SDF/SDFX Data Page Definitions Specification

15300018

Rev 3.16

27 July 2012

communications
Klein Associates, Inc.

Revision History

Revision	Date	Author	Description
1.00		C. Chase	Initial Release.
1.01		C. Chase	Changed System 3000 SBP to short from unsigned short.
1.02		C. Chase	Added raw data fields
1.03		C. Chase	Revised raw data fields, added raw data config word.
1.04	3/27/2002	C. Chase	Version 3 Header Release
2.00	3/06/2007	K. Garner	Version 4 Header Release
2.01	5/18/2007	K. Garner	Add wing angle of attack and emergency switch state
2.02	6/6/2007	K. Garner	Add layback method, layback fish position, pressure sensor offset, and magnetic heading offset.
2.03	6/14/2007	K. Garner	Fix size in reserved3 field in Type3 header.
2.04	11/05/2007	K. Garner	Documentation change only. Add notes about the ping marker that is in the file when saved by SonarPro®. Add tpuSwVersion and capabilityMask to SDF header.
2.05	04/15/2008	D. Clapp / K. Garner	Added flag for txVersion, and numSamplesExtra for MF processing. Added system capabilities bit to indicate towfish is a 5000 V2.
2.06	2/13/2009	K. Garner	Added new page versions for the 5000 in order to support Bathymetry. Added motionSensorType to header to support Klein Motion Sensor. Added time reference counts to header to support TPU versions that support h/w time stamping of events. Added ping interval to header. Added more detail on pageVersion behavior
2.07	6/02/2009	K. Garner	Note: Changes for vxWorks 8.X. This is to integrate the POS MV. 2.06 was never released. Perhaps it should be to capture vxWorks 7.X which went to Gesma and OYO. For now, start a new version to see where we are going from here. Add motionSensorType value of 2 to account for POS MV. Update header capabilityMask - Added a bit indicating system is hull mounted - Added a bit indicating system configured for input from array sound speed sensor Add secondsOfWeek to header to support POS MV. Added speedSoundSource to header.
3.00	11/4/2009	K. Garner	Reformat entire document. Add information on each sonar type's use of the data page header. Added SDFX. Remove all 7000 references. Add 5900 Data Page Types.
3.01	1/16/2010	K. Garner	Add Pressure sensor fields to header. Clarify which systems use the 5000/5001 page type. Clean up document in preparation of release.

3.02	3/1/2010	K. Garner	Correct errors in SDFX structrures: - POS MV Group 1 - Sound Velocity Profile 1
3.03	3/2/2010	K. Garner	Add detailed definition of Bathy Calibration 1 record.
3.04	3/5/10	K. Garner	- Add new section on interpreting Klein Bathymetric data. - Update depth computation based on new pressure configuration values in header. - Add additional information on bathymetry output data in each data page type.
3.05	7/23/2010	K. Garner	- Add new SDFX data record for Octopus F180 Attitude and Positioning System
3.06	10/28/2010	K. Garner	- Add new SDFX structure type for TSS DMS-XX motion sensor - Add bit definitions to header errorFlags field
3.10	4/7/2010	K. Garner	Update to 5911 page
3.11	2/17/2011	K. Garner	- Add UUV-3500 page - Add UUV-3500 3511 page for processed bathy - Redefine page type 5910 as a Processed ping w/o motion compensation - Add page type 5920 as a Processed ping with motion compensation - Add processedPingSpacing to header - Add ambientTemperature to header - Add saturationDetectThreshold to header - Add definition to header resMode for 5900 systems - Add postProcessVersion bit definitions for 5900 - 5900/UUV-3500 documentation clean up
3.12	5/20/2011	K. Garner	- Add Bathy Output page for NOAA LRSSS (System 7180 engine) towfish - Add SDFX structure definitions for NOAA LRSSS (System 7180 engine) towfish - Fix description of header capabilityMask bit definitions - Add UUV-3500 capability bit to header capabilityMask - Clarify header speedFilterSwitch interpretation - Clarify header configuration bits
3.12A	6/7/2011	K. Garner	- Indicate document valid for LRSSS distribution only. This version of the specification is to be used only for LRSSS and not other Klein sonar systems. This is due to some of the specifications for non-LRSSS sonars are preliminary.
3.13	7/14/2011	K. Garner	- Separate UUV-3500 page types for low frequency and high frequency. - Add sonarFreq to header
3.14	10/25/2011	K. Garner	- Add new SDFX structure type for UUV-3500 diagnostic results - Modify comment on bathy angle vectors to indicate that angle is computed using the SDF header speedSound value (rather than 1500 m/s). This change was made effective in SonarPro 12.1. - Add postProcessVersion bit 12 definition (sound velocity)

3.15	12/27/2011	K. Garner	- UUV-3500 V1.10 of sonar software supports header value temperatureAmbient - Add additional clarification to LRSSS pageVersion 7191 on which vectors are populated and which are reserved placeholders. - Remove “preliminary do not distribute” header
3.16	2/08/2012	MLF	- Add bathy output arrays for signal-to-noise ratio and uncertainty. - Add postProcessVersion bits for “sound speed from sound velocity profile” and “bathy intensity, s/n ratio, uncertainty present.” - Add Version 3 bathy process settings structure definition. - Add OceanServer IMU SDF extension record.

Table of Contents

1. Introduction

This document describes the SDF and SDFX data page formats for the System 3000 (NGS), System 5000 and System 5900.

2. Overview

The main data component of a Klein Sonar is the data page format. These data page formats are used to output the Sonar data from the TPU (via Ethernet) and as a data file format for output from SonarPro® (*.sdf files).

The Klein data page format name has traditionally been called SDF. This document defines the SDF format as well as an extension to the SDF format called SDFX. The SDFX format was defined in 2009 as a way to extend the SDF format while staying backwards compatible with the large amount of existing SDF files and SDF file readers. A legacy SDF file reader, if properly constructed, can read all the traditional SDF information from an SDFX file. The legacy reader simply skips over the new SDFX data. Obviously, the legacy reader would require an update to read any new data in the SDFX portion of the file or data page.

2.1. *SDF Data Page Layout*

The generic SDF data page consists of a data page header followed by a variable number of data page channels (Figure 1). The data page header is the same for all sonar types. (Note: header fields may be interpreted differently depending on the sonar type.). The data page header contains a field called "pageVersion" that determines the layout of the data page channels.

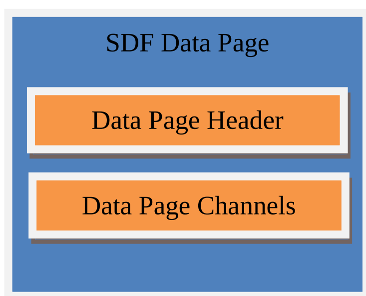


Figure 1 - SDF Data Page Layout

2.2. *SDFX Data Page Layout*

The SDFX format adds a variable sized “extension” to the end of the SDF data page (Figure 2). The Data Page Header indicates whether the data page extension is present or not via the sdfExtensionSize field.

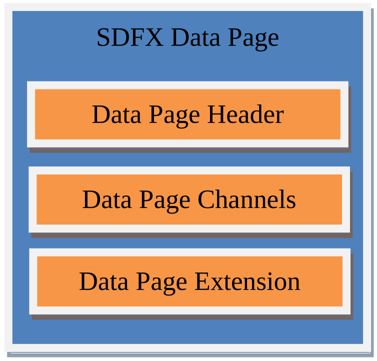


Figure 2 - SDFX Data Page Layout

3. Description

Each system has a unique data page structure as defined by the pageVersion field in the header. The header is ostensibly the same for each page structure but the data portion of a data page is unique. The pageVersion field can have the following values:

Table 1 – Valid pageVersion values

Towfish Type	Header version	pageVersion value
3000	3	3000
3000	4	3001
UUV 3500 sonar data(Obsolete)	4	3500
UUV 3500 sonar data Low Frequency	4	3501
UUV 3500 sonar data High Frequency	4	3502
UUV-3500 Bathy Pulse Compressed Data Page (4-channel)	4	5903
UUV-3500 processed bathy data	4	3511
5000	3	5000
5000	4	5001
5000 Bathy	4	5002
Reserved. Note: This was used as a prototype of the first 15 channel V2 system.	4	5003
5000 V2 Bathy w/ 3 dedicated bathy channels	4	5004
5900 Side Scan and Bathy Raw Data Page (64-Channel)	4	5900
S5900 Side Scan and Bathy Pulse Compressed Data Page (64-channel)	4	5901
S5900 Side Scan QC Beamformed page	4	5902
S5900 Bathy Pulse Compressed Data Page (For Side Scan Sonar) (4-channel)	4	5903
5900 Gap Filler and Bathy Raw Data Page (64-Channel)	4	5905
S5900 Gap Filler and Bathy Pulse Compressed Data Page (64-channel)	4	5906
5900 Side Scan Processed Data Page (no motion compensation)	4	5910
5900 Bathy Processed (For Side Scan Sonar) Data Page	4	5911
5900 Gap Filler Processed Data Page	4	5915
5900 Bathy Processed (For Gap Filler Sonar) Data Page	4	5916
5900 Side Scan Processed Data Page w/ Motion Compensation	4	5920
7180 Bathy Processed Data Page	4	7191

See section 0 for additional information on the SDF pageVersion.

The data page structure is the data page produced by the System's Transceiver and Processing Unit (TPU). See the section 4 for how these individual data pages are concatenated into ".sdf" files.

3.1. *Data Page Header*

The data page header is the same for all sonar types. The interpretation of individual fields may be different for the various sonars. This is particularly true for the header fields that represent a sonar setting, e.g., a transmit waveform value of 1 will not indicate the same pulse length for a 3000 system as compared to a 5000 system. Table 2 defines the data page header fields, definitions, and the usage by the different sonar systems and applications. For each sonar system/application column, a value of “X” or a more detailed explanation of a field’s use indicates it is used. A value of “NA” that it is not applicable (not used) to that system/application. Unused values default to 0.

Table 2 - Data Page Header Definition

Field #	Field	Definition	3000	3900	5000	5900	3500
1	U32 numberBytes	total number of bytes in page	X	X	X	X	X
2	U32 pageVersion	data page structure	X (Note 1)	X (Note 1)	X (Note 1)	X (Note 1)	X
3	U32 configuration	Bit field indicates which channels or beams are populated in the data page channels.	X (Note 2)	X (Note 2)	X (Note 2)	N/A	X (Note 2)
4	U32 pingNumber	increments by 1 for each sonar ping.	X	X	X	X	X
5	U32 numSamples	count of samples in processed side-scan data, if processed data exists in page, or count of samples in rawdata channels if no processed data exists. The difference is stored in numSamplesExtra (type3 header).	X	X	X	X	X
6	U32 beamsToDisplay	Bit field indicates which side scan channels should be used to form the raster image.	N/A	N/A	X (Note 3)	N/A	N/A
7	U32 errorFlags	Bit field: Bit 0: Invalid speed Bit 1: GPS data error Bit 2: Telemetry error Bit 3: Sensor Checksum Error Bit 4: Towfish Leak Bit 5: Bad Data Bit 6: Watchdog Bit 7: Compass Error Bit 8: No GPS Lat/Lon sentence input Bit 9: No GPS speed sentence input Bit 10: No GPS ZDA sentence input Bit 11: No data from Motion Reference Unit or MRU error Bit 12: No 1 PPS input Not all bits are valid for all sonar configurations. For example, towfish must be equipped with a leak sensor to set the towfish leak flag.	X	X	X	X	X
8	U32 range	m	X	X	X	X	X

Field #	Field	Definition	3000	3900	5000	5900	3500
9	U32 speedFish	cm/s. If manualSpeedSwitch set, this value was entered manually. Otherwise, calculated from speed source (generally GPS). If the speedFilterSwitch field is set, the value is filtered with a 90/10 low pass filter.	X	X	X (Note 14)	X	X
10	U32 speedSound	speed of sound at the transducer arrays from a specified source. Sound Speed value (cm/s)	X	X	X	X	X
11	U32 resMode	System 5000 page versions: 0 = Normal (20 cm) 1 = High Resolution (10 cm) System 5900 pageVersion = 5002: QC Beamformer Resolution: 0 = “W0” (0.5m resolution) 1 = “W1” (1m resolution) System 5900 pageVersion = 5910,5920 0 = Resolution 0 1 = Resolution 1	N/A	N/A	X	X	N/A
12	U32 txWaveform	Transmit waveform number (Note 17)	X	X	X	X	X
13	U32 respDiv	Responder divisor	X	X	X	X	X
14	U32 respFreq	Responder frequency enum	X	X	X	X	X
15	U32 manualSpeedSwitch	0 = speedFish is from speed sensor 1 = speedFish is manual value from master	X	X	X	X	X
16	U32 despeckleSwitch	0 = Despeckling off 1 = Low 2 = Medium 3 = High	X	X	X (Note 4)	N/A	N/A
17	U32 speedFilterSwitch	Bit 0: 0 = speed filter is off 1 = speed filter is on Bits 31-1: Reserved	X	X	X	X	X
18	U32 year	TPU Time of ping, Calendar Year (includes century) (Note 15)	X	X	X	X	X
19	U32 month	TPU Time of ping, Calendar month (1-12) (Note 15)	X	X	X	X	X
20	U32 day	TPU Time of ping, Calendar day (1-31) (Note 15)	X	X	X	X	X
21	U32 hour	TPU Time of ping, hour (0-23) (Note 15)	X	X	X	X	X
22	U32 minute	TPU Time of ping, minute (0-59) (Note 15)	X	X	X	X	X

Field #	Field	Definition	3000	3900	5000	5900	3500
23	U32 second	TPU Time of ping, second (0-59) (Note 15)	X	X	X	X	X
24	U32 hSecond	TPU Time of ping, hundredths of second (1-99) (Note 15)	X	X	X	X	X
25	U32 fixTimeHour	Time of last serial NMEA message, hour	X	X	X	X	X
26	U32 fixTimeMinute	Time of last serial NMEA message, minute	X	X	X	X	X
27	float fixTimeSecond	Time of last serial NMEA message, second	X	X	X	X	X
28	float heading	heading from towfish compass (deg.) (Note 18)	X	X	X	X	X
29	float pitch	pitch from compass (deg.)	X	X	X	X	X
30	float roll	roll from compass (deg.)	X	X	X	X	X
31	float depth	from towfish (Volts) (Note 5)	X	X	X	X	X
32	float altitude	from towfish (meters)	X	X	X	X	X
33	float temperature	from towfish (Degrees C)	X	X	X	X	X
34	float speed	from serial NMEA, updated on GPS update, m/s	X	X	X	X	X
35	float shipHeading	from serial NMEA – Course Over Ground, Degrees	X	X	X	X	X
36	float magneticVariation	from serial NMEA $--RMC message,	X	X	X	X	X
37	double shipLat	from serial NMEA, radians	X	X	X	X	X
38	double shipLon	from serial NMEA, radians	X	X	X	X	X
39	double fishLat	from serial NMEA, radians (Note 6)	X	X	X	X	X
40	double fishLon	from serial NMEA, radians (Note 6)	X	X	X	X	X
	Added at version 3 (Note 13)						
41	U32 tvgPage	System 5000: TVG page number System 3000: Bits 7- 0: Low Frequency TVG page number Bits 15 – 8: High Frequency TVG page number System UUV-3500 0 = Low Gain 1 = High Gain System 5900: Bits 7-0: Receiver Side Scan Gain number (Note 16) Bits 15-8: Receiver Bathymetry Gain number (Note 16)	X	X	X	X	X
42	U32 headerSize	number of bytes in header	X	X	X	X	X

Field #	Field	Definition	3000	3900	5000	5900	3500
43	U32 fixTimeYear	Time of last serial NMEA message, year	X	X	X	X	X
44	U32 fixTimeMonth	Time of last serial NMEA message, month	X	X	X	X	X
45	U32 fixTimeDay	Time of last serial NMEA message, day	X	X	X	X	X
46	float auxPitch	aux data from AUV or other sensors (Note 7). Units are sensor or AUV specific.	X	X	X	X	X
47	float auxRoll		X	X	X	X	X
48	float auxDepth		X	X	X	X	X
49	float auxAlt		X	X	X	X	X
50	float cableOut	m (Note 8)	X	X	X	X	X
51	float fseconds	TPU Time of ping, fractional seconds (seconds) (Note 15)	X	X	X	X	X
52	U32 altimeter	altimeter off/on 0 = off 1 = on	X	X	X	X	X
53	U32 sampleFreq	Hz	X	X	X	X	X
54	U32 depressorType	towfish wing type enum (Note 8)	X	X	X	X	X
55	U32 cableType	towfish cable type enum (Note 8)	X	X	X	X	X
56	F32 shieveXoff	X winch offset from datum (m) (Note 8)	X	X	X	X	X
57	F32 shieveYoff	Y winch offset from datum (m) (Note 8)	X	X	X	X	X
58	F32 shieveZoff	Vertical winch offset from datum (m) (Note 8)	X	X	X	X	X
59	F32 GPSheight	Vertical GPS Offset from datum (m) (Note 8)	X	X	X	X	X

