EDGETECH SONAR DATA FILE FORMAT

DESCRIPTION OF THE EDGETECH (.jsf) FILE FORMAT

Document No. 990-0000048-1000

Revision: 1.13 / September 2011

Email: sales@edgetech.com
Web: http://www.edgetech.com

4 Little Brook Road
West Wareham, MA 02576
Tel: (508) 291-0057
Fax: (508) 291-2491

1141 Holland Drive, Suite 1
Boca Raton, FL 33487
Tel: (561) 995-7767
Fax: (561) 995-7761

TABLE OF CONTENTS

INTRODUCTION

EdgeTech topsides by default record in JSF file format, which consists of a set of messages. Each message begins with a 16-byte header which indicates the type of data to follow and its size. Different types of data will have different message numbers (Message Type field). Sonar data is recorded on a per-channel basis. Therefore, for a single frequency side-scan system there will be two messages per ping – one for port (channel 0) and one for starboard (channel 1). Other types of sensors (such as pitch roll) will have their own message numbers as well, and will similarly have a single message per reading set. A typical file might have the following:

Header1: Sonar Data for Subsystem SSL and channel 0 (Port Side)
Sonar Data Header for Message 1 (Ping number 1, Port, Time Stamp)
Sonar Data for Message 1 (16-bit integers – one per sample)

Header2: Sonar Data for Subsystem SSL and channel 1 (Starboard Side)
Sonar Data Header for Message 2 (Ping number 1, Starboard, Time Stamp)
Sonar Data for Message 2 (16-bit integers – one per sample)

Header3: Pitch Roll reading
Serial Device Standard Header (Time Stamp)
Pitch Roll Data Structure

Header4: Sonar Data for Subsystem SB
Sonar Data Header for Message 4 (Ping number 1, Time Stamp)
Sonar Data for Message 4 (16-bit integer pairs – one pair per sample)

Since data is stored in a binary format, the byte ordering of 16-bit and 32-bit values is important. JSF uses little endian (Intel) format for binary data where the least significant bytes are stored first. This is the native format for Intel x86 computers such as the IBM PC and compatibles. If data is read on a big endian machine (such as most Sun Workstations), you will need to byte reverse the data so that the 2 bytes of a 16-bit value are flipped, and the 4 bytes of a 32-bit value are flipped (Bytes 0, 1, 2, 3 become Bytes 3, 2, 1, 0).

EdgeTech topsides often have an option to limit the size of created files, and will generate a sequence of files for a long data run. In this case, the concatenation of these files will yield a valid JSF file for a longer run time if desired.

This document describes the messages of common interest to those processing JSF files. It is not intended to be a complete description of all valid messages.

C/C++ Code Example for Reading a JSF File

Here is a C code example for reading an entire JSF file.

```
void readFile(char *fileName)
{
  FILE *fid;
  int i;
  SonarMessageHeaderType hdr; /* Basic 16-byte message header */

  fid = fopen(fileName, “rb”);
  if (fid == NULL) return;

  while(!feof(fid))
  {
    if (fread(&hdr, sizeof(hdr), 1, fid) != 1)
      break;
    if (hdr.startOfMessage != SONAR_MESSAGE_HEADER_START)
    {
      printf(“Invalid file format\n”);
      break;
    }
    for(i = 0; i < hdr.byteCount; i++)
    {
      if (getc(fid) == EOF)
      {
        printf(“Invalid file format\n”);
        break;
      }
    }
    printf(“Message Type %d\n”, hdr.sonarMessage);
  }
  fclose(fid);
}
```

Details of data structure of the types of messages in a JSF file are described below.

16-Byte Message Header

A JSF file is a collection of messages, and every message in a JSF file begins with a sixteen byte long header. It identifies the type and size of the message, as well as the originating subsystem and channel.

Byte Offsets	Description	Size
0 – 1	Marker for the Start of Header (always 0x1601)	UINT16
2	Version number of protocol used (e.g.10)	UINT8
3	Session Identifier	UINT8
4 – 5	Message Type (e.g. 80 = Sonar Trace Data)	UINT16
6	Command Type 2 = Normal data source	UINT8
7	Subsystem Number 0 = Sub-bottom 20 = Single or Lower Frequency Side Scan 21 = Higher Frequency Side Scan	UINT8
8	Channel for a Multi-Channel Subsystem For Side Scan Subsystems, 0 = Port, 1 = Starboard For Serial Ports: Port #	UINT8
9	Sequence Number	UINT8
10 – 11	Reserved	UINT16
12 – 15	Size of following Message in Bytes	UINT32

Every message header begins (bytes 0 - 1) with a 0x1601 start of a message header marker. This serves as a sanity check during file processing.

The Protocol level (byte 2) indicates which revision of this specification was used to write that message. Messages of differing protocol levels may be interspersed in the same file. Maintaining backward compatibility in the public interface is a priority. Protocol level changes involve additional messages or changes to the non-public portion of the interface.

