available channel types
SequenceNumber	**UINT32**	
Padding	**2 x UINT8**	Padding to align the next field to the next 4 byte boundary

[1] *LastChannelAdrress* field not present if channel is of type **DIGITAL_DEPTH** or **NOISE_WINDOW**

Table 7 – Channel Types

Channel Type Name	Value	Description
PRIMARY_SONAR	0	Primary sonar
SECONDARY_SONAR	1	Secondary sonar
DOWNSCAN	2	Downscan
LEFT_SIDESCAN	3	Left sidescan
RIGHT_SIDESCAN	4	Right sidescan
SIDESCAN	5	Combined Left + Right sidescan
FORWARD_SCAN	6	Forward scan
DIGITAL_DEPTH	7	Digital Depth ping
NOISE_WINDOW	8	Noise Window data
STRUCTURE_SCAN_3D	9	Structure Scan 3D

Table 8 – SL3SoundingSetup Structure

Field Name	Field Type	Description
UpperLimit	**FLOAT32**	Chart lower limit in feet
LowerLimit	**FLOAT32**	Chart upper limit in feet
BurstLength	**UINT16**	Ping burst length in usec
Integration	**UINT8**	Processing integration in cycles
Gain	**UINT8**	Receiver gain in dB
VideoIntegration	**UINT8**	Legacy value – not used
Frequency	**UINT8**	Nominal ping frequency in kHz
PingPeriod	**UINT16**	Ping period in ms
Correlation	**UINT8**	Legacy value – not used
Discrimination	**UINT8**	Noise rejection (or Sonar ASP) setting: 0 – Off 1 – Low 2 – Medium 3 – High
Flags	**UINT16**	Bit 0: Gain boost enabled.
RunningTime	**UINT32**	Running time in ms from the start of recording. See section 0

Table 9 – SL3ColumnInformation Structure

Field Name	Field Type	Description
NumRangeCells	**UINT32**	Number of range cells (or bytes) in the corresponding **ChannelData**
Depth	**FLOAT32**	Depth below transducer in feet
KeelOffset	**FLOAT32**	Keel offset in feet
AutoSenseRangeCell	**UINT8**	Amplitude of the autosense value in range cell units
DigitalRangeCell	**UINT8**	Amplitude of the bottom echo in range cell units
NoiseCount	**UINT8**	Legacy value – not used
NoisePeak	**UINT8**	Legacy value – not used
NoiseAverage	**UINT8**	Legacy value – not used
RVGMaxAttenuation	**UINT8**	Legacy value – not used
RVGDepth	**UINT16**	Legacy value – not used

Table 10 – SL3FishID Structure

Field Name	Field Type	Description
FishDepths	**FLOAT32 x 4**	Array of fish depths in feet
FishSizes	**UINT8 x 4**	Array of fish size in the range 0 – 6,,with 6 being the largest

Table 11 - SL3DigitalInformation Structure

Field Name	Field Type	Description
GroundSpeed	**FLOAT32**	Speed over ground in knots
WaterTemperature	**FLOAT32**	Water temperature in °C
X Position	**INT32**	X value of the Cartesian position in metres using Mercator projection
Y Position	**INT32**	Y value of the Cartesian position in metres using Mercator projection
WaterSpeed	**FLOAT32**	Legacy value – not used
Track	**FLOAT32**	Course in radians
Altitude	**FLOAT32**	Altitude in feet
Heading	**FLOAT32**	Heading in radians
ValidFlags	**UINT32**	Bit field to indicate the validity of each of the above fields: Bit 1: *GroundSpeed* valid Bit 2: *WaterTemperature* valid Bit 4*: X/Y Position* valid Bit 6: *WaterSpeed* valid Bit 7: *Track* valid Bit 8: *Heading* valid Bit 9: *Altitude* valid

Table 12 – SL3SounderSetup Structure

Field Name	Field Type	Description
Flags	**UINT8**	Bit field: Bit 0: *SounderSetup* data is valid Bit 1: Manual range mode enabled
FishingMode	**UINT8**	Fishing mode index
PingSpeed	**INT8**	Ping speed index for Panel #1
NoiseRejection	**UINT8**	Noise Rejection setting
MillisecondOffset	**UINT32**	Timestamp offset from the first timestamp in ms

3.3.1.1. Timestamp Management

The timestamp for a data packet is constructed from a *FirstTimeStamp* value in seconds and a high precision offset value in milliseconds, as follows:

TimeStamp (sec) = FirstTimeStamp (sec) + MillisecondOffset (ms) / 1000

The method of obtaining the FirstTimeStamp value differs between the different minor revisions of the SL3 file format. In all versions the *MilllisecondOffset* value is obtained from the corresponding **SL3SounderSetup** data.