Field #	Field	Definition	3000	3900	5000	5900	3500
60	U32 rawDataConfig	System 5000: Bit field indicates which raw data channels are populated in the data page. Bits 15-0: Port raw data config, bit 0 is channel 0, bit 1 is channel 1, etc. Bits 31-16: Stbd raw data config, bit 16 is channel 0, bit 17 is channel 1, etc. System UUV-3500, Hydroscan: 0 = standard operation 1 = factory test mode	N/A	N/A	X	N/A	X
	Added at version 4 (Note 13)						
	Size of this header addition should stay 256 bytes (64 U32s)		X	X	X	X	X
61	U32 header3ExtensionSize	Size of only this header extension. Must be equal to 256 bytes.	X	X	X	X	X
62	U32 sbpTxWaveform	Tx waveform for Sub Bottom Profiler (SBP)	X	N/A	N/A	N/A	N/A
63	U32 sbpPreAmpGain	0 = low 1 = high	X	N/A	N/A	N/A	N/A
64	U32 sbpDataRaw	0 = Processed SBP data, 1 = Raw SBP data	X	N/A	N/A	N/A	N/A
65	U32 sbpNumSamples	Number of SBP samples in data page. May be different from side scan.	X	N/A	N/A	N/A	N/A
66	U32 sbpSampleFreq	Sample frequency of SBP channel	X	N/A	N/A	N/A	N/A
67	U32 sbpTxWaveformVersion	Tx waveform version of the SBP Tx board.	X	N/A	N/A	N/A	N/A
68	float wingAngle	Angle of actuated wing in degrees	N/A	N/A	X (Note 10)	N/A	N/A
69	U32 emergencySwitchState	State of system emergency switch - 1 = on, 0 = off	N/A	N/A	X (Note 10)	N/A	N/A
	Layback position parameters set by SonarPro. Set to zero by TPU.						
70	U32 laybackMethod	Method used to calculate layback. 0 = Pythagorean theorem (Note 9) 1 = Hull Mount system. Calculated transducer positions from SDFX Ship Config Info.	X	X	X	X	X

Field #	Field	Definition	3000	3900	5000	5900	3500
71	double laybackFishLat	fish latitude as determined by layback calculation (in radians) (Note 9) or latitude of transducer for hull mount system.	X	X	X	X	X
72	double laybackFishLon	fish longitude as determined by layback calculation (in radians) (Note 9) or longitude of transducer for hull mount system.	X	X	X	X	X
73	float fishHeadingOffset	Magnetic heading offset applied to towfish heading, in degrees (Note 11)	X	X	X	X	X
74	float pressureSensorOffset	psi (Note 12)	X	X	X	X	X
	Added at TPU s/w version 6.13						
75	U32 tpuSwVersion	Version of the TPU s/w, 0xVVNNMMDD where VV = Major Version Number NN = Minor Version Number MM = Month DD = Day	X	X	X	X	X
	Added at vxWorks version 6.17						

Field #	Field	Definition	3000	3900	5000	5900	3500
76	U32 capabilityMask	Bit mask defining various system capabilities. * Bit 0 = Configured for raw data (System 5000 only) * Bit 1 = Configured for actuated wing * Bit 2 = Configured with Sub Bottom Profiler option * Bit 3 = Configured with header ver. 4 * Bit 4 = Configured to allow a single oversampled frequency (3000 only) * Bit 5 = Configured to allow dual frequency operation (3000 only) * Added at tpuSwVersion 6.17. * Bit 6 = 5000 (V1 or V2) system with ver. 2 Demux * Bit 7 = 5000 V2 towfish * Added at tpuSwVersion 7.00 * Bit 8 = Configured for external trigger (Slave mode) * Added at tpuSwVersion 8.00 Bit 9 = Configured with hull mount transducers Bit 10 = Configured to accept input from array sound speed sensor. Bit 11 = Pressure sensor parameters are valid in header. Bit 12 = Configured with psig pressure sensor. Otherwise, psia assumed. Bit 13 = 3900 system configured with 1dB TVG steps from -7dB to +7dB for high frequency. Otherwise, configured for 3dB steps from -21dB to +21dB. Bit 14 = Configured as UUV-3500. This is a "3000-like" multi-channel single beam system.	X	X	X	X	X
	Added at vxWorks version V6.19						
77	U32 txVersion	TPU Transmitter Version 0 = 5000, 1 = 5000 V2	N/A	N/A	X	N/A	N/A
	Added at vxWorks version V6.22						
78	U32 numSamplesExtra	The extra number of samples included in each data channel to account for chirp Tx waveforms. This value is zero for other waveforms.	N/A	N/A	X	N/A	X
	Added at vxWorks version V7.00						

Field #	Field	Definition	3000	3900	5000	5900	3500
79	U32 postProcessVersion	The version of software that was used to post-process this data. Set to 0 by TPU as default. * Bits 31-24: Major version number of post-processing software. * Bits 23-16: Minor version number of post-processing software * Bits 15-0: Bit mask which defines the type of post processing performed as follows: Bit 13: Bathymetric processing results include intensity, signal-to-noise ratio, and uncertainty data. Bit 12: Sound velocity value used comes from sound velocity profile at depth Bit 11: Set when 5900 beamformer interpolation mode. Clear when 5900 “5000-like” integral ping mode. Bit only valid for System 5900 when Bit 0 is set. Bit 10: Set when 5900 Motion focus radius override correction applied Bit 9: Set when 5900 Motion steer correction applied Bit 8: Set when 5900 Motion altitude correction applied Bit 7: Set when 5900 Motion Pitch Rate correction applied Bit 6: Set when 5900 Motion Initial Pitch correction applied Bit 5: Set when 5900 Motion Yaw correction applied Bit 4: Set when Bathy V2 processing done with new scale factors Bit 3: Set when Bathy processing included ship geometry (lever arms) correction Bit 2: Set when Bathy processing included sound velocity correction Bit 1: Set for Bathy processing Bit 0: Set for 5000/5900 Beamform processing	N/A	N/A	X	X	X
80	U16 motionSensorType	The type of motion sensor present in the towfish 0 = Standard TCM Compass 1 = KMS-01 Configuration 1 2 = POS MV Version 3 or 4	N/A	N/A	X	X	N/A

Field #	Field	Definition	3000	3900	5000	5900	3500
81	U16 pingTimeRefCount	TPU Reference counter value when ping trigger from fish received	N/A	N/A	X	X	N/A
82	U16 extTrigTimeRefCount	TPU Reference counter value when external trigger received	N/A	N/A	X	X	N/A
83	U16 onePpsTimeRefCount	TPU Reference counter value when 1PPS signal received.	N/A	N/A	X	X	N/A
84	U32 timeRefCountWeight	1 LSB of TPU time reference counter value in nanoseconds	N/A	N/A	X	X	N/A
85	F32 altitudeBathy	The altitude used to compute the bathymetric solution	N/A	N/A	X	X	X
86	F32 pingInterval	The ping interval in seconds	N/A	N/A	X	X	X
	Added at vxWorks version V8.00						
87	U32 sdfExtensionSize	Size (in bytes), of the SDF extension area. If 0, no extension present.	N/A	N/A	X	X	X
88	double secondsOfWeek	The seconds of the week ping trigger time. Useful for comparing to sensors (like the Applanix POS MV) who provide time stamps as seconds of the week. Computed from header year, month, day, hour, minute, fseconds	X	X	X	N/A	N/A
89	U32 speedSoundSource	The source of the header speedSound value: 0 = manual speed of sound 1 = array speed of sound sensor	X	X	X	X	X
90	F32 pressureSensorMax	maximum pressure reading of pressure sensor	X	X	X	X	X
91	F32 pressureSensorVoltageMin	minimum voltage reading from pressure sensor (Volts)	X	X	X	X	X
92	F32 pressureSensorVoltageMax	maximum voltage reading from pressure sensor (Volts)	X	X	X	X	X
93	U32 processedPingNumber	increments by 1 for each processed ping				X	N/A
94	F32 processedPingSpacing	The along track spacing of the processed ping (meters)	N/A	N/A	N/A	X	N/A
95	F32 temperatureAmbient	Towfish internal temperature (Degrees C)	N/A	N/A	N/A	X	X
99	S32 saturationDetectThreshold	Saturation Detect Threshold System 5900/ System UUV-3500: -1 = Off 0 = 0% 1 = 12% 2 = 25% 3 = 38% 4 = 50% 5 = 62% 6 = 75% 7 = 88%	N/A	N/A	N/A	X	X
100	U32 sonarFreq	The operating frequency of the Sonar (kHz)					X

Field #	Field	Definition	3000	3900	5000	5900	3500
	U32 reserved3[26]	Reserved data portion. Array size set to keep this header addition at 256 bytes.	X	X	X	X	

Notes:

1. See Table 1 for valid pageVersion values.
2. The configuration bit field is system type dependent. The System 3000, 3900, and 5000 Mk1 use this field to indicate which data page channels are populated (A 1 in a particular bit indicates the data page channel for that field will contain data). The UUV-3500 uses this field to indicate which channels have been enabled for data acquisition.

System 3000/3900	System 5000 Mk1	System 5000 V2	UUV-3500
Bit 0 – Low Frequency Port channel populated Bit 1 – Low Frequency Stbd channel populated Bit 2 – High Frequency Port channel populated Bit 3 – High Frequency Stbd channel populated Bit 4 – Sub Bottom Profiler channel populated Bits 31-5: Reserved	Bit 0: Beam 1 populated Bit 1: Beam 2 populated Bit 2: Beam 3 populated Bit 3: Beam 4 populated Bit 4: Beam 5 populated Bit 5: Beam 6 populated Bit 6: Beam 7 populated Bit 7: Beam 8 populated Bit 8: Beam 9 populated Bit 9: Beam 10 populated Bits 21-10: Reserved Bit 22: Echo 1 populated Bit 23: Echo 2 populated Bit 24: Reserved Bit 25: Reserved Bit 26: Roll populated Bit 27: Yaw populated Bit 28: Aux Record 1 pop. Bit 29: Aux Record 2 pop. Bit 30: Reserved Bit 31: Reserved	N/A	Bit 0 – Channel 0 (LF Port side scan/Bathy) Bit 1 – Channel 1 (LF Stbd side scan/Bathy) Bit 2 – Channel 2 (HF Port side scan) Bit 3 – Channel 3 (HF Stbd side scan) Bit 4 – Channel 4 (LF Bathy) Bit 5 – Channel 5 (LF Bathy) Bit 6 – Channel 6 (LF Bathy) Bit 7 – Channel 7 (LF Bathy) Bits 23-8: Reserved. Bits 31-24: Framing Mode, coded as follows: 0 – Reserved 1 – LF Side Scan Only 2 – HF Side Scan Only 3 – LF and HF Side Scan 4 – Reserved (LF Bathy only) 5 – LF Side Scan and LF Bathy only 6 – LF Bathy and HF Side Scan 7 – LF and HF Side Scan and LF Bathy 8 – 255 – Reserved.

3. beamsToDisplay is a 5-bit wide field with bit 0 indicating that channels 1,6 are to be displayed, bit 1 indicating that channels 2,7 are to be displayed,...,bit 4 indicating that channels 5 and 10 are to be displayed. For a System 5000, channels 1-5 are port and 6-10 starboard, where 1 and 6 are the latest in time (beams 1/6 are fore and 5/10 are aft on fish).

Number Of Beams To Display	Active Beams (by number)		beamsToDisplay value (hex)
	Port	**Stbd**	
1	3	8	0x00000084
2	3,4	8,9	0x0000018C
3	2,3,4	7,8,9	0x000001CE
4	2,3,4,5	7,8,9,10	0x000003DE
5	1,2,3,4,5	6,7,8,9,10	0x000003FF

4. Despeckling is not supported by all System 5000 TPUs.
5. The depth value in the ping header is the raw voltage reading from the pressure sensor in volts. The towfish pressure sensors have a maximum pressure rating and a given voltage range. A number of different optional pressure sensors that have various maximum pressures and voltage ranges are supported. Depth is computed as follows:

   depth = ((((volts - voltageMin) / (voltageMax - voltageMin)) * maxPressure) - 14.696) / 1.487

   where,

   voltageMax is the maximum voltage reading of the pressure sensor
   voltageMin is the minimum voltage reading of the pressure sensor
   maxPressure is the maximum rated pressure for a given sensor in psia
   14.696 is standard atmospheric pressure
   1.487 is psi/meter

   As of vxWorks 8.0, the header fields pressureSensorMax, pressureSensorVoltageMin, and pressureSensorVoltageMax, and bit 11 of the capabilityMask may be set. If pressureSensorMax is non-zero, then the aforementioned fields can be used as inputs to the above equation.
6. The fishLat/fishLon are the towfish latitude and longitude, respectively.
   These fields are populated by the TPU if the TPU receives a “$--TLL” or “$PKLA,LL” on its serial Command/Navigation input. Otherwise, these fields are set to zero.
   The SonarPro® behavior with regards to fish position has changed over time.
   <u>SonarPro® V11.0 and later</u>
   These versions of SonarPro® have a towfish position preference. The preference selections are as follows:
   - Automatic (Use USBL if present, else use Layback if valid, else use Ship position): This selection setting mimics the behavior described in the following paragraph titled “SonarPro® V10.0 and earlier”
   - Use USBL ($TLL) Position as input to TPU: This selection uses the fish position in the header all the time (even if zero).
   - Use Layback to calculate towfish position: This selection computes fish position based upon layback. This setting will cause SonarPro® to override any non-zero fish position values from the TPU.
   - Use Ship Position: This selection sets the fish position to the ship position. This setting will cause SonarPro® to override any non-zero fish position values from the TPU.

   <u>SonarPro® V10.0 and earlier</u>

SonarPro® uses the fishLat and fishLat returned from the TPU as long as the fields are non-zero. If the fields are zero, SonarPro® does the following:

- If the cable out value is less than or equal to zero, the fish latitude and longitude are set to the ship latitude and longitude
- If the cable out value is greater than zero, the fish position is calculated based on the layback. The layback is calculated by using a right triangle and the Pythagorean Theorem. The layback is calculated as follows:
  - Depth = GPSheight + shieveZoff + fishDepth
  - Layback = square root(cableOut^2 + Depth^2)
- Note: SonarPro® uses the GPSheight, shieveZoff, and cableOut fields from the header.

SonarPro® does not modify the fishLat and fishLat values when saving to SDF files. The value from the TPU is preserved. When SonarPro® is saving the data to XTF files, the XTF fields representing towfish position are filled in with the towfish position based upon the above rules, i.e., the value from the TPU if non-zero, otherwise, the value calculated by SonarPro® (based upon the rules above).

7. Aux values are set by sending a $PAUV command to the TPU on its Command/Navigation input.
8. The layback fields may be set by the TPU master. These fields are provided to storage of the values used to compute towfish position via layback. The values are set by SonarPro® when SonarPro® is the TPU master. These values are used by SonarPro® when connected to the TPU as a slave.
9. The layback fish method and position are not used by the TPU. The TPU will always populate these fields with zero. SonarPro® will populate these fields when it calculates fish position based upon layback and the values will be set when SonarPro® stores the data to SDF files. For hull-mount systems, SonarPro® will populate these fields with the computed the position of the hull mount transducers and the values will be set when SonarPro® stores the data to SDF files.
10. Only valid for towfish equipped with an actuated (motor controlled) wing.
11. The fishHeadingOffset field may be set by the TPU master. SonarPro® adds this offset to the heading field to produce an adjusted towfish heading. This value may be used to calibrate the heading sensor to remove any offsets inherent in a given sensor.
12. The pressureSensorOffset field may be set by the TPU master. SonarPro® adds this offset to the depth value computed from the sensor depth field to produce an adjusted towfish depth. This value may be used to calibrate the pressure sensor to remove any offsets inherent in a given sensor.
13. The original version of the header (pageVersion < 3) had 44 words. The version 3 header contains 64 words. The version 4 header contains 128 words. The added words are noted in the comments.
14. For hull mount systems equipped with an Applanix POS MV™, speedFish is calculated from POS MV Group 1 message vesselSpeed. Otherwise, calculated from speed provided via NMEA RMC or VTG sentences. If the speedFilterSwitch is set, the speed value is filtered with a 90/10 low pass filter.
15. If the TPU 1 PPS input is to set the TPU time (not all TPUs support a 1PPS input), the TPU time of ping is derived from the 1PPS time.
16. The 5900 receiver gain settings use the tvgPage field in the header. The number for the receiver side scan gain and bathymetry gain represents the following gain values:

Gain Number	Gain Level
0	-22 dB
1	-16 dB
2	-10 dB
3	-4 dB
4	+2 dB

5	+8 dB
6	+14 dB
7	+20 dB

17. The txWaveform value is interpreted differently depending on the towfish type.
    a. For System 5000 and System 5000 V2, the txWaveform is a simple integer representing the transmit waveform number.
    b. For System 3000 and System 3900, the txWaveform is a bit field. Bits 0 thru 7 represent the Low Frequency transmit waveform number while Bits 8 thru 15 represent the High Frequency transmit waveform number.
    c. For System 5900, the txWaveform is a bit field.
        Bits 3:0: Side scan transmit waveform number
        Bits 7:4: Side scan bathy transmit waveform number
        Bits 11:8; Unused
        Bit 12: Side scan transmitter enable (1 = enabled, 0 = disabled)
        Bit 13: Side scan bathy transmitter enable (1 = enabled, 0 = disabled)
    d. For System 3500, the txWaveform is a bit field:
        Bits 3:0: Low Frequency transmit waveform number
        Bits 6:4: Unused
        Bit 7: Low Frequency transmitter enable (1 = enabled, 0 = disabled)
        Bits 11:8: High Frquency transmit waveform number
        Bits 14:12: Unused
        Bit 15: High Frequency transmitter enable (1 = enabled, 0 – disabled)

18. For hull-mount systems, the heading field is overwritten with the ship heading.

3.2. Data Page Channels

The header is followed by a unique data structure for each system. The data structure is also flexible in that the first value of each channel indicates the number of samples for that channel. A number of examples are as follows:

Table 3 – Data Page channel types

Channel Type	Description
unsigned short channel[]	First 2 bytes (as type unsigned short) indicate the number of unsigned short data samples that follow.
long channel[]	First 4 bytes (as type long) indicate the number of long (32-bit) data samples that follow.
float channel[]	First 4 bytes (as type float) indicate the number of float (32-bit) data samples that follow.