The session identifier (byte 3) is used for internal routing and can be ignored.

The Message Type field (bytes 4 - 5) defines the type of data to follow. Some data formats of interest are detailed in the following sections. If the Message Type field contains an unwanted or unknown (i.e. not defined below) type, use the Size of the message (bytes 12 – 15) to skip over the data to the next Message Header.

The message protocol is used for command and control as well as data. The command type field (byte 6) while of interest during real time operation can normally be ignored when reading JSF files.

The subsystem number (byte 7) is used to determine the source of the data. Common subsystem assignments are as follows:

Sub-bottom data - 0

Single frequency sidescan data - 20

Lower frequency data of a dual frequency sidescan - 20

Higher frequency data of a dual frequency sidescan - 21

Raw serial data - 100

Parsed serial data - 101

The channel (byte 8) is zero for single channel (most sub-bottom) systems. For most sidescan systems, it is zero for port and one for starboard. For serial port data the channel number is the logical port number, which often differs from the physical COM port in use.

The sequence number (byte 9) and reserved fields (bytes 10 - 11) can normally be ignored when reading JSF files.

The byte count (bytes 12 – 15) is the number of bytes until the next start of message header. This is the amount of additional data to read if processing the current message, or the amount of data to skip over if the current message is not of interest.

Message Type 80: Sonar Data Message (jsfdefs.h)

The Sonar Data Message consists of a single ping (receiver sounding period) of data for a single channel (such as Port Side Low Frequency Side-Scan). Standard sidescan sub-systems have two channels of data, port and starboard. Standard sub-bottom sub-systems have a single channel of data. Data files with higher channel counts exist. Which fields have data present depends on the system used and data acquisition procedures. This message may contain data from multiple non-acoustic sensors. Non-acoustic data contained in this message will often not be time interpolated. EdgeTech strongly recommends that if high positional or situational accuracy is required that the individual sensor messages be processed. Otherwise, this may be the only message that will need to be interpreted in a JSF file. Validity flags indicate which auxiliary fields are populated. By convention, if a value is not present the field is set to 0.

A Sonar Data Message consists of a 240-byte header, which is very similar to a traditional SEGY header, and the actual acoustic sample data follows the header. This 240-byte header described below:

Byte Offsets	Description	Size
0 – 3	Ping Time in seconds [since the start of time based on time() function] (1/1/1970) (added in protocol version 8)[1]	INT32
4 – 7	Starting Depth (window offset) in samples - usually zero	UINT32
8 – 11	Ping Number (increments with each ping)	UINT32
12 – 15	Reserved – Do not use	INT16 x 2
16 – 17	MSBs – Most Significant Bits – High order bits to extend 16 bit unsigned short values to 20 bits. The 4 MSB bits become the most significant portion of the new 20 bit value. Bits 0 - 3 – start frequency Bits 4 - 7 – end frequency Bits 8 – 11 – samples in this packet Bits 12 – 15 – reserved (added in protocol version 10)(see description below)	UINT16
18 – 27	Reserved – Do not use	INT16 x 5
28 – 29	ID Code (always 1) 1 = Seismic Data	INT16
30 – 31	Validity Flag (See mapping below)	UINT16
32 – 33	Reserved – Do not use	UINT16
34 – 35	Data Format 0 = 1 short per sample - Envelope Data 1 = 2 shorts per sample - Analytic Signal Data, (Real, Imaginary)	INT16
36 – 37	Distance from Antenna to Tow point in Centimeters, Aft + (Fish Aft = +)	INT16
38 – 39	Distance from Antenna to Tow Point in Centimeters, Starboard + (Fish to Starboard = +)	INT16
40 – 43	Reserved – Do not use	INT16 x 2

The Most Significant Bits fields are used to extend sixteen bit integers to twenty bits. These are added as needed when the range of possible values exceeds what can be stored in a sixteen bit integer. The simplest way to use these additional bits is to treat the value as a 32 bit integer, the existing value becomes the least significant 16 bits, the MSB field becomes the next most significant 4 bits with the most significant 12 bits set to zeros.

[1] NOTE: For protocol revisions 7 and before this field was always zero.

Validity flags bitmap:
- Bit 0: Lat Lon or XY valid
- Bit 1: Course valid
- Bit 2: Speed valid
- Bit 3: Heading valid
- Bit 4: Pressure valid
- Bit 5: Pitch roll valid
- Bit 6: Altitude valid
- Bit 7: Reserved
- Bit 8: Water temperature valid
- Bit 9: Depth valid
- Bit 10: Annotation valid
- Bit 11: Cable counter valid
- Bit 12: KP valid
- Bit 13: Position interpolated

Navigation Data

This is most often the position at the last navigation fix. See the time NMEA Navigation Data fields for the time of the fix. On most systems, position is not time interpolated and should not be used for mosaicing or other processing requiring high accuracy positioning. Validity flags indicate whether these fields are valid and interpolated. The representation of the navigation data depends on the coordinate-units field. For Latitude / Longitude representations, a positive value designates east of the Greenwich Meridian or north of the equator.