In V3.0 and V3.1 of the SL3 file format the FirstTimeStamp value is obtained from the *RunningTime* field very first **SL3DataPacket**. The timestamping in these versions is only approximate (error of ~ +/- 500 ms) and is done at the time of recording each data packet.

In V3.2 of the SL3 file format the channels are permitted to have different time bases to allow for high precision timestamping of locally recorded channels, and approximate timestamping of networked channels. In this case, the *FirstTimeStamp* value is obtained from the first **SL3DataPacket** for each channel. For locally recorded channels, the timestamping is done at the point of data acquisition and has a much-reduced error of ~+/- 20 ms.

3.4. ChannelData Structure

The **ChannelData** structure contains the sonar information for one ping. The format varies according to the channel type as given in Table 13. Each of these data structures are described in the subsequent sections. All of these channel data structures obtain their size information from the *NumRangeCells* field in the corresponding **SL3ColumnInformation** data.

Table 13 – ChannelData Type for each ChannelType

Channel Type Name	ChannelData Type	Typical Size
PRIMARY_SONAR	**SonarChannelData**, see section 3.4.1	3072 range cells
SECONDARY_SONAR	**SonarChannelData**, see section 3.4.1	3072 range cells
DOWNSCAN	**SonarChannelData**, see section 3.4.1	1400 range cells
LEFT_SIDESCAN	**SonarChannelData**, see section 3.4.1	1400 range cells
RIGHT_SIDESCAN	**SonarChannelData**, see section 3.4.1	1400 range cells
SIDESCAN	**SideScanChannelData**, see section 3.4.3	2800 range cells
FORWARD_SCAN	**ForwardScanChannelData**, see section 3.4.4	Variable up to a max of 3000 raw point samples, 384 line data points, and 256 noise window samples
DIGITAL_DEPTH	**SonarChannelData**, see section 3.4.1	2000 range cells
NOISE_WINDOW	**NoiseWindowChannelData**, see section 3.4.2	256 ADC samples
STRUCTURE_SCAN_3D	**SS3DChannelData**, see section 3.4.5	Variable up to a max of 3000 raw point samples, 384 line data points, 256 noise window samples per side.

3.4.1. SonarChannelData Structure

The **SonarChannelData** structure is the simplest of all the ChannelData structures as it contains only an array of range cells as show in Table 14. The number of range cells is given by the corresponding *NumRangeCells* field in the **SL3ColumnInformation** data.

Table 14 – SonarChannelData Structure

Field Name	Field Type	Description
RangeCell #1	**UINT8**	Range cell amplitude in ~ dB * 255 / 140
RangeCell #2	**UINT8**	Range cell amplitude in ~ dB * 255 / 140
…	**...**	…
RangeCell #N	**UINT8**	Range cell amplitude in ~ dB * 255 / 140

3.4.2. NoiseWindowChannelData Structure

The **NoiseWindowChannelData** structure contains a small set of raw ADC data samples just prior to the transmission of the ping. This is used to analyse the background noise and optimise the ping characteristics for best performance. The number of bytes is given by the corresponding *NumRangeCells* field in the **SL3ColumnInformation** data. The number of samples is half the number of bytes.

Table 15 – NoiseWindowChannelData Structure

Field Name	Field Type	Description
Sample #1	**UINT16**	Raw 12 bit ADC sample
Sample #2	**UINT16**	Raw 12 bit ADC sample
…	**…**	…
Sample #N	**UINT16**	Raw 12 bit ADC sample

3.4.3. SideScanChannelData

The **SideScanWindowChannelData** structure contains the range cell data from both the Left and Right side scan channels as shown in Table 16. The total number of range cells is given by the corresponding *NumRangeCells* field in the **SL3ColumnInformation** data.

Table 16 – SideScanWindowChannelData Structure

Field Name	Field Type	Description
LeftRangeCell #1	**UINT8**	Range cell amplitude in ~ dB * 255 / 140
LeftRangeCell #2	**UINT8**	Range cell amplitude in ~ dB * 255 / 140
…	**…**	…
LeftRangeCell #N	**UINT8**	Range cell amplitude in ~ dB * 255 / 140
RightRangeCell #1	**UINT8**	Range cell amplitude in ~ dB * 255 / 140
RightRangeCell #1	**UINT8**	Range cell amplitude in ~ dB * 255 / 140
…	**…**	…
RightRangeCell #1	**UINT8**	Range cell amplitude in ~ dB * 255 / 140

3.4.4. ForwardScanChannelData Structure

The **ForwardScanChannelData** contains both the raw point data generated by the Forward Scan sensor and the processed Forward Scan data in the form of line data (in Cartesian coordinates) representing the calculated depth values. The content of the **ForwardScanChannelData** is given in Table 18. The number of values in the *LineData* fields is given by *NumRangeCells* / 8 in the corresponding **SL3ColumnInformation** data.