The arrays are therefore variable size and form a type of link-list (via the size information at the beginning of each channel).

3.2.1. System 3000

The System 3000 Version 4 data channel structure is as follows:

Table 4 - System 3000 Version 4 data channel structure

Vector Number	Type	Name	Description
1	unsigned short	portlf[]	Port low-freq
2	unsigned short	stbdlf[]	Stbd low-freq
3	unsigned short	porthf[]	Port high-freq
4	unsigned short	stbdhf[]	Stbd high-freq
5	long	sbp[]	Sub-bottom profiler

Each of the side scan data channel samples is a 16-bit unsigned quantity. Each of the Sub-bottom channel samples is a 32-bit signed quantity.

The System 3000 Version 3 data channel structure is as follows:

Table 5 - System 3000 Version 3 data channel structure

Vector Number	Type	Name	Description
1	unsigned short	portlf[]	Port low-freq
2	unsigned short	stbdlf[]	Stbd low-freq
3	unsigned short	porthf[]	Port high-freq
4	unsigned short	stbdhf[]	Stbd high-freq
5	short	sbp[]	Sub-bottom profiler

Each of the data channel samples is a 16-bit unsigned quantity.

Note: The “configuration” word in the data page header may be examined to determine the type of data in the data page. The “configuration” word is a bit mask that may contain a logical OR of the values in Table 6.

Table 6 – Data page header “configuration” word bit mask values

Bit Value	Meaning
0x03	Data page contains Low Frequency Side Scan Data when set
0x0C	Data page contains High Frequency Side Scan Data when set
0x10	Data page contains SBP data when set

3.2.2. System UUV-3500 Channel data page (pageVersion 3500) Obsolete

The System UUV 3500 data channel structure is as follows:

Table 7 - System UUV 3500 data channel structure

Vector Number	Type	Name	Description
1	`unsigned long`	`magData1[]`	
2	`unsigned long`	`magData2[]`	
3	`unsigned long`	`magData3[]`	
4	`unsigned long`	`magData4[]`	
5	`unsigned long`	`magData5[]`	
6	`unsigned long`	`magData6[]`	
7	`unsigned long`	`magData7[]`	
8	`unsigned long`	`magData8[]`	
9	`long`	`chanData1I[]`	
10	`long`	`chanData1Q[]`	
11	`long`	`chanData2I[]`	
12	`long`	`chanData2Q[]`	
13	`long`	`chanData3I[]`	
14	`long`	`chanData3Q[]`	
15	`long`	`chanData4I[]`	
16	`long`	`chanData4Q[]`	
17	`long`	`chanData5I[]`	
18	`long`	`chanData5Q[]`	
19	`long`	`chanData6I[]`	
20	`long`	`chanData6Q[]`	
21	`long`	`chanData7I[]`	
22	`long`	`chanData7Q[]`	
23	`long`	`chanData8I[]`	
24	`long`	`chanData8Q[]`	

Note: The "rawDataConfig" word in the data page header may be examined to determine the type of data produced by each channel of the SONAR. The "rawDataConfig" values are defined in Table 8.

Table 8 – UUV-3500 Data page header "rawDataConfig" definition

Value	Meaning	Data Page vectors Populated (Vector Number)
0	Channels 1, 2, 3, and 4 operate in magnitude mode.	1 thru 4.
1	Reserved.	
2	Channels 1, 2, 5, 6, 7, and 8 operate in narrow band complex mode. Channels 3 and 4 operate in magnitude mode.	3, 4, 9 thru 12, 17 thru 24.

3.2.3. System UUV-3500 Low Frequency Channel data page (pageVersion 3501)

The System UUV 3500 Low Frequency data channel structure is as follows:

Table 9 - System UUV 3500 Low Frequency data channel structure

Vector Number	Type	Name	Description
1	unsigned long	portLf[]	Low Frequency Port side scan
2	unsigned long	stbdLf[]	Low Frequency Stbd side scan

3.2.4. System UUV-3500 High Frequency Channel data page (pageVersion 3502)

The System UUV 3500 High Frequency data channel structure is as follows:

Table 10 - System UUV 3500 High Frequency data channel structure

Vector Number	Type	Name	Description
1	unsigned long	portHf[]	High Frequency Port side scan
2	unsigned long	stbdHf[]	High Frequency Stbd side scan

3.2.5. System UUV-3500 – Bathy Pulse Compressed Data Page (4-channel, pageVersion 3503)

The System UUV-3500 Bathy Pulse Compressed data page contains the pulse compressed version of the bathymetry channels (staves). This data page may be used directly for bathymetric processing without the need for performing pulse compression (match filtering). The pageVersion for this data page type is 3503.

The System UUV-3500 Bathy Pulse Compressed Beamformed data channel structure is as follows:

Table 11 - System UUV-3500 Bathy Pulse Compressed Data Page (4-channel) data channel structure

Vector Number	Type	Name	Description
1	long	chanData1i[]	
2	long	chanData1q[]	
3	long	chanData2i[]	
4	long	chanData2q[]	
5	long	chanData3i[]	
6	long	chanData3q[]	
7	long	chanData4i[]	
8	long	chanData4q[]	
9	long	chanData5i[]	
10	long	chanData5q[]	
11	long	chanData6i[]	
12	long	chanData6q[]	
13	long	chanData7i[]	
14	long	chanData7q[]	
15	long	chanData8i[]	
16	long	chanData8q[]	

3.2.6. UUV-3500 Bathy Processed Data Page (pageVersion 3511)

The UUV-3500 Bathy Processed page contains the processed bathymetry data based on the raw data bathymetry channels in the side scan arrays.
The pageVersion for this data page type is 3511.

Bathymetry Notes:

- A processed bathy value equal to -32768 (Hex 0x8000) is defined as Not A Number (NAN) and should be ignored.
- Range to sea bed detection is computed as (vector index/sampleFreq) * 750 m/s. The sampleFreq is from the SDF header.

- The angle of arrival is relative to the transducer face.
- Across track offsets (Y) are positive to starboard and negative to port.
- Depth values (Z) are relative to the sonar transducers. The values are positive and increase as the distance from the towfish to the seafloor increases.
- The rollVector array contains a roll correction value for each sample in the bathyPortAngle and bathyStbdAngle arrays. The roll vector is generated based on the motion correction source selected by the user. The motion correction source is recorded in the Bathy Processed Settings Record 1. To obtain the roll corrected angle of arrival, the rollVector should be subtracted from the bathyPortAngle array and added to the bathyStbdAngle array.
- The pitchVector and heaveVector arrays contain a pitch and heave value for each bathymetry sample if the user selected a motion sensor capable of pitch and heave measurement. These vectors may be used to apply lever arm corrections. The Ship Configuration Info 1 record contains the relative offsets between the sonar and the sensors.
- Along track offsets (X), Across track offsets (Y) and Depth values (Z) are computed with the roll vector applied to the bathyPortAngle and bathyStbdAngle arrays. The offsets and depth values may be computed with lever arm and/or sound velocity correction (user selectable). The SDF header value postProcessVersion indicates which processing was applied.

The UUV-3500 Bathy Processed data channel structure is as follows:

Table 12 – UUV-3500 Bathy Processed data channel structure

Vector Number	Type	Name	Description
1	short	bathyPortIntensity[]	Port "Beamformed" bathy backscatter intensity
2	short	bathyPortAngle[]	Best angle of arrival port * bathyScaleAngle (based upon SDF header speedSound sound speed)
3	short	bathyPortQuality[]	Quality associated with angle arrival (Quality * bathyScaleQuality)
4	short	bathyPortX[]	Along track port bathymetry * bathyScaleXYZ
5	short	bathyPortY[]	Horizontal (across track) port bathymetry, relative to towfish * bathyScaleXYZ
6	short	bathyPortZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
7	short	bathyStbdIntensity[]	Stbd "Beamformed" bathy backscatter intensity
8	short	bathyStbdAngle[]	Best angle of arrival stbd * bathyScaleAngle (based upon SDF header speedSound

			sound speed)
9	short	bathyStbdQuality[]	Quality associated with angle arrival (Quality * bathyScaleQuality)
10	short	bathyStbdX[]	Along track stbd bathymetry * bathyScaleXYZ
11	short	bathyStbdY[]	Horizontal (across track) stbd bathymetry, relative to towfish * bathyScaleXYZ
12	short	bathyStbdZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
13	short	rollVector[]	Bathy Roll correction applied to angle for X,Y,Z calculations * bathyRollScale.
14	short	pitchVector[]	Bathy Pitch correction applied to angle for X,Y,Z calculations * bathyPitchScale.
15	short	heaveVector[]	Bathy Heave correction applied to angle for X,Y,Z calculations * bathyHeaveScale. (heave at center point between port/stbd transducers)
16	short	bathyPortSNR[]	Port signal-to-noise ratio, * bathyScaleSNR
17	uns'nd short	bathyPortUncertainty[]	Port uncertainty * bathyScaleUncertainty
18	short	bathyStbdSNR[]	Starboard signal-to-noise ratio, * bathyScaleSNR
19	uns'nd short	bathyStbdUncertainty[]	Starboard uncertainty * bathyScaleUncertainty
20	short	reserved1[]	Future, reserved
21	short	reserved2[]	Future, reserved

3.2.7. System 5000 (Legacy vxWorks 6.31 and earlier or Side Scan Only Systems - pageVersion 5000/5001)

This data channel structure is the legacy data structure for System 5000 and System 5000 V2 with TPU vxWorks version 6.31 and earlier.

This data channel structure is the data structure for System 5000 and System 5000 V2 with TPU vxWorks version 7.00 and later that are side-scan only systems, i.e., no dedicated bathymetry channels.

The header pageVersion is 5000 or 5001 for this channel structure. The pageVersion value is dependent upon the size of the header. If the size of the header is 256 bytes, the pageVersion = 5000. If the size of the header is 512 bytes, the pageVersion = 5001.

The System 5000 data channel structure is as follows:

Table 13 - System 5000 and System 5000 V2 (Side-Scan only) data channel structure

Vector Number	Type	Name	Description
1	unsigned short	chan1Data[]	processed side-scan data
2	unsigned short	chan2Data[]	
3	unsigned short	chan3Data[]	
4	unsigned short	chan4Data[]	
5	unsigned short	chan5Data[]	
6	unsigned short	chan6Data[]	
7	unsigned short	chan7Data[]	
8	unsigned short	chan8Data[]	
9	unsigned short	chan9Data[]	
10	unsigned short	chan10Data[]	
11	short	bathyPort1i[]	Bathy 1 synthesized from element data
12	short	bathyPort1q[]	
13	short	bathyPort2i[]	Bathy 2,3 raw from arrays
14	short	bathyPort2q[]	
15	short	bathyPort3i[]	
16	short	bathyPort3q[]	
17	short	bathyStbd1i[]	
18	short	bathyStbd1q[]	
19	short	bathyStbd2i[]	
20	short	bathyStbd2q[]	
21	short	bathyStbd3i[]	
22	short	bathyStbd3q[]	
23	short	echo1[]	Raw altimeter 1 data
24	short	echo2[]	Raw altimeter 2 data
25	short	subBottom1[]	
26	short	subBottom2[]	
27	short	rollSensor[]	
28	short	yawRate[]	
29	short	rawdataPort1i[]	1-12 = port raw side-scan
30	short	rawdataPort1q[]	
31	short	rawdataPort2i[]	
32	short	rawdataPort2q[]	
33	short	rawdataPort3i[]	
34	short	rawdataPort2q[]	
35	short	rawdataPort4i[]	
36	short	rawdataPort4q[]	
37	short	rawdataPort5i[]	
38	short	rawdataPort5q[]	
39	short	rawdataPort6i[]	
40	short	rawdataPort6q[]	
41	short	rawdataPort7i[]	
42	short	rawdataPort7q[]	
43	short	rawdataPort8i[]	
44	short	rawdataPort8q[]	
45	short	rawdataPort9i[]	

46	short	rawdataPort9q[]	
47	short	rawdataPort10i[]	
48	short	rawdataPort10q[]	
49	short	rawdataPort11i[]	
50	short	rawdataPort11q[]	
51	short	rawdataPort12i[]	
52	short	rawdataPort12q[]	
53	short	rawdataPort13i[]	bathy string #2
54	short	rawdataPort13q[]	
55	short	rawdataPort14i[]	bathy string #3
56	short	rawdataPort14q[]	
57	short	rawdataStbd1i[]	1-12 = stbd raw side-scan
58	short	rawdataStbd1q[]	
59	short	rawdataStbd2i[]	
60	short	rawdataStbd2q[]	
61	short	rawdataStbd3i[]	
62	short	rawdataStbd2q[]	
63	short	rawdataStbd4i[]	
64	short	rawdataStbd4q[]	
65	short	rawdataStbd5i[]	
66	short	rawdataStbd5q[]	
67	short	rawdataStbd6i[]	
68	short	rawdataStbd6q[]	
69	short	rawdataStbd7i[]	
70	short	rawdataStbd7q[]	
71	short	rawdataStbd8i[]	
72	short	rawdataStbd8q[]	
73	short	rawdataStbd9i[]	
74	short	rawdataStbd9q[]	
75	short	rawdataStbd10i[]	
76	short	rawdataStbd10q[]	
77	short	rawdataStbd11i[]	
78	short	rawdataStbd11q[]	
79	short	rawdataStbd12i[]	
80	short	rawdataStbd12q[]	
81	short	rawdataStbd13i[]	bathy string #2
82	short	rawdataStbd13q[]	
83	short	rawdataStbd14i[]	bathy string #3
84	short	rawdataStbd14q[]	

3.2.8. *System 5000 w/ Bathy support (vxWorks V7.00 and later, pageVersion 5002)*

This data channel structure is for System 5000 and System 5000 V2 systems that are equipped with 14-total raw data channels per side with 2 of the 14 channels dedicated for bathy. This data format will be output by the TPU if the System is equipped with a Klein Motion Sensor or if the TPU rawDataConfig is set to acquire the raw bathymetry channels. Alternatively, this data format will be output by the Klein Bathy engine after the data is processed by SonarPro or the KleinSDK. The header pageVersion is 5002 for this channel structure.

Bathymetry Notes:

- A processed bathy value equal to -32768 (Hex 0x8000) is defined as Not A Number (NAN) and should be ignored.
- Range to sea bed detection is computed as (vector index/samplesFreq) * 750 m/s. The sampleFreq is from the SDF header.
- The angle of arrival is relative to the transducer face.
- Across track offsets (Y) are positive to starboard and negative to port.
- Depth values (Z) are relative to the sonar transducers. The values are positive and increase as the distance from the towfish to the seafloor increases.
- The rollVector array contains a roll correction value for each sample in the bathyPortAngle and bathyStbdAngle arrays. The roll vector is generated based on the motion correction source selected by the user. The motion correction source is recorded in the Bathy Processed Settings Record 1. To obtain the roll corrected angle of arrival, the rollVector should be subtracted from the bathyPortAngle array and added to the bathyStbdAngle array.
- Across track offsets (Y) and Depth values (Z) are computed with the roll vector applied to the bathyPortAngle and bathyStbdAngle arrays.