Byte Offsets	Description	Size
44 – 47	Kilometer of pipe (see bytes 30-31)	FLOAT32
48 – 79	Reserved – Do not use	INT16 x 16
80 – 83	X in millimeters or decimeters or Longitude in Minutes of Arc / 10000 (see bytes 30-31 and 88-89)	INT32
84 – 87	Y in millimeters or decimeters or Latitude in 0.0001 Minutes of Arc (see bytes 30-31 and 88-89)	INT32
88 – 89	Coordinate Units 1 = X, Y in millimeters 2 = Longitude, Latitude in minutes of arc times 10^{-4} 3 = X, Y in decimeters	INT16

Pulse Information

This data describes the transmitted pulse characteristics, as well as sampling parameters.

Byte Offsets	Description	Size
90 – 113	Annotation String (ASCII Data)	UINT8 x 24
114 – 115	Number of data samples in this packet See bytes 16 – 17 for MSB information Note: Very large sample sizes require multiple packets.	UINT16
116 – 119	Sampling Interval in Nanoseconds	UINT32
120 – 121	Gain Factor of ADC	UINT16
122 – 123	User Transmit Level Setting (0 – 100) percent	INT16
124 – 125	Reserved – Do not use	INT16
126 – 127	Transmit pulse starting frequency in decahertz (daHz) (units of 10Hz) See bytes 17 – 18 for MSB information	UINT16
128 – 129	Transmit pulse ending frequency in decahertz (daHz)(units of 10Hz) See bytes 16 – 17 for MSB information	UINT16
130 – 131	Sweep Length in milliseconds	UINT16
132 – 135	Pressure in milliPSI (1 unit = 1/1000 PSI) (see bytes 30-31)	INT32
136 - 139	Depth in millimeters (if not = 0) (see bytes 30-31)	INT32
140 – 141	Sample Freq of the data in hertz, modulo 65536 NOTE *	UINT16
142 – 143	Outgoing pulse identifier	UINT16
144 – 147	Altitude in millimeters (If bottom tracking valid) 0 implies not filled (see bytes 30-31)	INT32
148 - 155	Reserved – Do not use	INT32 x 2

NOTE* : For all data types EXCEPT RAW (Data Format = 2) this is the Sampling Frequency of the data. For RAW data, this is one-half the Sample Frequency of the data (Fs/2). All values are modulo 65536. Use this in conjunction with the Sample interval (Bytes 114-115) to calculate correct sample rate.

CPU Time

The time that the Acoustic data was recorded. The time of the start of the ping of data represented by the following trace data is the Ping Time.

The Ping Time in seconds since 1/1/1970, (in the same time base as the device messages that follow), is contained in bytes 0-3. The millisecondsToday field (bytes 200-203), MODULO 1000, will yield the milliseconds in current second, compatible with the time stamp of the device messages. This time stamp is only valid for data recorded in Protocol Revision 8 and above.

The Ping Time can also be determined from the Year, Day, Hour, Minute and Seconds as per bytes 156 to 165. Thus provides 1 second level accuracy and resolution. For higher resolution (milliseconds) use the Year, and Day values of bytes 156 to 159, and then use the milliSecondsToday value of bytes 200-203 to complete the timestamp. System time is set to UTC, regardless of time zone. This time format is backwards compatible with all older Protocol Revisions.

These 2 time stamps are equivalent and identical.

Byte Offsets	Description	Size
156 – 157	Year (e.g. 2009) (see Bytes 0-3) (Should not be used)	INT16
158 – 159	Day (1 – 366) (Should not be used)	INT16
160 – 161	Hour (see Bytes 200-203) (Should not be used)	INT16
162 – 163	Minute (Should not be used)	INT16
164 – 165	Second (should not be used)	INT16
166 – 167	Time Basis (always 3)	INT16

Weighting Factor

The trace data is transmitted as sixteen bit integers in block floating point format per message. This saves bandwidth and storage space while preserving dynamic range. The weighting factor MUST BE applied to each of the sixteen bit integer values to restore the original floating point value.

Byte Offsets	Description	Size
168 – 169	Weighting Factor N (Signed Value!) Defined as 2^{-N}	INT16
170 – 171	Number of pulses in the water	INT16

Orientation Sensor Data

These fields contain useful information about the attitude of the sonar sensor. The Compass heading will be magnetic heading of the towfish. If a Gyro sensor is properly interfaced to the Discover Topside acquisition unit, with a valid NMEA HDT message, this field will contain the Gyro heading, relative to true north

Byte Offsets	Description	Size
172 – 173	Compass Heading (0 to 360) in units of 1/100 degree (see bytes 30-31)	UINT16
174 – 175	Pitch: Scale by 180 / 32768 to get degrees, + = bow up (see bytes 30-31)	INT16
176 – 177	Roll: Scale by 180 / 32768 to get degrees, + = port up (see bytes 30-31)	INT16
178 – 179	Tow fish electronics Temperature, in unit of 1/10th degree C	INT16

Miscellaneous Data

Byte Offsets	Description	Size
180 – 181	Reserved – Do not use	INT16
182 – 183	Trigger Source 0 = Internal 1 = External 2 = Coupled	INT16
184 – 185	Mark Number 0 = No Mark	UINT16

NMEA Navigation Data

These fields contain the time of the last position fix. If the position data is interpolated this will be the same as the CPU and ping time.