Table 17 – ForwardScanDataChannel Structure

Field Name	Field Type	Description
LineData	**LineDataArray**	Calculated bottom values. See Table 18
PointData	**PointData**	Raw point data from the sensor. See Table 19

Table 18 – LineDataArray Structure

Field Name	Field Type	Description
XLineData #1	**FLOAT32**	Forward/Side range in feet of the calculated bottom
YLineData #1	**FLOAT32**	Depth in feet of the calculated bottom
XLinetData #2	**FLOAT32**	Forward/Side range in feet of the calculated bottom
YLineData #2	**FLOAT32**	Depth in feet of the calculated bottom
…	**…**	…
XLineData #N	**FLOAT32**	Forward/Side range in feet of the calculated bottom
YLineData #N	**FLOAT32**	Depth in feet of the calculated bottom

Table 19 - PointData structure

Field Name		**Field Type**	**Description**
CSMForwardHeader	*SequenceNumber*	**UINT16**	Sequence number
	NoiseWindow	**UINT16 x 256**	Array of raw 12 bit ADC samples
	NumSamples	**UINT16**	Number of samples in the *Points* array
	MessageMode	**UINT8**	*Points* data format. Currently set to 2
Points	*Angle #1*	**INT16**	Angle in degrees x 512 from the horizontal sensor plane
	Range Index #1	**UINT16**	Range index from the sensor
	Amplitude #1	**UINT8**	Amplitude in a logarithmic scale
	Angle #2	**INT16**	Angle in degrees x 512 from the horizontal sensor plane
	Range Index #2	**UINT16**	Range index from the sensor
	Amplitude #2	**UINT8**	Amplitude in a logarithmic scale
	…	…	…
	Angle #N	**INT16**	Angle in degrees x 512 from the horizontal sensor plane
	Range Index #N	**UINT16**	Range index from the sensor
	Amplitude #N	**UINT8**	Amplitude in a logarithmic scale

The *RangeIndex* values can be converted to a range value as follows:

*Range (feet) = RangeIndex x C / (2 * Fs)*

Where C = sound speed in feet/sec (4800 ft/s)

Fs = Sample rate of the sensor in Hz

(102860 Hz for **FORWARD_SCAN**, and 213333.3 Hz for **STRUCTURE_SCAN_3D**)

3.4.5. SS3DChannelData Structure

The **SS3DChannelData** contains both the raw point data generated by the SS3D sensor and the processed SS3D data in the form of line data (in Cartesian coordinates) representing the calculated bottom points. The content of the **SS3DChannelData** is given in Table 20.

Table 20 – ForwardScanDataChannel Structure

Field Name	**Field Type**	**Description**
Header	**SL3SideScanHeader**	SS3D data header. See Table 21
LeftLineData	**LineDataArray**	Calculated bottom from the LHS sensor. See Table 18
RightLineData	**LineDataArray**	Calculated bottom from the RHS sensor. See Table 18
LeftPointData	**PointData**	Raw point data from the LHS sensor. See Table 19
RightPointData	**PointData**	Raw point data from the RHS sensor. See Table 19

Table 21 – Sl3SideScanHeader Structure

Field Name	**Field Type**	**Description**
HeaderSize	**UINT32**	Size of this header in bytes
LeftLineSize	**UINT32**	Size of the *LeftLineData* in bytes
RightLineSize	**UINT32**	Size of the *RightLineData* in bytes
TotalPointDataSize	**UINT32**	Total number of bytes in the *Left3DData* and *Right3DData*
LeftPointDataSize	**UINT32**	Size of the *LeftPointData* in bytes
RightPointDataSize	**UINT32**	Size of the *RightPointData* in bytes
Spare1	**UINT32 x 4**	Spare values
AngleFromSurface	**FLOAT32**	Correction angle in degrees
LeftDepth	**FLOAT32**	Left side depth in feet
RightDepth	**FLOAT32**	Right side depth in feet
BottomDepth	**FLOAT32**	Composite depth from left/right in feet
DepthOffset	**FLOAT32**	Calibration offset in feet
Spare2	**FLOAT32 x 4**	Spare values