The data channel structure is as follows:

Table 14 - System 5000 14-Channels per Side (vxWorks 6.31 and earlier) data channel structure

Vector Number	Type	Name	Description
1	unsigned short	chan1Data[]	processed side-scan data
2	unsigned short	chan2Data[]	
3	unsigned short	chan3Data[]	
4	unsigned short	chan4Data[]	
5	unsigned short	chan5Data[]	
6	unsigned short	chan6Data[]	
7	unsigned short	chan7Data[]	
8	unsigned short	chan8Data[]	
9	unsigned short	chan9Data[]	
10	unsigned short	chan10Data[]	
11	short	bathyPortAngle[]	Best direction of arrival relative to transducers (radians * 10000/2*PI)
12	short	bathyPortQuality[]	Quality associated with angle arrival (Quality * 10000)
13	short	bathyPortY[]	Horizontal bathymetry, relative to towfish (centimeters)
14	short	bathyPortZ[]	Vertical bathymetry, relative to towfish (centimeters)
15	short	bathyStbdAngle[]	Best direction of

			arrival relative to transducers (radians * 10000/2*PI)
16	short	bathyStbdQuality[]	Quality associated with angle arrival (Quality * 10000)
17	short	bathyStbdY[]	Horizontal bathymetry, relative to towfish (centimeters)
18	short	bathyStbdZ[]	Vertical bathymetry, relative to towfish (centimeters)
19	short	rollVector[]	Roll correction value (in milliRadians) applied to angle for Y, Z calculations.
20	short	reserved1[]	Future, reserved
21	short	reserved2[]	Future, reserved
22	short	reserved3[]	Future, reserved
23	short	echo1[]	
24	short	echo2[]	
25	short	auxRecords1[]	Auxiliary records channel 1
26	short	auxRecords2[]	Auxiliary records channel 2
27	short	rollSensor[]	
28	short	yawRate[]	
29	short	rawdataPort1i[]	1-12 = port raw side-scan
30	short	rawdataPort1q[]	
31	short	rawdataPort2i[]	
32	short	rawdataPort2q[]	
33	short	rawdataPort3i[]	
34	short	rawdataPort2q[]	
35	short	rawdataPort4i[]	
36	short	rawdataPort4q[]	
37	short	rawdataPort5i[]	
38	short	rawdataPort5q[]	
39	short	rawdataPort6i[]	
40	short	rawdataPort6q[]	
41	short	rawdataPort7i[]	
42	short	rawdataPort7q[]	
43	short	rawdataPort8i[]	
44	short	rawdataPort8q[]	
45	short	rawdataPort9i[]	
46	short	rawdataPort9q[]	
47	short	rawdataPort10i[]	
48	short	rawdataPort10q[]	
49	short	rawdataPort11i[]	
50	short	rawdataPort11q[]	
51	short	rawdataPort12i[]	
52	short	rawdataPort12q[]	
53	short	rawdataPort13i[]	bathy string #2
54	short	rawdataPort13q[]	
55	short	rawdataPort14i[]	bathy string #3

56	short	rawdataPort14q[]	
57	short	rawdataStbd1i[]	1-12 = stbd raw side-scan
58	short	rawdataStbd1q[]	
59	short	rawdataStbd2i[]	
60	short	rawdataStbd2q[]	
61	short	rawdataStbd3i[]	
62	short	rawdataStbd2q[]	
63	short	rawdataStbd4i[]	
64	short	rawdataStbd4q[]	
65	short	rawdataStbd5i[]	
66	short	rawdataStbd5q[]	
67	short	rawdataStbd6i[]	
68	short	rawdataStbd6q[]	
69	short	rawdataStbd7i[]	
70	short	rawdataStbd7q[]	
71	short	rawdataStbd8i[]	
72	short	rawdataStbd8q[]	
73	short	rawdataStbd9i[]	
74	short	rawdataStbd9q[]	
75	short	rawdataStbd10i[]	
76	short	rawdataStbd10q[]	
77	short	rawdataStbd11i[]	
78	short	rawdataStbd11q[]	
79	short	rawdataStbd12i[]	
80	short	rawdataStbd12q[]	
81	short	rawdataStbd13i[]	bathy string #2
82	short	rawdataStbd13q[]	
83	short	rawdataStbd14i[]	bathy string #3
84	short	rawdataStbd14q[]	

3.2.9. System 5000 V2 with15-Channels per side, including 3 dedicated Bathy channels (vxWorks V8.06 and later, pageVersion 5004)

This data channel structure is System 5000 V2 systems that are equipped with 15-total raw data channels (per side) with 3 of the 15 channels dedicated for bathy. The header pageVersion is 5004 for this channel structure. This data channel layout expands upon the bathy vectors to accommodate an along track component and an intensity component.

Bathymetry Notes:

- A processed bathy value equal to -32768 (Hex 0x8000) is defined as Not A Number (NAN) and should be ignored.
- Range to sea bed detection is computed as (vector index/samplesFreq) * 750 m/s. The sampleFreq is from the SDF header.
- The angle of arrival is relative to the transducer face.
- Across track offsets (Y) are positive to starboard and negative to port.
- Depth values (Z) are relative to the sonar transducers. The values are positive and increase as the distance from the towfish to the seafloor increases.
- The rollVector array contains a roll correction value for each sample in the bathyPortAngle and bathyStbdAngle arrays. The roll vector is generated based on the motion correction source selected by the user. The motion correction source is recorded in the Bathy Processed Settings Record 1. To obtain the roll corrected angle of arrival, the rollVector should be subtracted from the bathyPortAngle array and added to the bathyStbdAngle array.

- The pitchVector and heaveVector arrays contain a pitch and heave value for each bathymetry sample if the user selected a motion sensor capable of pitch and heave measurement. These vectors may be used to apply lever arm corrections. The Ship Configuration Info 1 record contains the relative offsets between the sonar and the sensors.
- Along track offsets (X), Across track offsets (Y) and Depth values (Z) are computed with the roll vector applied to the bathyPortAngle and bathyStbdAngle arrays. The offsets and depth values may be computed with lever arm and/or sound velocity correction (user selectable). The SDF header value postProcessVersion indicates which processing was applied.

The System 5000 V2, which has been equipped with 3 bathy channels (per side), data structure is as follows:

Table 15 - System 5000 V2 15 Channels per Side (vxWorks 8.06 and later) data channel structure

Vector Number	Type	Name	Description
1	unsigned short	chan1Data[]	processed side-scan data
2	unsigned short	chan2Data[]	
3	unsigned short	chan3Data[]	
4	unsigned short	chan4Data[]	
5	unsigned short	chan5Data[]	
6	unsigned short	chan6Data[]	
7	unsigned short	chan7Data[]	
8	unsigned short	chan8Data[]	
9	unsigned short	chan9Data[]	
10	unsigned short	chan10Data[]	
11	short	bathyPortIntensity[]	Port "Beamformed" bathy backscatter intensity
12	short	bathyPortAngle[]	Best angle of arrival port * bathyScaleAngle (based upon SDF header speedSound sound speed)
13	short	bathyPortQuality[]	Quality associated with angle arrival (Quality * bathyScaleQuality)
14	short	bathyPortX[]	Along track port bathymetry * bathyScaleXYZ
15	short	bathyPortY[]	Horizontal (across track) port bathymetry, relative to towfish * bathyScaleXYZ
16	short	bathyPortZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
17	short	bathyStbdIntensity[]	Stbd "Beamformed" bathy backscatter intensity

18	short	bathyStbdAngle[]	Best angle of arrival stbd * bathyScaleAngle (based upon SDF header speedSound sound speed)
19	short	bathyStbdQuality[]	Quality associated with angle arrival (Quality * bathyScaleQuality)
20	short	bathyStbdX[]	Along track stbd bathymetry * bathyScaleXYZ
21	short	bathyStbdY[]	Horizontal (across track) stbd bathymetry, relative to towfish * bathyScaleXYZ
22	short	bathyStbdZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
23	short	rollVector[]	Bathy Roll correction applied to angle for X, Y, Z calculations * bathyRollScale.
24	short	pitchVector[]	Bathy Pitch correction applied to angle for X, Y, Z calculations * bathyPitchScale.
25	short	heaveVector[]	Bathy Heave correction applied to angle for X, Y, Z calculations * bathyHeaveScale. (heave at center point between port/stbd transducers)
26	short	bathyPortSNR[]	Port signal-to-noise ratio * bathyScaleSNR
27	uns'nd short	bathyPortUncertainty[]	Port uncertainty * bathyScaleUncertainty
28	short	bathyStbdSNR[]	Starboard signal-to-noise ratio * bathyScaleSNR
29	uns'nd short	bathyStbdUncertainty[]	Starboard uncertainty * bathyScaleUncertainty
30	short	reserved1[]	Future, reserved
31	short	reserved2[]	Future, reserved
32	short	rawdataPort1i[]	1-15 = port raw channels
33	short	rawdataPort1q[]	
34	short	rawdataPort2i[]	
35	short	rawdataPort2q[]	
36	short	rawdataPort3i[]	
37	short	rawdataPort2q[]	
38	short	rawdataPort4i[]	
39	short	rawdataPort4q[]	
40	short	rawdataPort5i[]	
41	short	rawdataPort5q[]	
42	short	rawdataPort6i[]	
43	short	rawdataPort6q[]	
44	short	rawdataPort7i[]	
45	short	rawdataPort7q[]	
46	short	rawdataPort8i[]	

47	short	rawdataPort8q[]	
48	short	rawdataPort9i[]	
49	short	rawdataPort9q[]	
50	short	rawdataPort10i[]	
51	short	rawdataPort10q[]	
52	short	rawdataPort11i[]	
53	short	rawdataPort11q[]	
54	short	rawdataPort12i[]	
55	short	rawdataPort12q[]	
56	short	rawdataPort13i[]	
57	short	rawdataPort13q[]	
58	short	rawdataPort14i[]	
59	short	rawdataPort14q[]	
60	short	rawdataPort15i[]	
61	short	rawdataPort15q[]	
62	short	rawdataPortReserved1[]	future, reserved
63	short	rawdataPortReserved2[]	future, reserved
64	short	rawdataPortReserved3[]	future, reserved
65	short	rawdataPortReserved4[]	future, reserved
66	short	rawdataPortReserved5[]	future, reserved
67	short	rawdataPortReserved6[]	future, reserved
68	short	rawdataStbd1i[]	1-15 = stbd raw channels
69	short	rawdataStbd1q[]	
70	short	rawdataStbd2i[]	
71	short	rawdataStbd2q[]	
72	short	rawdataStbd3i[]	
73	short	rawdataStbd2q[]	
74	short	rawdataStbd4i[]	
75	short	rawdataStbd4q[]	
76	short	rawdataStbd5i[]	
77	short	rawdataStbd5q[]	
78	short	rawdataStbd6i[]	
79	short	rawdataStbd6q[]	
80	short	rawdataStbd7i[]	
81	short	rawdataStbd7q[]	
82	short	rawdataStbd8i[]	
83	short	rawdataStbd8q[]	
84	short	rawdataStbd9i[]	
85	short	rawdataStbd9q[]	
86	short	rawdataStbd10i[]	
87	short	rawdataStbd10q[]	
88	short	rawdataStbd11i[]	
89	short	rawdataStbd11q[]	
90	short	rawdataStbd12i[]	
91	short	rawdataStbd12q[]	
92	short	rawdataStbd13i[]	
93	short	rawdataStbd13q[]	
94	short	rawdataStbd14i[]	
95	short	rawdataStbd14q[]	
96	short	rawdataStbd15i[]	
97	short	rawdataStbd15q[]	
98	short	rawdataStbdReserved1[]	future, reserved
99	short	rawdataStbdReserved2[]	future, reserved
100	short	rawdataStbdReserved3[]	future, reserved

101	short	rawdataStbdReserved4[]	future, reserved
102	short	rawdataStbdReserved5[]	future, reserved
103	short	rawdataStbdReserved6[]	future, reserved
104	long	subbottom1[]	future, reserved
105	long	subbottom2[]	future, reserved
106	long	subbottom3[]	future, reserved
107	long	subbottom4[]	future, reserved

3.2.10. System 7180 (NOAA LRSSS) Bathy Processed Data Page (pageVersion 7191)

The System 7180 Bathy Processed page contains the processed bathymetry data based on the NOAA LRSSS SDF2 raw data input.
The pageVersion for this data page type is 7191.

Bathymetry Notes:

- A processed bathy value equal to -32768 (Hex 0x8000) is defined as Not A Number (NAN) and should be ignored.
- Range to sea bed detection is computed as (vector index/sampleFreq) * 750 m/s. The sampleFreq is from the SDF header.
- The angle of arrival is relative to the transducer face.
- Across track offsets (Y) are positive to starboard and negative to port.
- Depth values (Z) are relative to the sonar transducers. The values are positive and increase as the distance from the towfish to the seafloor increases.
- The rollVector array contains a roll correction value for each sample in the bathyPortAngle and bathyStbdAngle arrays. The roll vector is generated based on the Octans Motion Reference Unit. The motion correction source is recorded in the Bathy Processed Settings Record 1. To obtain the roll corrected angle of arrival, the rollVector should be subtracted from the bathyPortAngle array and added to the bathyStbdAngle array.
- The pitchVector array contains a pitch value for each bathymetry sample in the bathyPortAngle and bathyStbdAngle arrays. The pitch vector is generated based on the Octans Motion Reference Unit.
- Across track offsets (Y) and Depth values (Z) are computed with the roll vector applied to the bathyPortAngle and bathyStbdAngle arrays. The offsets and depth values may be computed with sound velocity correction (user selectable). The SDF header value postProcessVersion indicates which processing was applied.
- The following vectors are included to maintain compatibility with other Klein Sonar systems. They are defined but unused in this application: bathyPortX, bathyStbdX, pitchVector, heaveVector.

NOTES:

- Vectors which are not populated but are place holders reserved for future use consist of a sample count of 0 for that vector. In this specific case, the vectors are declared as 16-bits shorts. Therefore, for each unpopulated vector, there will be 2 bytes that are set to 0 to indicate that vector is empty.
- Intensity vectors are not populated. Reserved for future use.
- Quality2 vectors are not populated. Reserved for future use.
- Uncertainty vectors are not populated. Reserved for future use.
- Bathy X vectors are not populated. Reserved for future use.

The System 7180 Bathy Processed data channel structure is as follows:

Table 16 - 7180 Bathy Processed data channel structure

Vector	Type	Name	Description

Number			
1	short	bathyPortIntensity[]	Place holder for future, Reserved.
2	short	bathyPortAngle[]	Best angle of arrival port * bathyScaleAngle (based upon 1500 m/s sound speed)
3	short	bathyPortQuality1[]	Quality factor #1 associated with angle arrival (Quality * bathyScaleQuality1)
4	short	bathyPortQuality2[]	Place holder for future, Reserved.
5	short	bathyPortUncertainty[]	Place holder for future, Reserved.
6	short	bathyPortX[]	Place holder for future, Reserved.
7	short	bathyPortY[]	Horizontal (across track) port bathymetry, relative to towfish * bathyScaleXYZ
8	short	bathyPortZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
9	short	bathyStbdIntensity[]	Place holder for future, Reserved.
10	short	bathyStbdAngle[]	Best angle of arrival stbd * bathyScaleAngle (based upon 1500 m/s sound speed)
11	short	bathyStbdQuality1[]	Quality Factor #1 associated with angle arrival (Quality * bathyScaleQuality)
12	short	bathyStbdQuality2[]	Place holder for future, Reserved..
13	short	bathyStbdUncertainty[]	Place holder for future, Reserved.
14	short	bathyStbdX[]	Place holder for future, Reserved.
15	short	bathyStbdY[]	Horizontal (across track) stbd bathymetry, relative to towfish * bathyScaleXYZ
16	short	bathyStbdZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
17	short	rollVector[]	Bathy Roll

			correction to be applied to angle for X,Y,Z calculations * bathyRollScale.
18	short	pitchVector	Per sample Pitch values * bathyPitchScale.
19	short	heaveVector[]	Place holder for future, Reserved.
20	short	reserved1[]	Future, reserved
21	short	reserved2[]	Future, reserved
22	short	reserved3[]	Future, reserved
23	short	reserved4[]	Future, reserved
24	short	reserved5[]	Future, reserved
25	short	reserved6[]	Future, reserved

3.3. SDF Extension Layout

The SDFX extension (SDFX) is a variable sized storage area. The first 4 bytes are the size of extension area in bytes, including the size bytes. The SDFX may contain any number of data records. Each data record has a common header defining the record ID, size, and version. For each record ID, there may be a unique data format. The end of the SDFX area is indicated by a SDFX record header with a recordId of 0xEEEEEEEE. An "empty" SDF extension containing the size bytes and the end header record would have an sdfExtensionSize of 68 (4 bytes for the sdfExtensionSize plus 64 bytes for the SDFX end header record).

U32 sdfExtensionSize;
Record 1
Record 2
…
Record N
Record End

Each data record in the SDFX area begins with the following header:

Data type	Parameter	Description
U32	recordId	Unique record identifier, 1 or greater.
U32	recordNumBytes	Total number of bytes in record, including this header
U32	headerVersion	The version of this header
U32	recordVersion	The version of this record
U32	reserved[12]	Reserved

The SDFX data record is terminated with the SDFX End Data Record as follows:

Data type	Parameter	Description
U32	recordId	0xEEEEEEEE
U32	recordNumBytes	64
U32	headerVersion	1
U32	recordVersion	1
U32	reserved[12]	Reserved

The next table summarizes the SDF record types.

Record Type	Record Id	Description
End Data Record	0xEEEEEEEE	Data record indicating the end of the SDFX section.