Byte Offsets	Description	Size
186 – 187	Hour (0 – 23)	INT16
188 – 189	Minutes (0 – 59)	INT16
190 – 191	Seconds (0 – 59)	INT16
192 – 193	Course	INT16
194 – 195	Speed	INT16
196 – 197	Day (1 – 366)	INT16
198 – 199	Year	INT16

Other Miscellaneous Data

Byte Offsets	Description	Size
200 – 203	Milliseconds today (since midnight) (use in conjunction with Year / Day to get time of Ping)	UINT32
204 – 205	Maximum Absolute Value of ADC samples in this packet	UINT16
206 – 207	Reserved – Do not use	INT16
208 – 209	Reserved – Do not use	INT16
210 – 215	Sonar Software Version Number - ASCII	INT8 x 6
216 – 219	Initial Spherical Correction Factor (Useful for multi-ping / deep application) * 100	INT32
220 – 221	Packet Number Each ping starts with packet 1	UINT16
222 – 223	100 times the A/D Decimation Factor. Data is normally sampled at a high Rate. Digital filters are applied to precisely limit the signal bandwidth.	INT16
224 – 225	Decimation Factor after the FFT	INT16
226 – 227	Water Temperature in units of 1/10 degree C (see bytes 30-31)	INT16
228 – 231	Layback in meters	FLOAT32
232 – 235	Reserved – Do not use	INT32
236 – 237	Cable Out in decimeters (see bytes 30-31)	UINT16
238 – 239	Reserved – Do not use	UINT16

Sonar Trace Data

Sonar trace data follows the 240-byte header and consists of sixteen bit integer values. The number of integers to be read can be found by multiplying the number of samples in the trace (bytes 114-115) by the number of integers per sample for the data type used (1 or 2). Further doubling will yield the byte size of the data section. This should exactly match the preceding Message Header byte count, (bytes 12 –15) less the header size of 240.

Each of the data sample values then needs to be scaled by the weighting factor thus:

ScaledDatasample = datasample * 2^(-N). (NOTE Sign !)

Future expansions of this data format will use floating point values to represent samples, and will result in other valid values for Data Format (bytes 34-35). Data readers will be more robust if this data section is skipped over if the Data Type does not match the 4 values presented here.

Message Type 82: Side Scan Data Message (sidescandefs.h)

Side-Scan Data Messages are no longer used, and are only described here for historical reasons. While configuring a sonar to generate these messages is still possible, new systems are not configured in that manner. If your sonar is storing Side-Scan Data Messages the configuration should be changed to store Sonar Data Messages instead. Side-Scan Data Messages are never stored by Discover, and are only encountered in data stored by sonar. This data is almost always compressed rendering it unusable without further processing. A Side-Scan Data Message is similar to a Sonar Data Message. It contains the exactly the same acoustic data. While the Side Scan Data Message was intended for Side Scan data, it can also be used for Sub-bottom data. The system configuration determines which type of data is actually stored. Each Side Scan Data Message has an 80 byte header, the content of which is defined below. As with Sonar Data Messages, unused fields should be set to 0.

Byte Offsets	Description	Size
0 – 1	Subsystem (0 .. n)	UINT16
2 – 3	Channel Number (0 .. n)	UINT16
4 – 7	Ping number (increments with each ping period)	UINT32
8 – 9	Packet number (1..n) Each ping starts with packet 1	UINT16
10 – 11	TriggerSource (0 = internal, 1 = external)	UINT16
12 – 15	Samples in this packet	UINT32
16 – 19	Sample interval in ns of stored data	UINT32
20 – 23	Starting Depth (window offset) in samples	UINT32
24 – 25	Weighting Factor : Defined as 2^{-N} volts	INT16
26 – 27	Gain factor of ADC	UINT16
28 – 29	Maximum absolute value for ADC samples for this packet	UINT16
30 – 31	Range Setting (in decameters) (meters times 10)	UINT16
32 – 33	Unique pulse identifier	UINT16
34 – 35	Mark Number (0 = no mark)	UINT16
36 – 37	Data format 0 = 1 short per sample - envelope data the total number of bytes of data to follow is 2 * samples 1 = 2 shorts per sample - stored as real(1), imag(1), the total number of bytes of data to follow is 4 * samples	UINT16
38	Number of simultaneous pulses in the water	UINT8
39	Reserved – Do not use	UINT8