Ship Configuration Info 1	1	System configuration information
Bathy Calibration Record 1	2	Bathymetric system calibration information
Bathy Engine Settings Record 1	3	Private configuration information for the Bathy processing engine
Bathy Processed Settings Record 1	4	Bathymetric output scale factors and user selected bathy processing selections
UUV-3500 Diag 1	0x10	UUV-3500 Diagnostics results number 1
POS MV Group 1 Data	0x101	Record derived from POS MV Group 1
POS MV Group 102 Data	0x166	Record derived from POS MV Group 102
POS MV Group 103 Data	0x167	Record derived from POS MV Group 103
POS MV Group 112 Data	0x170	Record derived from POS MV Group 112
Sound Velocity Profile 1	0x200	Sound Velocity Profile record
NMEA Data Record 1	0x201	Record derived from NMEA serial input
System 5900 Sensor Data Record 1	0x202	Record derived from System 5900 sensor interface board
Klein Motion Sensor Raw Data 1	0x300	Raw Klein Motion Sensor (KMS) data packets along with TPU receive times
Klein Motion Sensor Processed Data 1	0x301	Processed data derived from raw KMS data packets
Octopus F180 Data	0x400	Record derived from F180 MCOM data packet
Teledyne TSS DMS-XX	0x500	Record derived from Teledyne TSS DMS-XX TSS1 data packet
Ocean Server IMU	0x600	Record derived from Ocean Server Inertial Measurement Unit serial message
System 7180 SDF2 data page header	0x7000	Record converted from SDF2 format "FISHPACDATAPAGEHDR" structure
Reserved for LRSS	0x7001-0x701F	Reserved for future use for LRSSS.
System 7180 SDF2 Octans sensor	0x7100	Record converted from SDF2 Octans sensor format "OCTANSDATASECTION" structure
System 7180 SDF2 Actuator Data Section sensor	0x7101	Record converted from SDF2 Actuator sensor format "ACTUATOR_DATASECTION" (subType 0) structure
System 7180 SDF2 Actuator Initialization Section sensor	0x7102	Record converted from SDF2 Actuator sensor format "ACTUATOR_INITIALIZATION_DATASECTION" (subType 1) structure
System 7180 SDF2 DVL sensor	0x7103	Record converted from SDF2 DVL sensor format "DVLDATASECTION" structure
System 7180 SDF2 GPS sensor	0x7104	Record converted from SDF2 GPS sensor format "GPSDATASECTION" structure
System 7180 SDF2 SVP sensor	0x7105	Record converted from SDF2 SVP sensor format
System 7180 SDF2 Transmissometer sensor	0x7106	Record converted from SDF2 Transmissometer sensor format
System 7180 SDF2 Depth sensor	0x7107	Record converted from SDF2 Depth sensor format
System 7180 SDF2	0x7108	Record converted from SDF2 Cable Out sensor format

Cable Out sensor		"CABLEOUTDATASECTION" structure
Reserved for LRSS	0x7109- 0x711F	Reserved for future use for LRSSS.

3.3.1. Ship Configuration Info 1

The Ship Configuration Info 1 contains the system configuration information for a particular ship installation.

Data type	Parameter	units
SDFX record header	recordHeader	The SDFX Record header recordId = 1 recordVersion = 1
F32	ShipLength	ship dimension, for validity check and display (meters)
F32	ShipWidth	ship dimension, for validity check and display (meters)
F32	ShipHeight	ship dimension, for validity check and display (meters)
F32	DatumX	SonarPro display only. Datum position from front of ship (meters)
F32	DatumY	SonarPro display only. Datum position from port side (meters)
F32	DatumZ	SonarPro display only. Datum position from top of ship (meters)
F32	ArrayX	Distance of arrays forward of datum (meters)
F32	ArrayY	Distance of midpoint between arrays starboard of datum (meters)
F32	ArrayZ	Distance of arrays below datum (meters)
F32	ArraySpacing	Array separation laterally (meters)
F32	MotionX	Distance of motion sensor forward of datum (meters)
F32	MotionY	Distance of motion sensor starboard of datum (meters)
F32	MotionZ	Distance of motion sensor below datum (meters)
F32	PositionX	Distance of position sensor forward of datum (meters)
F32	PostionY	Distance of position sensor starboard of datum (meters)
F32	PositionZ	Distance of position sensor below datum (meters)
F32	Draft	Array depth below waterline (meters)
F32	RollBiasPort	Port array roll bias (positive if port side high) (degrees)
F32	PitchBiasPort	Port array pitch bias (positive if bow high) (degrees)
F32	HeadingBiasPort	Port array heading bias (positive if clockwise) (degrees)
F32	RollBiasStbd	Starboard array roll bias (positive if port side high) (degrees)
F32	PitchBiasStbd	Starboard array pitch bias (positive if bow high) (degrees)
F32	HeadingBiasStbd	Starboard array heading bias (positive if clockwise) (degrees)

3.3.2. Bathy Calibration Record 1

The Bathy Calibration Record 1 contains the system calibration information for a particular transducer (hull-mount or towed) installation.

Data type	Parameter	units
SDFX record header	recordHeader	The SDFX Record header recordId = 2 recordVersion = 1
U32	SystemId	System Identification value. TBD. COULD BE TOWFISH S/N
U32	TransducerPortId	Port Transducer Identification value. TBD. COULD BE DUCER S/N
U32	TransducerStbdId	Starboard Transducer Identification value. TBD. COULD BE DUCER S/N
U32	TransducerCountPort	The number of Port bathy transducers

U32	TransducerCountStbd	The number of Starboard bathy transducers
U32	TransducerOrientation	Enumerated value defining transducer layout, e.g., vertical. Reserved for future use.
F32	TransducerSpacingPort1	Port bathy transducer 1 distance from bathy channel 1 (wavelength)
F32	TransducerSpacingPort2	Port bathy transducer 2 distance from bathy channel 1 (wavelength)
F32	TransducerSpacingPort3	Port bathy transducer 3 distance from bathy channel 1 (wavelength)
F32	TransducerSpacingPort4	Port bathy transducer 4 distance from bathy channel 1 (wavelength)
F32	TransducerSpacingStbd1	Starboard bathy transducer 1 distance from bathy channel 1 (wavelength)
F32	TransducerSpacingStbd2	Starboard bathy transducer 2 distance from bathy channel 1 (wavelength)
F32	TransducerSpacingStbd3	Starboard bathy transducer 3 distance from bathy channel 1 (wavelength)
F32	TransducerSpacingStbd4	Starboard bathy transducer 4 distance from bathy channel 1 (wavelength)
F32	PhaseCorrectionPort1	Port channel 1 electronic/transducer phase correction (radians)
F32	PhaseCorrectionPort2	Port channel 2 electronic/transducer phase correction (radians)
F32	PhaseCorrectionPort3	Port channel 3 electronic/transducer phase correction (radians)
F32	PhaseCorrectionPort4	Port channel 4 electronic/transducer phase correction (radians)
F32	PhaseCorrectionStbd1	Starboard channel 1 electronic/transducer phase correction (radians)
F32	PhaseCorrectionStbd2	Starboard channel 2 electronic/transducer phase correction (radians)
F32	PhaseCorrectionStbd3	Starboard channel 3 electronic/transducer phase correction (radians)
F32	PhaseCorrectionStbd4	Starboard channel 4 electronic/transducer phase correction (radians)
F32	DepressionAnglePort	Port Array depression angle (from horizontal, mechanical degrees)
F32	DepressionAngleStbd	Starboard Array depression angle (from horizontal, mechanical degrees)

3.3.3. Bathy Engine Settings Record 1
Reserved.

3.3.4. Bathy Processed Settings Record 1
The Bathy Processed Setting record contains the bathy output vector scale factors and the user selected bathy processing selections, e.g., the type of motion correction or sound speed correction applied.

Data type	**Parameter**	**units**
SDFX record header	recordHeader	The SDFX Record header recordId = 4 recordVersion = 1
F32	bathyScaleAngle	If this record not present in the SDF record, divide bathy[Port,Stbd]Angle by (10000 / 2 * PI) to get radians; otherwise, use this value.
F32	bathyScaleQuality	If this record not present in the SDF record, divide

		bathy Quality vector values by 10000 to get result; otherwise, use this value.
F32	bathyScaleXyz	If this record not present in the SDF record, divide bathy X, Y and Z values by 100 to get meters; otherwise, use this value.
F32	bathyScaleRoll	If this record not present in the SDF record, divide bathy roll vector by 1000 to get radians; otherwise, use this value
F32	bathyScalePitch	Divide bathy pitch vector by this value to get radians.
F32	bathyScaleHeave	Divide bathy heave vector by this value to get meters.
F32	bathyScaleSoundSpeedCorrection	Multiply bathy sound speed correction vectors by this value to get meters.
U32	bathyMotionType	Type of motion correction applied to bathy output. 0 = None 1 = Towfish compass 2 = Auxiliary sensor 3 = Klein Motion Sensor 4 = POS MV
U32	bathySoundSpeedType	Type of sound speed correction applied. 0 = None 1 = Array Sound Speed 2 = Sound Velocity Profile 3 = Array Sound Speed and Sound Velocity Profile
U32	bathyProcessingOptimizations	Processing optimization utilized. 0 = Exhaustive Search 1 = Normal Search

Data type	**Parameter**	**units**
SDFX record header	recordHeader	The SDFX Record header recordId = 4 recordVersion = 2
F32	bathyScaleIntensity	Divide bathy intensity vectors by this value to get intensity (magnitude).
F32	bathyScaleAngle	Divide bathy Angle vectors by this value to get radians.
F32	bathyScaleQuality1	Divide bathy Quality1 vector values by this value to get value between 0 and 1.
F32	bathyScaleQuality2	Divide bathy Quality2 vector values by this value to get value in between TBD.
F32	bathyScaleUncertainty	Divide Uncertainty vectors by this value to get uncertainty.
F32	bathyScaleXyz	Divide bathy X, Y and Z values by this value to get meters.
F32	bathyScaleRoll	Divide bathy roll vector by this value to get radians.
F32	bathyScalePitch	Divide bathy pitch vector by this value to get radians.
F32	bathyScaleHeave	Divide bathy heave vector by this value to get meters.
F32	bathyScaleSoundSpeedCorrection	Multiply bathy sound speed correction vectors by this value to get meters.
U32	bathyMotionType	Type of motion correction applied to bathy output.

		0 = None 1 = Towfish compass 2 = Auxiliary sensor 3 = Klein Motion Sensor 4 = POS MV 5 = F180 6 = DMS 7 = Octans
U32	bathySoundSpeedType	Type of sound speed correction applied. 0 = None 1 = Array Sound Speed 2 = Sound Velocity Profile 3 = Array Sound Speed and Sound Velocity Profile
U32	bathyProcessingOptimizations	Processing optimization utilized. 0 = Exhaustive Search 1 = Normal Search

Data type	**Parameter**	**units**
SDFX record header	recordHeader	The SDFX Record header recordId = 4 recordVersion = 3
F32	bathyScaleIntensity	Divide bathy intensity vectors by this value to get intensity (magnitude).
F32	bathyScaleAngle	If this record not present in the SDF record, divide bathy[Port,Stbd]Angle by (10000 / 2 * PI) to get radians; otherwise, use this value.
F32	bathyScaleQuality	If this record not present in the SDF record, divide bathy Quality vector values by 10000 to get result; otherwise, use this value.
F32	bathyScaleQuality2	If this record not present in the SDF record, divide bathy Quality2 vector values by this value to get value, otherwise, use this value (future use).
F32	bathyScaleUncertainty	Divide bathy Uncertainty vector values by this value to get uncertainty in radians.
F32	bathyScaleSNR	Divide bathy SNR vector values by this value to get signal-to-noise ratio value in dB
F32	bathyScaleXyz	If this record not present in the SDF record, divide bathy X, Y and Z values by 100 to get meters; otherwise, use this value.
F32	bathyScaleRoll	If this record not present in the SDF record, divide bathy roll vector by 1000 to get radians; otherwise, use this value
F32	bathyScalePitch	Divide bathy pitch vector by this value to get radians.
F32	bathyScaleHeave	Divide bathy heave vector by this value to get meters.
F32	bathyScaleSoundSpeedCorrection	Multiply bathy sound speed correction vectors by this value to get meters.
F32	reserved1	(future use)
F32	reserved2	(future use)
U32	bathyMotionType	Type of motion correction applied to bathy output. 0 = None 1 = Towfish compass

		2 = Auxiliary sensor 3 = Klein Motion Sensor 4 = POS MV
U32	bathySoundSpeedType	Type of sound speed correction applied. 0 = None 1 = Array Sound Speed 2 = Sound Velocity Profile 3 = Array Sound Speed and Sound Velocity Profile
U32	bathyProcessingOptimizations	Processing optimization utilized. 0 = Exhaustive Search 1 = Normal Search

3.3.5. UUV-3500 Diagnostics 1

The signal level diagnostic results are a set of signal level pairs.
The following special type (KLEIN_RX_SIG_LEVEL_1) defines the signal level pair.

Data type	**Parameter**	**Description**
F32	rxLevelTxRegion	Average Receiver signal level, during the first 4ms of signal reception, referenced to full scale (dB). Note: This value is only relevant when the system is in the Tx Signal Level Test Mode.
F32	rxLevel	Average Receiver signal level, across entire range, referenced to full scale (dB)

The UUV-3500 Diagnostics 1 record contains the results of the last self-test diagnostics run on the system.

Data type	**Parameter**	**units**
SDFX record header	recordHeader	The SDFX Record header recordId = 0x10 recordVersion = 1
U32	pingNumber	Latest ping number for which results were computed
U32	rxGainLevel	Receiver gain setting: 0 = Low (0dB), 1 = High (+12dB)
U32	rxGainMode	Receiver gain mode: 0 = auto, 1 = flat low, 2 = flat high
U32	txWaveform	Transmit waveform setting (same as SDF header)
U32	commMask	Communications results mask Bit 0: CDI3000 passed communication test Bit 1: Rx passed communication test Bit 2: Tx passed communication test This mask is updated once at system power on. To update this value, the sonar must be instructed to run the communications test again.
KLEIN_RX_SIG_LEVEL_1	chan[8]	Signal level results are computed for every sonar ping.

3.3.6. POS MV Data Records

The Applanix POS MV™ data records are stored in the SDFX. The TPU accepts multiple message types from the POS MV. Each of these messages has some common information such as group number and message timestamp. The common information is grouped together in a common data structure followed by the unique data for that message.

3.3.6.1. POS MV GROUP COMMON structure

Data type	**Parameter**	**Description**
SDFX Record Header	recordHeader	The SDFX Record header

U16	tpuMsgRcvdTimeRefCount	TPU Reference count value for the POS MV message that was used to generate this message
U32	tpuUdpMessageCount	TPU UDP message counter. Increments each UDP message.
U16	dataSource	POS MV Group number that data was acquired from
U8	timeType	POS specific bitmask
U8	timeSelect	Selects which time in this message to use for synchronization 1 = use time1 field of this message 2 – use time2 field of this message
U8	pad[2]	c
F64	time1	seconds
F64	time2	seconds

3.3.6.2. Record derived from POS MV Group 1

Data type	Parameter	Description
POS MV GROUP COMMON	common	Data common to all POS MV groups – includes SDFX record header recordId = 0x101 recordVersion = 1
F32	vesselSpeed	m/s
F32	vesselHeading	degrees = from "Vessel track angle"
F64	lat	Latitude, degrees
F64	lon	Longitude, degrees
F64	roll	Roll, degrees
F64	pitch	Pitch, degrees

3.3.6.3. Record derived from POS MV Group 102 or 103

Data type	Parameter	Description
POS MV GROUP COMMON	common	Data common to all POS MV groups – includes SDFX record header recordId = 0x166 (Group 102) or 0x167 (Group 103) recordVersion = 1
F64	lat	Latitude, degrees
F64	lon	Longitude, degrees
F64	roll	Roll, degrees
F64	pitch	Pitch, degrees
F64	heading	Heading, degrees
F32	heave	Heave, m (positive down)

3.3.6.4. Record derived from POS MV Group 112

Data type	Parameter	Description
POS MV GROUP COMMON	common	Data common to all POS MV groups – includes SDFX record header recordId = 0x170 recordVersion = 1
U32	timeSource	Source of UTC data and time in this message 0 = $--ZDA
U32	year	Year
U32	month	Month, 01 to 12
U32	day	Day, 01 to 31
U32	hour	UTC hour
U32	minute	UTC minute
U32	second	UTC second

3.3.7. *Record Sound Velocity Profile 1*

The Sound Velocity Profile 1 record is a series of depth and sound speed pairs.
The following special type (svpEntry) defines the depth/sound speed pair.

Data type	Parameter	units
F32	depth	m
F32	soundSpeed	m/s

Data type	Parameter	units
SDFX record header	recordHeader	The SDFX Record header recordId = 0x200 recordVersion = 1
U32	year	year (includes century)
U32	month	month (1-12)
U32	day	day (1-31)
U32	hour	hour (0-23)
U32	minute	minute (0-59)
F32	seconds	seconds
U32	type	User defined value that can describe the type of profile. For example, 0 could indicate the data is raw sensor while 1 could describe that the data has been processed.
U32	numChars	Number of characters in description string
U32	numPairs	Number of depth and sound speed pairs.
U8	description[]	A UTF-8 null terminated string describing this profile
svpEntry	svpRecord[]	Variable length array of depth/soundSpeed pairs

3.3.8. *NMEA Data Record 1*

The NMEA Data Record 1 is a structure that stores data input to the TPU via serial NMEA-0183 messages. The structure encompasses all fields that are parsed by the TPU. The structure indicates a bit mask word to indicate which of the values in the structure are valid for the specific record.

Data type	Parameter	Description
SDFX Record Header	recordHeader	The SDFX Record header recordId = 0x201 recordVersion = 1
U16	tpuMsgRcvdTimeRefCount	TPU Reference count value for the NMEA-0183 message that was used to generate this record
U32	tpuNmeaMessageCount	TPU NMEA message counter. Increments each NMEA message.
U16	dataSource	NMEA message type that generated this data: 0 = $--RMC 1 = $--GLL 2 = $--GGA 3 = $--VTG 4 = $--TLL 5 = $PAUV 6 = $PTSAG 7 = $--ZDA
U32	updateMask	Each bit in this mask indicates the data field(s) that are updated with the NMEA input that caused this data record Bit 0 = fixTime* fields valid Bit 1 = shipLat/shipLon valid

		Bit 2 = fishLat/fishLon valid Bit 3 = speed valid Bit 4 = shipHeading valid Bit 5 = magneticVariation valid Bit 6 = aux* fields valid
U32	fixTimeYear	Time from last serial NMEA message, year
U32	fixTimeMonth	Time from last serial NMEA message, month
U32	fixTimeDay	Time from last serial NMEA message, day
U32	fixTimeHour	Time from last serial NMEA message, hour
U32	fixTimeMinute	Time from last serial NMEA message, minute
F32	fixTimeSecond	Time from last serial NMEA message, second
F64	shipLat	from serial NMEA, radians
F64	shipLon	from serial NMEA, radians
F64	fishLat	from serial NMEA, radians
F64	fishLon	from serial NMEA, radians
F32	speed	from serial NMEA, m/s
F32	shipHeading	from serial NMEA – Course Over Ground, Degrees
F32	magneticVariation	from serial NMEA $--RMC message
F32	auxPitch	from $PAUV message
F32	auxRoll	from $PAUV message
F32	auxDepth	from $PAUV message
F32	auxAlt	from $PAUV message

3.3.9. System 5900 Sensor Data Record 1

The System 5900 Sensor Data Record 1 is a structure that stores data recorded from the System 5900 Sensor Interface board. The structure encompasses all fields that can be generated by the System 5900 Sensor Interface board. The structure indicates a bit mask word to indicate which of the values in the structure are valid for the specific record.

Data type	**Parameter**	**Description**
SDFX Record Header	recordHeader	The SDFX Record header recordId = 0x202 recordVersion = 1
U16	tpuMsgRcvdTimeRefCount	TPU Reference count value for the Sensor Interface message that was used to generate this record
U32	tpuSensorMessageCount	TPU Sensor message counter. Increments each Sensor Interface board message.
U32	updateMask	Each bit in this mask indicates the data field(s) that are updated with the NMEA input that caused this data record Bit 0 = compass fields valid Bit 1 = altitude valid Bit 2 = pressure valid Bit 3 = airTemp valid Bit 4 = waterTemp valid Bit 5 = leak valid Bit 6 = voltage fields valid
U8	pad[2]	pad to keep following items long word aligned
F32	compassHeading	Heading from compass sensor, degrees
F32	compassPitch	Pitch from compass sensor, degrees
F32	compassRoll	Roll from compass sensor, degrees
F32	altitude	meters
F32	pressure	Volts FIXME5900. IS THIS REALLY VOLTS?
F32	airTemp	Air temperature, Degrees C

F32	waterTemp	Water temperature, Degrees C
U32	leak	0 = Ok 1 = Towfish leak detected
F32	maxV1	Voltage 1 max reading over sample interval
F32	minV1	Voltage 1 min reading over sample interval
F32	minV2	Voltage 2 max reading over sample interval
F32	minV2	Voltage 2 min reading over sample interval
F32	minV3	Voltage 3 max reading over sample interval
F32	minV3	Voltage 3 min reading over sample interval
F32	minV4	Voltage 4 max reading over sample interval
F32	minV4	Voltage 4 min reading over sample interval

3.3.10. Klein Motion Sensor Data Records
The Klein Motion Sensor (KMS) data records are stored in the SDFX.

3.3.10.1. Raw Klein Motion Sensor Record 1
The Raw Klein Motion Sensor (KMS) Record 1 contains the raw data packets that were generated by the KMS during this ping.

Data type	**Parameter**	**units**
SDFX record header	recordHeader	The SDFX Record header recordId = 0x300 recordVersion = 1
U16	badCrc	Data from sensor failed CRC check
U16	tpuMsgRcvdTimeRefCount	TPU Reference count value when 1st byte of sensor message received
U32	memsAccelLsbWeight	LSB weight of MEMS acceleration data in micro g's
U32	memsDegreeLsbWeight	LSB weight of MEMS roll, pitch, and yaw rates in micro degrees per second
U32	fogLsbWeight	LSB weight of FOG data in nano degrees per second
U32	imuTimeRefCountLsbWeight	LSB weight of IMU time reference counter in nanoseconds
U8	kms01Message[30]	The raw binary message from the KMS-01

3.3.10.2. Processed Klein Motion Sensor Record 1
The Processed Klein Motion Sensor (KMS) Record 1 contains the processed data that is derived from the raw KMS data.

Data type	**Parameter**	**units**
SDFX record header	recordHeader	The SDFX Record header recordId = 0x301 recordVersion = 1
S32	refSampleCountA	The time stamp of these data records in reference to the current ping in sample counts. For example, if the motion sensor message arrived 10 ms before the current ping, this value would be equal to (SampleFreq * -0.010). This value may be negative if the value occurred before the current ping. This value may be greater than the ping duration if it occurred after the ping completed. This time stamp applies to these values:

		roll
S32	refSampleCountB	Reserved.
U16	tpuMsgRcvdTimeRefCount	TPU Reference count value for the KMS message that was used to generate this processed record.
U16	kmsMessageCount	The 7-bit message count in the KMS message that generated this roll. Strictly for reference.
U32	messageNumber	Increments each processed data record
U16	mask	Bit mask that indicates what values are derived from motion sensor
S32	roll	The derived roll value in micro-radians
S32	reserved[8]	Reserved

3.3.11. Octopus F180 Attitude and Positioning System

The Octopus F180 Attitude and Positioning System data records are stored in the SDFX. The TPU accepts the F180 MCOM message packet and derives this record from the MCOM packet.

Data Type	**Parameter**	**Description**
SDFX record header	recordHeader	The SDFX Record header recordId = 0x400 recordVersion = 1
U16	tpuMsgRcvdTimeRefCount	TPU Reference count value for the MCOM packet that was used to generate this record
U32	tpuUdpMessageCount	TPU UDP message counter. Increments each UDP message.
U16	mcomMessageTime	MCOM message bytes 1 & 2. Time as milliseconds into the minute of GPS time. Units = 1ms. Range: 0 to 59,999ms.
U16	mcomMessageCount	MCOM message bytes 66 & 67. Cyclic packet counter.
U8	navigationStatus	MCOM message byte 21. Navigation status byte from the MCOM message packet.
U8	pad1	Pad to keep following doubles/floats long aligned.
F64	secondsOfWeek	The time the motion data is valid in seconds of the week. Adjusted for F180 sampling delays.
F64	lat	Latitude, degrees
F64	lon	Longitude, degrees
F32	roll	Roll, degrees, positive is port up, starboard down
F32	pitch	Pitch, degrees, positive is bow up, stern down
F32	heading	Heading, degrees
F32	onlineHeave	Short-term Heave, meters, positive up
F32	longPeriodHeave	Long-term Heave, meters, positive up
U32	statusChannel10Valid	Indicates the gpsTime in this record is valid when set
U32	gpsTime	Time in minutes since GPS began from Status Channel 0.
U32	statusChannel40Valid	Indicates the mcomVersion, mcomOutputDelay, iHeaveLatency, and iHeaveFilterPeriod values in this record are valid when set.
U16	mcomVersion	MCOM Version from Status Channel 40
U16	mcomOutputDelay	MCOM Output delay from Status Channel 40
U16	iHeaveLatency	iHeave (Long-period Heave) latency from Status Channel 40
U16	iHeaveFilterPeriod	iHeave (Long-period Heave) filter period from Status Channel 40

3.3.12. Teledyne TSS DMS-XX Dynamic Motion Sensors

The Teledyne TSS DMS-XX Dynamic Motion Sensor data records are stored in the SDFX. The TPU accepts the sensor's TSS1 message packet and derives this record from the TSS1 data packet.

Data Type	**Parameter**	**Description**
SDFX record header	recordHeader	The SDFX Record header recordId = 0x500 recordVersion = 1
U16	tpuMsgRcvdTimeRefCount	TPU Reference count value for the TSS1 packet that was used to generate this record
U32	tpuMessageCount	TPU TSS1 message counter. Increments each TSS1 message.

U8	tssStatus	Status flag from TSS1 message:
		`0 = 'U' - "Unaided Mode - Settled Condition"`
		`1 = 'u' - "Unaided Mode - Settling"`
		`2 = 'G' - "GPS Aided Mode - Settled Condition"`
		`3 = 'g' - "GPS Aided Mode - Settling"`
		`4 = 'H' - "Heading Aided Mode - Settled Condition"`
		`5 = 'h' - "Heading Aided Mode - Settling"`
		`6 = 'F' - "Full Aided Mode - Settled Condition"`
		`7 = 'f' - "Full Aided Mode - Settling"`
		`F = Unknown.  Unrecognized status character`

U8	pad1	Pad to keep following doubles/floats long aligned.
F64	secondsOfWeek	The time the motion data is valid in seconds of the week. Adjusted for TSS1 transmission time.
F32	roll	Roll, degrees, positive is port up, starboard down
F32	pitch	Pitch, degrees, positive is bow up, stern down
F32	heave	heave, meters, positive up

3.3.13. OceanServer IMU Motion Sensor

The OceanServer Inertial Motion Unit sensor records are stored in the SDFX. The TPU accepts the sensor's serial data stream and derives this record from it.

Data Type	**Parameter**	**Description**
SDFX record header	recordHeader	The SDFX Record header recordId = 0x600 recordVersion = 1
S32	refSampleCountA	The time stamp of these data records in reference to the current ping in sample counts. For example, if the motion sensor message arrived 10 ms before the current ping, this value would be equal to (SampleFreq * -0.010). This value may be negative if the value occurred before the current ping. This value may be greater than the ping duration if it occurred after the ping completed. This time stamp applies to these values: roll, pitch, heading

S32	refSampleCountB	Reserved.
U32	tpuMsgRcvdTimeRefCount	TPU Reference count value for the character sequence that was used to generate this record
U32	messageNumber	Increments for each IMU message received
F32	heading	Heading, degrees
F32	pitch	Pitch, degrees, positive is bow up, stern down
F32	roll	Roll, degrees, positive is port up, starboard down

3.3.14. System 7180 Data Records

The NOAA LRSSS System 7180 raw data records are in SDF2 format. When the System 7180 data is processed with the Klein Bathymetric engine, the SDF2 data is converted to SDFX for output. A number of the SDF2 data records are converted to SDFX data records in the process. The SDFX data records are defined in this section along with a reference to the original SDF2 data structure.

3.3.14.1. Common SDF2 data types

There are a number of common data types used in the SDF2 format. These data types are defined here.

1. The FISHPACSECTIONHDR is converted directly to the SDFX Record header type. The FISHPACSECTIONHDR member “sectionNumRecords” is discarded.
2. Some members of the FISHPACKDATAPAGEHDR type do not apply in the SDFX data record, however, they are not modified and retain the value from the SDF2 structure, e.g., numDataSections and numBytes.
3. The FISHPACDATAPAGEHDR type is converted to the following structure:

Data type	Parameter	Description
SDFX Record Header	recordHeader	The SDFX Record header recordId = 0x7000 recordVersion = 1
U32	version	Header version
U32	pingNumber	
U32	year	Year of time stamp (time stamp of ping trigger from towfish)
U32	month	month of year (1 – 12)
U32	day	day of month (1 – 31)
U32	hour	hour of day (0 – 23)
U32	minute	minute of hour (0 – 59)
U32	second	second of minute (0 – 59)
U32	hSecond	hundredths of seconds (0 – 99)
F32	fseconds	fractional seconds, always less than 1
U32	range	meters
U32	errorFlags	
U32	speedSound	
U32	txWaveform	byte0 = MBSS TX_X, byte1 = MBSS TX_Y, byte 2 = MBES TX_X byte 3 = MBES TX_Y Note: MBSS/MBES specify which bottle the Tx board is in, NOT functionality! Note: byte 0 is least significant byte
U32	tvgPage	byte0 = TVG_A, byte1 = TVG_B, etc.
F32	depth	from towfish – not used
F32	altitude	meters, from towfish – calculated from down looker.
U32	dataQuality	Processed data quality flag, Bit 0 = Port, Bit 1 = Stbd, Bit 2 = NAS. Bit set indicates saturated data.
U32	numDataSections	Total number of Data sections in this Data Page.

U32	numSensorSections	Total number of Sensor sections in this Data Page
U32	numBytes	Total number of bytes in this Data Page.
U32	numBytesLastPage	Total number of bytes in the last Data Page.
U32	txWaveformMaskMbss	MBSS bottle Tx Waveform mask. 0 = Tx Off. 16 LSBs = TX_X, 16 MSBs = TX_Y.
U32	txWaveformMaskMbes	MBES bottle Tx Waveform mask. 0 = Tx Off. 16 LSBs = TX_X, 16 MSBs = TX_Y.
		All swVersion fields have the format 0xVVNNBBBB, where VV = Major Version Number NN = Minor Version Number BBBB = Build number
U32	swVersionMbssTpu	MBSS TPU software version
U32	swVersionSonarProSdf2	SonarPro SDF2 software version
U32	swVersionMbssBf	MBSS Beamformer software version
U32	swVersionNasBf	NAS Beamformer software version
U32	calConfigFileNum	The hexadecimal calibration configuration file number to use for the towfish configuration that acquired this data.
U16	acqDelaySu1	The offset, in number of samples from sample 0 of the raw SU1 data, to the first valid MBSS, UL-ES and DL-ES sample. This is due to the MBSS trigger occurring after the beginning of the record.
U16	satDetBlankingPsBathy	The number of saturation counts to subtract from the computed P/S Bathy saturation count value.
U32	reserved[2]	reserve for future expansion

4. The FISHPACSENSORSSECTION type is converted to the following structure:

Data type	**Parameter**	**Description**
SDFX Record Header	recordHeader	The SDFX Record header recordId = 0x71XX, where XX is set to relevant sensor type, e.g., 02 for GPS sensor. recordVersion = 1
U32	type	SENSORTYPEVALUE from SDF2 format
U32	id	id of sensor, for case of multiple sensors of the same type
U32	subType	subtype of the sensor type, for case of multiple data for same type
U32	year	year of time stamp
U32	month	month of year (1 – 12)
U32	day	day of month (1 – 31)
U32	hour	hour of day (0 – 23)
U32	minute	minute of hour (0 – 59)
U32	second	second of minute (0 – 59)
U32	hSecond	hundredths of seconds (0 – 99)
F32	fseconds	fractional seconds, always less than 1
U32	numBytes	number of bytes of sensor data to follow

5. The OCTANSDATASECTION type is converted to the following structure:

Data type	**Parameter**	**Description**
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7100 recordVersion = 1
F32	heading	degrees
F32	pitch	degrees, + is bow down, - is bow up
F32	roll	degrees, + is CCW, - is CW

F32	surge	meters, + is forward, - is backward
F32	sway	meters, + left, - is right
F32	heave	meters, + is up, - is down
F32	speedX1	meters/sec, + is forward, - is backward
F32	speedX2	meters/sec, + is left, - is right
F32	speedX3	meters/sec, + is up, - is down
F32	latComp	Latitude Compensation, + is North, - is South
F32	latCompSpd	Latitude Compensation Speed, Knots
U32	status	32 bit status flag

6. The ACTUATOR_DATASECTION type is converted to the following structure:

Data type	Parameter	Description
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7101 recordVersion = 1
U32	actuatorType	1 = Pitch, 2 = Roll, 3 = Angle of Attack Wing
U32	isValid	Defines which fields contain valid data. 1 = valid commandPosition 2 = valid currentPosition 4 = valid limitSwitches 8 = valid anglePosition 16 = valid potentiometer[n] 32 = valid commandPosition is an incremental move. 64 = valid status
U32	mostRecent	Time stamp is valid for these items (Same bit values as isValid)
U32	commandPosition	The new motor position the actuator is going to
U32	currentPosition	The current motor position of the actuator
U32	limitSwitches	Sensor detect motor at position of limit switch
F32	anglePosition	Current angle position of control surface in degrees (for reference only)
U32	potentiometer[4]	Potentiometer value used to detect locations. (valid only for Angle of Attack Wing)
U32	status	