Computer date / time data acquired

Byte Offsets	Description	Size
40 – 43	Milliseconds today	UINT32
44 – 45	Year	INT16
46 – 47	Day of year (1 – 366)	UINT16
48 – 49	Hour of day (0 – 23)	UINT16
50 – 51	Minute (0 – 59)	UINT16
52 – 53	Second (0 – 59)	UINT16

Auxiliary sensor information

Byte Offsets	Description	Size
54 – 55	Compass heading in minutes (0 – 360) x 60	UINT16
56 – 57	Pitch Scale by 180 / 32768 to get degrees, + = bow up	INT16
58 – 59	Roll Scale by 180 / 32768 to get degrees, + = port up	INT16
60 – 61	Heave (centimeters)	INT16
62 – 63	Yaw (minutes)	INT16
64 – 67	Pressure in units of 1/1000 PSI	UINT32
68 – 69	Temperature in units of 1/10 of a degree Celsius	INT16
70 – 71	Water Temperature in units of 1/10 of a degree Celsius	INT16
72 – 75	Altitude in millimeters (or -1 if no valid reading)	INT32
76 – 79	Reserved – Do not use	UINT8 x 4

Sonar Trace Data

Sonar trace data follows the 80-byte header and consists of sixteen bit integer values. The number of integers to be read can be found by multiplying the number of samples in the trace (bytes 12-15) by the number of integers per sample for the data type used (1 or 2). Further doubling will yield the byte size of the data section. This should exactly match the preceding 16 byte Message Header byte count, (bytes 12 –15) less the header size of 80. Each of the data sample values then needs to be scaled by the weighting factor thus:

ScaledDatasample = datasample * 2^(- N).

Message Type 2020: Pitch Roll Data

A pitch roll message consists of a single reading from a pitch roll sensor such as a Seatex MRU, TSS or Octans device. Not all devices provide all data for the defined structure. Use the validity flags to determine which fields are populated.

Byte Offsets	Description	Size
0 – 3	Time in seconds (since the start of time based on time() function) (1/1/1970)	INT32
4 – 7	Milliseconds in the current second	INT32
8 – 11	Reserved – Do not use	UINT8 x 4
12 – 13	Acceleration in x: Multiply by (20 * 1.5) / (32768) to get Gs	INT16
14 – 15	Acceleration in y: Multiply by (20 * 1.5) / (32768) to get Gs	INT16
16 – 17	Acceleration in z: Multiply by (20 * 1.5) / (32768) to get Gs	INT16
18 – 19	Rate Gyro in x: Multiply by (500 * 1.5) / (32768) to get Degrees/Sec	INT16
20 – 21	Rate Gyro in y: Multiply by (500 * 1.5) / (32768) to get Degrees/Sec	INT16
22 – 23	Rate Gyro in y: Multiply by (500 * 1.5) / (32768) to get Degrees/Sec	INT16
24 – 25	Pitch Multiply by (180.0 / 32768.0) to get Degrees Bow up is positive	INT16
26 – 27	Roll: Multiply by (180.0 / 32768.0) to get Degrees Port up is positive	INT16
28 – 29	Temperature in units of 1/10 of a degree Celsius	INT16
30 – 31	Device specific info. This is device specific info provided for Diagnostic purposes	UINT16
32 – 33	Estimated Heave in millimeters	INT16
34 – 35	Heading in units of 0.01 Degrees (0…360)	UINT16
36 – 39	Data valid flags Bit 0: ax Bit 1: ay Bit 2: az Bit 3: rx Bit 4: ry Bit 5: rz Bit 6: pitch Bit 7: roll Bit 8: heave Bit 9: heading Bit 10: temperature Bit 11: devInfo	INT32
40 – 43	Reserved – Do not use	INT32

Message Type 2002: NMEA String

A NMEA String consists of a time stamp followed by a NMEA string as read from a GPS, Gyro or other device. Each message is a single NMEA string excluding the <CR>/<LF>.

Byte Offsets	Description	Size
0 – 3	Time in seconds since 1970	INT32
4 – 7	Milliseconds in the current second	INT32
8	Source, 1 = Sonar, 2 = Discover, 3 = ETSI	INT8
9 – 11	Reserved – Do not use	UINT8 x 3
12 – To Message Length	NMEA string data	INT8 x remaining length

Message Type 2060: Pressure Sensor Reading

If a pressure sensor is present in the system these messages will be in the data stream.

A single pressure sensor reading is provided, along with a time-stamp. While pressure sensors may be configured in different units, the default is PSI absolute.

Byte Offsets	Description	Size
0 – 3	Time in seconds (since the start of time based on time() function).	INT32
4 – 7	Milliseconds in the current second.	INT32
8 – 11	Reserved – Do not use	UNIT8 x 4
12 – 15	Pressure in units of 1/1000th of a PSI	INT32
16 – 19	Temperature in units of 1/1000th of degree Celsius.	INT32
20 – 23	Salinity in Parts Per Million	INT32
24 – 27	Data valid flags: Bit 0: pressure Bit 1: temp Bit 2: salt PPM Bit 3: conductivity Bit 4: sound velocity	INT32
28 – 31	Conductivity in micro-Siemens per cm	INT32
32 – 35	Velocity of Sound in mm per second	INT32
36 – 75	Reserved – Do not use	INT 32 x 10

Message Type 2080: Doppler Velocity Log Data (DVL)

This is data from a DVL (if fitted) and often includes velocity and altitude readings.

Byte Offsets	Description	Size
0 – 3	Time in seconds (since the start of time based on time() function).	INT32
4 – 7	Milliseconds in the current second.	INT32
8 – 11	Reserved – Do not use	UINT8 x 4
12 – 15	Flags. Indicates which values are present. Bit 0: X, Y Velocity present Bit 1: 1 => Velocity in ship coordinates 0 => Earth coordinates Bit 2: Z (Vertical Velocity) present Bit 3: X, Y Water Velocity present Bit 4: Z (Vertical Water Velocity) present Bit 5: Distance to bottom present Bit 6: Heading present Bit 7: Pitch present Bit 8: Roll present Bit 9: Temperature present Bit 10: Depth present Bit 11: Salinity present Bit 12: Sound velocity present ------- Bit 31: Error detected	UINT32
16 – 31	4 Integers: Distance to bottom in cm for up to 4 beams.	INT32 x 4
	A 0 value indicates an invalid or non-existing reading.	
32 – 33	X Velocity with respect to the bottom in mm / second Positive => Starboard or East. -32768 indicates an invalid reading.	INT16
34 – 35	Y Velocity: Positive => Forward or North (mm/second)	INT16
36 – 37	Z Vertical Velocity: Positive => Upward (mm/second)	INT16
38 – 39	X Velocity with respect to a water layer in mm / second Positive => Starboard or East	INT16 x 3
40 - 41	Y Velocity: Positive => Forward or North	
42 - 43	Z Vertical Velocity: Positive => Upward	
44 - 45	Depth from depth sensor in decimeters	UINT16
46 - 47	Pitch -180 to +180 degree (units = 0.01 of a degree) + Bow up	INT16
48 - 49	Roll -180 to +180 degrees (units = 0.01 of a degree) + Port up	INT16
50 - 51	Heading: 0 to 360 degrees (in units of 0.01 of a degree)	UINT16
52 - 53	Salinity in 1 part per thousand	UINT16
54 - 55	Temperature in units of 1/100 of a degree Celsius	INT16
56 - 57	Sound velocity in meters per second	INT16
58 - 71	Reserved – Do not use	INT16 x 7

Message Type 2090: Situation Message

A situation message is a composite of several motion / position sensors. This message is not commonly used. The detailed data structure is shown below:

Byte Offsets	Description	Size
0 - 3	Time in seconds (since the start of time based on time() function).	INT32
4 - 7	Milliseconds in the current second.	INT32
8 - 11	Reserved – Do not use	INT8 x 4
12 - 15	Validity Flags Validity Flags indicate which of the following fields are valid. If the corresponding bit is set the field is valid. Bit 0 : microsecondTimestamp Bit 1 : latitude Bit 2 : longitude Bit 3 : depth Bit 4 : heading Bit 5 : pitch Bit 6 : roll Bit 7 : XRelativePosition Bit 8 : YRelativePosition Bit 9 : ZRelativePosition Bit 10 : XVelocity Bit 11 : YVelocity Bit 12 : ZVelocity Bit 13 : NorthVelocity Bit 14 : EastVelocity Bit 15 : downVelocity Bit 16 : XAngularRate Bit 17 : YAngularRate Bit 18 : ZAngularRate Bit 19 : XAcceleration Bit 20 : YAcceleration Bit 21 : ZAcceleration Bit 22 : latitudeStandardDeviation Bit 23 : longitudeStandardDeviation Bit 24 : depthStandardDeviation Bit 25 : headingStandardDeviation Bit 26 : pitchStandardDeviation Bit 27 : rollStandardDeviation	UINT32
16 - 19	Reserved – Do not use	UINT x 4
20 - 27	Microsecond timestamp, us since 12:00:00 am GMT, January 1, 1970	UINT64
28 - 35	Double float: Latitude in degrees, north is positive	FLOAT64
36 - 43	Double float: Longitude in degrees, east is positive	FLOAT64
44 - 51	Double float: Depth in meters	FLOAT64
52 - 59	Double float: Heading in degrees	FLOAT64
60 - 67	Double float: Pitch in degrees, bow up is positive	FLOAT64
68 – 75	Double float: Roll in degrees, port up is positive	FLOAT64
76 - 83	Double float: X, forward, relative position in meters, surge	FLOAT64
84 - 91	Double float: Y, starboard, relative position in meters, sway	FLOAT64