7. The ACTUATOR_INITIALIZATION_DATASECTION type is converted to the following structure:

Data type	Parameter	Description
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7102 recordVersion = 1
U32	actuatorType	1 = Pitch, 2 = Roll, 3 = Angle of Attack Wing
U32	isValid	Defines which fields contain valid data. 1 = valid holdCurrent 2 = valid moveCurrent 4 = valid startVelocity 8 = valid slewMaxSpeed_PositionMode 16 = valid stopSpeed 32 = valid speed_VelocityMode 64 = valid microStepSize_Correction

		128 = valid limitState 256 = valid potentiometerSettings[n] 512 = valid status
U32	mostRecent	Time stamp is valid for these items. (Same bit values as isValid)
U32	holdCurrent	
U32	moveCurrent	
U32	startVelocity	?1
U32	slewMaxSpeed_PositionMode	?2
U32	stopSpeed	?3
U32	speed_VelocityMode	?5
U32	microStepSize	?6
U32	microStepSize_Correction	?7
U32	limitState	Type of limit control used
U32	potentiometerSettings[4]	Only potentiometerSettings[3] for AOA wing is set (?at)
U32	status	

8. The DVLDATASECTION type is converted to the following structure:

Data type	**Parameter**	**Description**
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7103 recordVersion = 1
PD5HdrType	MsgHeader	
SysConfigType	SysConfig	
VelocityType	BottomVelocity	
U16	RngToBottomBeam1	0 – 65535 cm, 0 = Bad detection
U16	RngToBottomBeam2	0 – 65535 cm, 0 = Bad detection
U16	RngToBottomBeam3	0 – 65535 cm, 0 = Bad detection
U16	RngToBottomBeam4	0 – 65535 cm, 0 = Bad detection
BottomStatType	BottomStatus	
VelocityType	RefLayerVelocity	
U16	RefLayerStart	0-9999 dm
U16	RefLayerEnd	0-9999 dm
RefStatType	RefLayerStatus	
TimeOfPingType	TimeOfPing	
BITResultsType	BITResults	
U16	SpeedOfSound	1400-1600 m/s LSD 1 m/s
U8	Salinity	0-40 parts per thousand (ppt)
U16	Depth	1-999 dm
S16	Pitch	-20.00 - +20.00 degrees LSD .01
S16	Roll	-20.00 - +20.00 degrees LSD .01
U16	Heading	000.00 – 359.99 degrees LSD .01
DMGType	DMGBottom	DMG over the bottom
DMGType	DMGReference	DMG over the water mass reference
U16	Checksum	

Note: The PD5HdrType, SysConfigType, VelocityType, BotomStatType, RefStatType, TimeOfPingType, BITResultsType, and DMGType are identical to the SDF2 format. These structure types are repeated here for clarity.

PD5HdrType:

Data type	**Parameter**	**Description**
U8	Id	Always 0x7D
U8	Type	1 = DVL_PD5MsgType

U16	NumOfBytes	Not including the checksum

SysConfigType:

Data type	Parameter	Description
U8	Frequency	2 = 300 kHz 3 = 600 kHz 4 = 1200 kHz
U8	Unused	
U8	ThreeBeamSolutionsComputed	bool
U8	TiltInformationUsed	bool
U8	VelocityType	0 = beam coordinate velocities 1 = instrument coordinate velocities 2 = ship coordinate velocities 3 = earth coordinate velocities

VelocityType:

Data type	Parameter	Description
S16	X	Positive indicates East Range: -32768 – 32768 Resolution: mm/sec 0x8000 = bad velocity
S16	Y	Positive indicates North Range: -32768 – 32768 Resolution: mm/sec 0x8000 = bad velocity
S16	Z	Positive indicates Up Range: -32768 – 32768 Resolution: mm/sec 0x8000 = bad velocity
S16	E	

BottomStatType:

Data type	Parameter	Description
U8	status	Bit 0 = Beam1LowCorrelation (0=ok) Bit 1 = Beam1LowEchoAmplitude (0=ok) Bit 2 = Beam2LowCorrelation (0=ok) Bit 3 = Beam2LowEchoAmplitude (0=ok) Bit 4 = Beam3LowCorrelation (0=ok) Bit 5 = Beam3LowEchoAmplitude (0=ok) Bit 6 = Beam4LowCorrelation (0=ok) Bit 7 = Beam4LowEchoAmplitude (0=ok)

RefStatType:

Data type	Parameter	Description
U8	status	Bit 0: Beam1LowCorrelation (0=ok) Bit 1: Beam2LowCorrelation (0=ok) Bit 2: Beam3LowCorrelation (0=ok) Bit 3: Beam4LowCorrelation (0=ok) Bit 4: AltitudeTooShallow (0=ok) Bits 7-5: = Unused

TimeOfPingType:

Data type	Parameter	Description
U8	Hour	
U8	Minute	

U8	Second	
U8	SecHundredth	

BITResultsType:

Data type	Parameter	Description
U16	results	Bit 7-0: Unused Bit 8: Reserved1 Bit 9: DSPError Bit 10: Reserved2 Bit 11: Demod0Error Bit 12: Demod1Error Bit 13: Reserved3 Bit 14: Reserved4 Bit 15: Reserved5

DMGType: Distance Made Good

Data type	Parameter	Description
S32	East	-10,000,000 - +10,000,000 mm LSD 1mm
S32	North	-10,000,000 - +10,000,000 mm LSD 1mm
S32	Up	-10,000,000 - +10,000,000 mm LSD 1mm
S32	Error	

9. The GPSDATASECTION type is converted to the following structure:

Data type	Parameter	Description
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7104 recordVersion = 1
float	speed	from GPS, updated on GPS update, m/s
float	shipHeading	from GPS, degrees
float	magneticVariation	from GPS, degrees
double	shipLat	from GPS, radians
double	shipLon	from GPS, radians
double	fishLat	radians
double	fishLon	radians

10. The SVP sensor data is converted to the following structure:

Data type	Parameter	Description
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7105 recordVersion = 1
char	buff[128]	SVP ASCII record

11. The Transmissometer sensor data is converted to the following structure:

Data type	Parameter	Description
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7106 recordVersion = 1
char	buff[128]	Transmissometer ASCII data

12. The Depth sensor data is converted to the following structure:

Data type	Parameter	Description
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7107 recordVersion = 1
char	buff[128]	Depth ASCII data

13. The CABLEOUTDATASECTION type is converted to the following structure:

Data type	Parameter	Description
FISHPACSENSORSSECTION	hdr	The SDFX Record header recordId = 0x7108 recordVersion = 1
char	DeviceName[80]	Name of the cable out device (NULL terminated)
float	gpsHeight	User selected. (meters)
float	sheaveOffset_X	User selected. (meters) (Positive toward aft)
float	sheaveOffset_Y	User selected. (meters) (Positive toward starboard)
float	sheaveOffset_Z	User selected (meters)
float	cableOut_ScaleFactor	cable out = (cableOut_ScaleFactor * cableOut) + cableOut_Offset
float	cableOut_Offset	User selected (meters)
float	cableOut	Amount of cable out (in meters. Already adjusted by scale factor and offset.)

3.3.15. System 5900 – Side Scan and Bathy Raw Data Page (64-Channel, pageVersion 5900)

The System 5900 Side Scan and Bathy Raw Data (64-Channel) page is utilized for 5900 towfish that are equipped with 64-channels dedicated for Side Scan and bathymetry. If the System 5900 is equipped with additional channels for other purposes, for example, a gap filler, the other channel data is defined elsewhere.
The pageVersion for this data page type is 5900.

The System 5900 Side Scan and Bathy Raw Data (64-Channel) data channel structure is as follows:

Table 17 - System 5900 Side Scan and Bathy Raw (64-channel) data channel structure

Vector Number	Type	Name	Description
1	long	rawdataPort1i[]	1-32 raw port data
2	long	rawdataPort1q[]	
3	long	rawdataPort2i[]	
4	long	rawdataPort2q[]	
5	long	rawdataPort3i[]	
6	long	rawdataPort3q[]	
7	long	rawdataPort4i[]	
8	long	rawdataPort4q[]	
9	long	rawdataPort5i[]	
10	long	rawdataPort5q[]	
11	long	rawdataPort6i[]	
12	long	rawdataPort6q[]	
13	long	rawdataPort7i[]	
14	long	rawdataPort7q[]	
15	long	rawdataPort8i[]	
16	long	rawdataPort8q[]	

17	long	rawdataPort9i[]	
18	long	rawdataPort9q[]	
19	long	rawdataPort10i[]	
20	long	rawdataPort10q[]	
21	long	rawdataPort11i[]	
22	long	rawdataPort11q[]	
23	long	rawdataPort12i[]	
24	long	rawdataPort12q[]	
25	long	rawdataPort13i[]	
26	long	rawdataPort13q[]	
27	long	rawdataPort14i[]	
28	long	rawdataPort14q[]	
29	long	rawdataPort15i[]	
30	long	rawdataPort15q[]	
31	long	rawdataPort16i[]	
32	long	rawdataPort16q[]	
33	long	rawdataPort17i[]	
34	long	rawdataPort17q[]	
35	long	rawdataPort18i[]	
36	long	rawdataPort18q[]	
37	long	rawdataPort19i[]	
38	long	rawdataPort19q[]	
39	long	rawdataPort20i[]	
40	long	rawdataPort20q[]	
41	long	rawdataPort21i[]	
42	long	rawdataPort21q[]	
43	long	rawdataPort22i[]	
44	long	rawdataPort22q[]	
45	long	rawdataPort23i[]	
46	long	rawdataPort23q[]	
47	long	rawdataPort24i[]	
48	long	rawdataPort24q[]	
49	long	rawdataPort25i[]	
50	long	rawdataPort25q[]	
51	long	rawdataPort26i[]	
52	long	rawdataPort26q[]	
53	long	rawdataPort27i[]	
54	long	rawdataPort27q[]	
55	long	rawdataPort28i[]	
56	long	rawdataPort28q[]	
57	long	rawdataPort29i[]	
58	long	rawdataPort29q[]	
59	long	rawdataPort30i[]	
60	long	rawdataPort30q[]	
61	long	rawdataPort31i[]	
62	long	rawdataPort31q[]	
63	long	rawdataPort32i[]	
64	long	rawdataPort32q[]	
65	long	rawdataStbd1i[]	1-32 raw stbd data
66	long	rawdataStbd1q[]	
67	long	rawdataStbd2i[]	
68	long	rawdataStbd2q[]	
69	long	rawdataStbd3i[]	
70	long	rawdataStbd3q[]	

71	long	rawdataStbd4i[]	
72	long	rawdataStbd4q[]	
73	long	rawdataStbd5i[]	
74	long	rawdataStbd5q[]	
75	long	rawdataStbd6i[]	
76	long	rawdataStbd6q[]	
77	long	rawdataStbd7i[]	
78	long	rawdataStbd7q[]	
79	long	rawdataStbd8i[]	
80	long	rawdataStbd8q[]	
81	long	rawdataStbd9i[]	
82	long	rawdataStbd9q[]	
83	long	rawdataStbd10i[]	
84	long	rawdataStbd10q[]	
85	long	rawdataStbd11i[]	
86	long	rawdataStbd11q[]	
87	long	rawdataStbd12i[]	
88	long	rawdataStbd12q[]	
89	long	rawdataStbd13i[]	
90	long	rawdataStbd13q[]	
91	long	rawdataStbd14i[]	
92	long	rawdataStbd14q[]	
93	long	rawdataStbd15i[]	
94	long	rawdataStbd15q[]	
95	long	rawdataStbd16i[]	
96	long	rawdataStbd16q[]	
97	long	rawdataStbd17i[]	
98	long	rawdataStbd17q[]	
99	long	rawdataStbd18i[]	
100	long	rawdataStbd18q[]	
101	long	rawdataStbd19i[]	
102	long	rawdataStbd19q[]	
103	long	rawdataStbd20i[]	
104	long	rawdataStbd20q[]	
105	long	rawdataStbd21i[]	
106	long	rawdataStbd21q[]	
107	long	rawdataStbd22i[]	
108	long	rawdataStbd22q[]	
109	long	rawdataStbd23i[]	
110	long	rawdataStbd23q[]	
111	long	rawdataStbd24i[]	
112	long	rawdataStbd24q[]	
113	long	rawdataStbd25i[]	
114	long	rawdataStbd25q[]	
115	long	rawdataStbd26i[]	
116	long	rawdataStbd26q[]	
117	long	rawdataStbd27i[]	
118	long	rawdataStbd27q[]	
119	long	rawdataStbd28i[]	
120	long	rawdataStbd28q[]	
121	long	rawdataStbd29i[]	
122	long	rawdataStbd29q[]	
123	long	rawdataStbd30i[]	
124	long	rawdataStbd30q[]	

125	long	rawdataStbd31i[]	
126	long	rawdataStbd31q[]	
127	long	rawdataStbd32i[]	
128	long	rawdataStbd32q[]	

3.3.16. System 5900 – Side Scan and Bathy Pulse Compressed Data Page (64-Channel pageVersion 5901)

The System 5900 Side Scan and Bathy Pulse Compressed Data (64-Channel) page is the output data page type when a System 5900 Side Scan and Bathy Raw Data Page is run thru a pulse compression engine. The pageVersion for this data page type is 5901.

The System 5900 Side Scan and Bathy Raw Data (64-Channel) data channel structure is as follows: Reserved.

3.3.17. System 5900 – Side Scan QC Beamformed Data Page (pageVersion 5902)

The System 5900 Side Scan QC Beamformed data page is the output data page type when the System 5900 Side Scan channels are beamformed into a single QC "beam" image. The pageVersion for this data page type is 5902.

The System 5900 Side Scan QC Beamformed data channel structure is as follows:

Table 18 - System 5900 Side Scan and Bathy Raw (64-channel) data channel structure

Vector Number	Type	Name	Description
1	float	portBeam[]	Processed side scan data
2	float	stbdBeam[]	

3.3.18. System 5900 – Bathy Pulse Compressed Data Page (For Side Scan Sonar, 4-channel, pageVersion 5903)

The System 5900 Bathy Pulse Compressed data page contains the pulse compressed version of the raw side scan bathymetry channels (staves). This data page may be used directly for bathymetric processing without the need for performing pulse compression (match filtering). The pageVersion for this data page type is 5903.

The System 5900 Bathy Pulse Compressed Beamformed data channel structure is as follows:

Table 19 - System 5900 Bathy Pulse Compressed Data Page (4-channel) data channel structure

Vector Number	Type	Name	Description
1	float	portBathy1i[]	
2	float	portBathy1q[]	
3	float	portBathy2i[]	
4	float	portBathy2q[]	
5	float	portBathy3i[]	
6	float	portBathy3q[]	
7	float	portBathy4i[]	
8	float	portBathy4q[]	
9	float	stbdBathy1i[]	
10	float	stbdBathy1q[]	
11	float	stbdBathy2i[]	
12	float	stbdBathy2q[]	
13	float	stbdBathy3i[]	
14	float	stbdBathy3q[]	
15	float	stbdBathy4i[]	
16	float	stbdBathy4q[]	

3.3.19. System 5900 Gap Filler and Bathy Raw Data Page (64-Channel, pageVersion 5905)

The System 5900 Gap Filler and Bathy Raw Data (64-Channel) page is utilized for 5900 towfish that are equipped with 64-channels dedicated for Gap Filler and bathymetry. The Gap Filler data page is separate from the side scan..
The pageVersion for this data page type is 5905.

The System 5900 Gap Filler and Bathy Raw Data (64-Channel) data channel structure is as follows:

Table 20 - System 5900 Side Scan and Bathy Raw (64-channel) data channel structure

Vector Number	Type	Name	Description
TBD			

3.3.20. System 5900 Gap Filler and Bathy Pulse Compressed Data Page (64-Channel, pageVersion 5906)

The System 5900 Gap Filler and Bathy Pulse Compressed Data (64-Channel) page is the output data page type when a System 5900 Gap Filler and Bathy Raw Data Page is run thru a pulse compression engine.
The pageVersion for this data page type is 5906.
The System 5900 Gap Filler and Bathy Pulse Compressed Data (64-Channel) data channel structure is as follows:
TBD

3.3.21. System 5900 Side Scan Processed Ping Data Page (pageVersion 5910)

The System 5900 Side Scan Processed page is utilized for beamformed data from a 5900 towfish. This data has NOT been motion compensated.
The pageVersion for this data page type is 5910.

The System 5900 Processed Ping data channel structure is as follows:

Table 21 - System 5900 Processed Ping data channel structure

Vector Number	Type	Name	Description
1	`float`	`portBeam[]`	`Processed side scan data`
2	`float`	`stbdBeam[]`	

3.3.22. System 5900 Bathy Processed (For Side Scan) Data Page (pageVersion 5911)

The System 5900 Bathy Processed (For Side Scan) page contains the processed bathymetry data based on the raw data bathymetry channels in the side scan arrays.
The pageVersion for this data page type is 5911.