92 - 99	Double float: Z, downward, relative position in meters, heave	FLOAT64
100 - 107	Double float: X, forward, velocity in meters per second	FLOAT64
108 - 115	Double float: Y, starboard, velocity in meters per second	FLOAT64
116 - 123	Double float: Z, downward, velocity in meters per second	FLOAT64
124 - 131	Double float: North velocity in meters per second	FLOAT64
132 - 139	Double float: East velocity in meters per second	FLOAT64
140 - 147	Double float: Down velocity in meters per second	FLOAT64
148 - 155	Double float: X angular rate in degrees per second, port up is positive	FLOAT64
156 - 163	Double float: Y angular rate in degrees per second, bow up is positive	FLOAT64
164 - 171	Double float: Z angular rate in degrees per second, starboard is positive	FLOAT64
172 - 179	Double float: X, forward, acceleration in meters per second per second	FLOAT64
180 - 187	Double float: Y, starboard, acceleration in meters per second per second	FLOAT64
188 - 195	Double float: Z, downward, acceleration in meters per second per second	FLOAT64
196 - 203	Double float: Latitude standard deviation in meters	FLOAT64
204 - 211	Double float: Longitude standard deviation in meters	FLOAT64
212 - 219	Double float: Depth standard deviation in meters	FLOAT64
220 - 227	Double float: Heading standard deviation in degrees	FLOAT64
228 - 235	Double float: Pitch standard deviation in degrees	FLOAT64
236 - 243	Double float: Roll standard deviation in degrees	FLOAT64
244 - 275	Reserved – Do not use	UINT16 x 16

Message Type 426: File Timestamp Message

File timestamp messages are often found at the beginning and end of a file. The contain timestamps in the following format:

Byte Offsets	Description	Size
0 – 3	Time in seconds (since the start of time based on time() function).	INT32
4 – 7	Milliseconds in the current second.	INT32

Message Type 428: File Padding Message

A file padding message is sometimes found at the end of the file. In some implementations files are padded to end of the sector to optimize the write process. These messages should be ignored.

Message Type 182: System Information Message

The system information message contains details of the system used to acquire data. This message is normally present at the beginning of a JSF file, and may be repeated if configuration parameters change.

Byte Offsets	Description	Size
0 - 3	System Type	INT32
4 – 7	Reserved – Do not use	
8 – 11	Version Number of Sonar Software used to generate data	INT32
12 – 19	Reserved – Do not use	
20 – 23	Serial Number of Tow Vehicle used to collect data	INT32
24 – End	Reserved – Do not use	

The size of the System Information Message is subject to change as more detailed information is may be added in future versions of the software. The byte count in the message header should be used to determine the total size of the structure and get to the next message in the file.

The System Type is the major type of system used to collect data. To date, the following system types are defined:

System Type Number	Description
1	2xxx Series, Combined Sub-Bottom / Side Scan with SIB Electronics
2	2xxx Series, Combined Sub-Bottom / Side Scan with FSIC Electronics
4	4300-MPX (Multi-Ping)

5	3200-XS, Sub-Bottom Profiler with AIC Electronics
6	4400-SAS, 12-Channel Side Scan
7	3200-XS, Sub Bottom Profiler with SIB Electronics
11	4200 Limited Multipulse Dual Frequency Side Scan
14	3100-P, Sub Bottom Profiler
16	2xxx Series, Dual Side Scan with SIB Electronics
17	4200 Multipulse Dual Frequency Side Scan
18	4700 Dynamic Focus
19	4200 Dual Frequency Side Scan
20	4200 Dual Frequency non Simultaneous Side Scan
21	2200-MP Combined Sub-Bottom / Dual Frequency Multipulse Side Scan
23	4600 Multipulse Bathymetric System
24	4200 Single Frequency Dynamically Focused Side Scan
25	4125 Dual Frequency Side Scan
27	4600 Monopulse Bathymetric System
128	4100, 272 /560A Side Scan

Message Type 2100: Cable Counter Data Message

Cable counter data messages:

Byte Offsets	Description	Size
0 – 3	Time in seconds (since the start of time based on time() function).	INT32
4 – 7	Milliseconds in the current second.	INT32
8 – 11	Reserved – Do not use	UNIT8 x 4
12 – 15	Cable Length in meters	FLOAT32
16 – 19	Cable Speed in meters / second	FLOAT32
20 – 21	Cable Length valid flag, 0 – Invalid,	INT16
22 – 23	Cable Speed valid flag, 0 – Invalid,	INT16
24 – 25	Cable Counter Error, 0 – No Error,	INT16
26 – 27	Cable Tension valid flag, 0 – Invalid.	INT16

28 – 31	Cable Tension in kilograms	FLOAT32

Message Type 2111: Container Timestamp Message

Some messages in a JSF file are generated by external entities then passed to the recording system in containers. These messages are only checked to see if their length matches that specified in the message header, no other validation is performed. These contained messages are always preceded by a container timestamp message. This message contains the receipt timestamp of the container message.