Bathymetry Notes:

- A processed bathy value equal to -32768 (Hex 0x8000) is defined as Not A Number (NAN) and should be ignored.
- Range to sea bed detection is computed as (vector index/samplesFreq) * 750 m/s. The sampleFreq is from the SDF header.
- The angle of arrival is relative to the transducer face.
- Across track offsets (Y) are positive to starboard and negative to port.
- Depth values (Z) are relative to the sonar transducers. The values are positive and increase as the distance from the towfish to the seafloor increases.
- The rollVector array contains a roll correction value for each sample in the bathyPortAngle and bathyStbdAngle arrays. The roll vector is generated based on the motion correction source selected by the user. The motion correction source is recorded in the Bathy Processed Settings

Record 1. To obtain the roll corrected angle of arrival, the rollVector should be subtracted from the bathyPortAngle array and added to the bathyStbdAngle array.

- The pitchVector and heaveVector arrays contain a pitch and heave value for each bathymetry sample if the user selected a motion sensor capable of pitch and heave measurement. These vectors may be used to apply lever arm corrections. The Ship Configuration Info 1 record contains the relative offsets between the sonar and the sensors.
- Along track offsets (X), Across track offsets (Y) and Depth values (Z) are computed with the roll vector applied to the bathyPortAngle and bathyStbdAngle arrays. The offsets and depth values may be computed with lever arm and/or sound velocity correction (user selectable). The SDF header value postProcessVersion indicates which processing was applied.

The System 5900 Bathy Processed (For Side Scan) data channel structure is as follows:

Table 22 - 5900 Bathy Processed (For Side Scan) data channel structure

Vector Number	Type	Name	Description
1	short	bathyPortIntensity[]	Port "Beamformed" bathy backscatter intensity
2	short	bathyPortAngle[]	Best angle of arrival port * bathyScaleAngle (based upon SDF header speedSound sound speed)
3	short	bathyPortQuality[]	Quality associated with angle arrival (Quality * bathyScaleQuality)
4	short	bathyPortX[]	Along track port bathymetry * bathyScaleXYZ
5	short	bathyPortY[]	Horizontal (across track) port bathymetry, relative to towfish * bathyScaleXYZ
6	short	bathyPortZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
7	short	bathyStbdIntensity[]	Stbd "Beamformed" bathy backscatter intensity
8	short	bathyStbdAngle[]	Best angle of arrival stbd * bathyScaleAngle (based upon SDF header speedSound sound speed)
9	short	bathyStbdQuality[]	Quality associated with angle arrival (Quality * bathyScaleQuality)
10	short	bathyStbdX[]	Along track stbd bathymetry *

			bathyScaleXYZ
11	short	bathyStbdY[]	Horizontal (across track) stbd bathymetry, relative to towfish * bathyScaleXYZ
12	short	bathyStbdZ[]	Vertical (depth) bathymetry, relative to towfish * bathyScaleXYZ
13	short	rollVector[]	Bathy Roll correction applied to angle for X,Y,Z calculations * bathyRollScale.
14	short	pitchVector[]	Bathy Pitch correction applied to angle for X,Y,Z calculations * bathyPitchScale.
15	short	heaveVector[]	Bathy Heave correction applied to angle for X,Y,Z calculations * bathyHeaveScale. (heave at center point between port/stbd transducers)
16	short	bathyPortSNR[]	Port signal-to-noise ratio * bathyScaleSNR
17	uns’nd short	bathyPortUncertainty[]	Port uncertainty * bathyScaleUncertainty
18	short	bathyStbdSNR[]	Starboard signal-to-noise ratio * bathyScaleSNR
19	uns’nd short	bathyStbdUncertainty[]	Starboard uncertainty * bathyScaleUncertainty
20	short	reserved1[]	Future, reserved
21	short	reserved2[]	Future, reserved

3.3.23. System 5900 Gap Filler Processed Ping Data Page (pageVersion 5915)

The System 5900 Gap Filler Processed page contains the output data tiles from the gap filler processing based on the raw gap filler sonar data from a 5900 towfish.
The pageVersion for this data page type is 5915.

The System 5900 Gap Filler Processed Ping data channel structure is as follows:

Table 23 - System 5900 Gap Filler Processed Ping data channel structure

Vector Number	Type	Name	Description
TBD			

3.3.24. System 5900 Bathy Processed (For Gap Filler) Data Page (pageVersion 5916)

The System 5900 Bathy Processed (For Gap Filler) page contains the processed bathymetry data based on the raw data bathymetry channels in the Gap Filler arrays.
The pageVersion for this data page type is 5916.

The System 5900 Bathy Processed (For Gap Filler) data channel structure is as follows:

Table 24 - 5900 Bathy Processed (For Gap Filler) data channel structure

Vector Number	Type	Name	Description
TBD			

3.3.25. System 5900 Side Scan Processed Ping Data Page (pageVersion 5920)

The System 5900 Side Scan Processed page is utilized for beamformed data from a 5900 towfish. This data has been motion compensated.
The pageVersion for this data page type is 5920.

The System 5900 Processed Ping data channel structure is as follows:

Table 25 - System 5900 Processed Ping (motion compensated) data channel structure

Vector Number	Type	Name	Description
1	`float`	`portBeam[]`	`Processed side scan data`
2	`float`	`stbdBeam[]`	

4. SDF Files

The SonarPro® generated Sonar Data Files (SDF) have an ".sdf" file extensions. These files are in the form; [data page][data page] … etc – where each data page is the ping marker followed by the SDF data page as described in section 3. The ping marker is a 32-bit value that never changes and is equal to 0xFFFFFFFF (2^32-1).

The SonarPro® generated SDF files may also contain SDFX data structures. Each SDFX structure that contains setting information will appear in the first ping of data in the SDF file and in any subsequent ping when those settings change. For example, the scale values for the bathymetry angle of arrival vectors and the roll vector are contained in the "Bathy Processed Settings Record 1".

5. Additional SDF pageVersion Information

The pageVersion in the SDF header defines the size of the header and the "data channel" layout of the page. For a given system configuration, the TPU will generate a particular SDF pageVersion. Subsequent data processing by SonarPro® or the KleinSDK may modify the pageVersion if the processing changes the SDF data page format. At this time, only System 5000 data pages can get modified by post processing. Table 26 provides additional information on the pageVersions generated by the TPU, the TPU vxWorks version, system configurations, and various notes on which pageVersion is generated by the TPU and how the pageVersion can be modified by post processing the data page with SonarPro® or the KleinSDK.

Table 26 - SDF pageVersion additional information

pageVersion	Header Type	Header Size (bytes)	vxWorks version	System Configuration	TPU behavior and modifications caused by Post Processing by SonarPro®/KleinSDK
5000	3	256	5.40	5000 raw data system	• Beamforming[1] converts to 5001 • Bathymetric processing converts to 5002
5001	4	512	6.12 and later	5000 raw data system, 5000 V2 system	• Bathymetric processing converts to 5002
5002	4	512	7.00 and later	5000 raw data system, 5000 V2 system	• TPU generates this page format when towfish contains Klein Motion Sensor • TPU generates this page format when the system is set for raw bathymetric channel acquisition • This data page format is output of the SonarPro®/KleinSDK bathymetric processing engine
5003	4	512	7.00 and later	5000 V2 system	• System contains 3 dedicated bathymetric transducer "strings"
5004	4	512	8.06 and later	5000 V2 system	• System contains 3 dedicated bathymetric transducer "strings"

1. Beamforming by the TPU DSP does not convert the data page format to 5001. This applies only to beamforming done by SonarPro® or the KleinSDK.

6. Interpreting Klein Sonar Bathymetric data

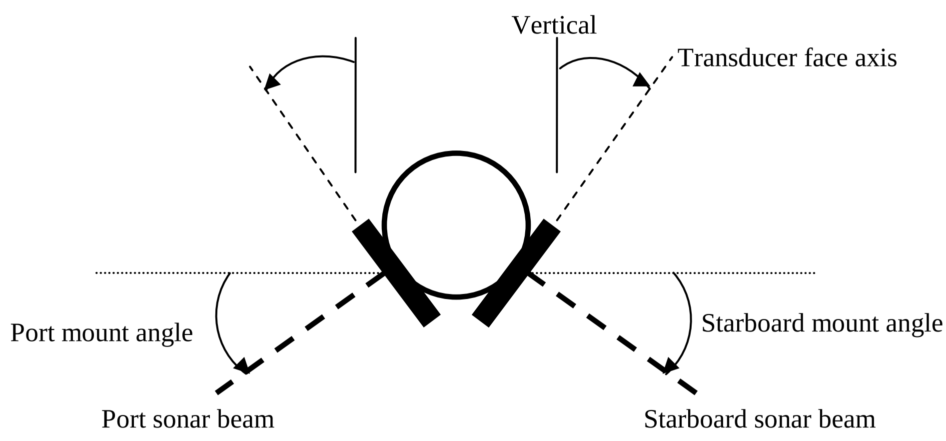
6.1. General Information

The Klein Bathymetry processing engine is used to generate the bathymetry data that is stored in the SDF records. The processing engine has a number of user configurable parameters as well as some system installation dependent settings. These settings are represented in either the SDF header record or in SDFX structures within each file or ping. The following list is a summary of these settings and other general information on interpreting the SDF bathymetry records.

1. The SDF header capabilityMask has a bit indicating whether the system is towed of hull (fixed) mounted.
2. The SDFX structures contain critical settings for the data processing. When the SDF records are generated by SonarPro®, each SDFX structure that contains setting information will appear in the first ping of data in the SDF file and in any subsequent ping that those settings change. For example, the scale values for the angle of arrival vectors are contained in the “Bathy Processed Settings Record 1”.
3. The SDFX structure “Bathy Processed Settings Record 1” has two members that indicate what type of correction has been applied to the output data; “bathyMotionType” and “bathySoundSpeedType”.
4. The SDF header postProcessVersion value indicates what type of processing was done to the data, for example, whether lever arm or sound speed correction was applied.
5. The SDFX structure “Ship Configuration Info 1” contains lever arm values, transducer biases, etc. These values can be used by the Klein Bathy processing engine or 3rd party software to apply motion and sound speed correction.
6. The calculated bathymetry depth (Z values) arrays are relative to the sonar transducers. To get true depth, it is necessary to account for the depth of the sonar itself. In the case of a fixed mount system, use the draft value in the “Ship Configuration Info 1” SDFX structure. In the case of a towed system, the SDF header pressure sensor value should be used. This value is in volts and must be converted to a distance value (meters) before adding to the bathymetry depth array. The conversion formula of volts to meters is given elsewhere in this document.
7. A fixed (hull) mount system is likely using a sensor such as the Applanix POS MV. In this case, the navigation information such as latitude, longitude, course, etc., is not filled into the SDF header. It is present in one of the SDFX POS MV data structures. The header value “motionSensorType” with a value of 2 indicates that a POS MV is being used.

6.2. Transducer mount angle

The port and starboard sonar transducers are mounted at an angle so that their beams, perpendicular to the transducer faces, are directed downward.



The *transducer mount angle* is the angle by which the transducer is rotated *downwards*, i.e., the angle by which the beam points downwards relative to the horizontal, or the angle by which the transducer face is rotated relative to the vertical. For both port and starboard sides, positive angles denote downward-pointing sonar beams. Thus, the starboard mount angle measures the clockwise rotation of the starboard transducer as viewed from the rear; the port mount angle measures the *counterclockwise* rotation of the port transducer.

6.3. Bathymetric data values in SDF files

In a SDF file, a ping record that contains bathymetric data contains the following data arrays, each containing a value for each sample in the ping.

- Port angle
- Port quality
- Port X (alongtrack)
- Port Y (acrosstrack)
- Port Z (depth)
- Starboard angle
- Starboard quality
- Starboard X (alongtrack)
- Starboard Y (acrosstrack)
- Starboard Z (depth)
- Roll Vector
- Pitch Vector
- Heave Vector

These are arrays of signed 16-bit integers, so the values must be multiplied by appropriate scale factors to obtain the true sample values. The applicable scale factors may be found in the "Bathy Processed Settings Record 1" record, typically found in the first ping record of the SDF file. At the time of this writing, the scale factor values in use are:

Parameter	Array values
Angle	Radians x 10000; radians x 10000 / 2π for early SDF files. If the *postProcessVersion* word in the SDF ping header has the NGS_POST_PROCESS_5000_BATHYV2 (0x10) bit set, the scale factor is radians x 10000.
Quality	Percent x 100
X, Y, Z	Meters x 100 (centimeters)
Roll, Pitch	Radians X 1000
Heave	Meters x 100 (centimeters)

Additional comments on the data arrays:

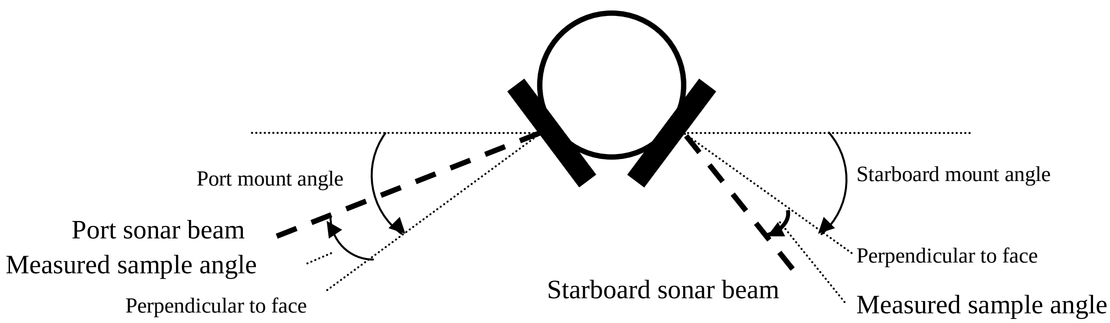
- A processed bathy value equal to -32768 (Hex 0x8000) is defined as Not A Number (NAN) and should be ignored.
- Across track offsets (Y) are positive to starboard and negative to port.
- Depth values (Z) are relative to the sonar transducers. The values are positive and increase as the distance from the towfish to the seafloor increases.
- The angle of arrival vectors are not corrected for roll.
- The rollVector array contains a roll correction value for each sample in the bathyPortAngle and bathyStbdAngle arrays. The roll vector is generated based on the motion correction source selected by the user. The motion correction source is recorded in the Bathy Processed Settings Record 1. To obtain the roll corrected angle of arrival, the rollVector should be subtracted from the bathyPortAngle array and added to the bathyStbdAngle array. Note: The units for the rollVector and the angle vectors are not the same. The values must be converted to a common unit before adding them together.

- The pitchVector and heaveVector arrays contain a pitch and heave value for each bathymetry sample if the user selected a motion sensor capable of pitch and heave measurement. These vectors may be used to apply lever arm corrections. The Ship Configuration Info 1 record contains the relative offsets between the sonar and the sensors.
- Along track offsets (X), Across track offsets (Y) and Depth values (Z) are computed with the roll vector applied to the bathyPortAngle and bathyStbdAngle arrays. The offsets and depth values may be computed with lever arm and/or sound velocity correction (user selectable). The SDF header value postProcessVersion indicates which processing was applied.
- The Quality vectors provide an indicator as to the quality of the solution. This is a value between 0 and 1 (once the scale value is applied). It is useful to provide this value as a "data cleaning" input to post process the data in order to discard outliers. There is not firm threshold value that can be used for all sea types and conditions. A value of 0.65 may be a valid default for this parameter. This value should be user adjustable when users are processing this data.

6.4. Angles in SDF file data

The angle values in a SDF ping record represent the angle of arrival *relative to the transducer face.* For example, an angle value of 0 corresponds to a beam perpendicular to the transducer. The angle values increase as the beam points increasingly downward, so that

- Positive angle values represent beam direction downward from the perpendicular, negative values represent beam direction upward from the perpendicular
- The angle values in an array are most positive for the innermost samples, most negative for the outermost samples.



In this figure, the starboard-side beam angle is positive, because the beam is below (increasing angle) the zero-angle reference - the line perpendicular to the transducer face. The port-side beam angle is negative, because the beam is above (decreasing) the line perpendicular to the transducer face.

6.5. Converting SDF angle values to absolute angle values

The SDF angle array values, relative to the transducer surface, are of limited usefulness. Most users will want to convert these to absolute angles relative to a horizontal or vertical reference. These are the necessary steps to do that.

1. Multiply the integer value in the array by an appropriate scale factor to obtain the angle in degrees. For example, if the angle array values represent radians x 10000, multiply each angle value by 1/10000 to obtain the angle in radians, then by 57.295779513082323 – the ratio of degrees per radian – to obtain the angle in degrees.
2. Add the appropriate transducer mount angle – the angle of the transducer face – to obtain the absolute beam angle relative to the horizontal.
3. If you need the beam angle relative to the vertical, subtract the result of step (2) from 90 degrees.

4. This results in angle values in the range of 0 to 90 degrees, for both port and starboard sides. If you use a signed angle convention, for example, if angles are relative to the vertical and increase from outermost port to outermost starboard, then port-side angle values must be multiplied by -1.

6.6. Angle values in GSF files

The beam angle values in GSF files are relative to the vertical and are expressed in degrees. Starboard-side data points (with positive acrosstrack values) have positive angles, port-side data points (with negative acrosstrack values) have negative angles.