Byte Offsets	Description	Size
0 – 3	Time in seconds (since the start of time based on time() function).	INT32
4 – 7	Milliseconds in the current second.	INT32
8 – 11	Reserved – Do not use	UNIT8 x 4

Message Type 9001: Discover-2 General Prefix Message

The Discover 2 topside is designed to support multiple sonar or other data source providers. Instead of introducing a new messaging format, EdgeTech has opted to introduce messages generated by Discover2 in a backward compatible way. The General Prefix Message proceeds most messages (such as NMEA strings, pitch/roll, etc.) written by Discover 2 and provides the supplementary context information for the message that follows it.

Discover-2 General Prefix Data Structure:

Byte Offsets	Description	Size
0 – 7	Timestamp in the higher resolution Discover2 format. This is equivalent to the Microsoft Dot Net DateTime.Ticks property. The resolution is 10^{-7} of a second (0.1 microsecond per increment), and is referenced to 12:00 midnight, Jan 1, 0001 C.E. in the Gregorian Calendar.	INT64

8 – 11	Data Source Serial Number. A unique serial number, which could for example distinguish one tow fish from another, otherwise identically configured tow fish.	INT32
12 – 13	Message Version Number. This is the version number of this message. This differs from the protocol version number in the main message header.	INT16
14 – 15	Data Source Device. For each Serial Number, there may be multiple devices.	UINT16

Message Type 9002: Discover-2 Situation Data

A typical towed system often contains two Sensor Platforms - a boat doing the towing and a sidescan sonar towfish. This message is a summary of the "situation data" for a sensor platform. This data is written for every defined sensor platform, normally at a 5Hz rate. A sensor platform may contain multiple sensors that provide the same type of data (e.g. Lat/Lon from RMC and GGL). In this case, the configuration of the run time system contains a prioritized list of situation data sources, and it only reports the highest priority available source data.

Byte Offsets	Description	Size
0 – 15	Discover-2 General Prefix Data Structure (see Discover-2 General Prefix Message)	Structure
16 – end	Discover-2 Situation Data Object	Varies

Discover-2 Situation Data Object

==========

The Discover-2 Situation Data Object is of a variable size. It contains a fixed size header, and then a variable length list of situation item details.

Situation Master Header:

Byte Offsets	Description	Size

0 – 15	GUID of platform. This is a unique value used to represent the source sensor platform. When Discover2 is configured and sensor platforms are defined the GUID is generated.	128-bits
16 – 17	Sensor Platform Type: 0: Boat 1: Towed Sensor Platform (towfish)	UINT16
18 – 19	Platform enumerator	UINT16
20 – 23	Number of Situation IDs Present in the List	UINT32

The variable sized list contains information for each situation ID present. A situation like position needs to convey multiple values (for example, latitude and longitude), whereas others, like altitude, only need to convey a single value. For this reason, each item can be of variable size. For each ID, there is a fixed part which consists of the following:

Situation ID Fixed Header:

Byte Offsets	Description	Size
0 – 1	Situation ID Code: 0: Altitude - in Meters 1: Cable Angle - in Degrees 2: Cable Out - in Meters 3: Cable Tension - in TBD Units 4: Cable Velocity - in Meters / Second 5: Course - in Degrees (0 to 360) 6: Depth - in Meters 7: Heading - in Degrees (0 to 360) 8: Heave - in Meters 9: Ice Thickness - in Meters 10: KP - Kilometer Point 11: Layback - in Meters 12: Navigation Position - units Vary 13: Pitch - in Degrees (-180 to 180) 14: Pressure - in PSI Absolute 15: Roll - in Degrees (-180 to 180) 16: Salinity in PPM 17: Sediment Sound Speed 1 - Meters / Second 18: Sediment Sound Speed 2 - Meters / Second 19: Speed - Meters / Second 20: Water Sound Speed - Meters / Second 21: Water Temperature - in Degrees C Others may be added in the future.	UINT16
2 – 2	Size of entry in bytes. This allows the skipping of unsupported types. To go from one list item to the next, add the size in bytes to the reference point.	UINT8

3 – 3	Derivation Source: 0: Unknown - There is no data available. 2: Un-interpolated - Most recent value used. 3: Interpolated for the specified timestamp time	UINT8
4 – 5	Derivation Flags: Only present if the Derivation Source is 2 or 3. These are hints of how the value was derived. An example is if water temperature or salinity were not available in calculating depth from pressure (and therefore, the nominal value for these was used). The state flag bit definitions will vary with the situation ID, and are presently TBD.	UINT16
6 – 7	Data format code: Only present if the Derivation Source is 2 or 3. The data format codes are as follows: 1: Double - An 8 byte double precision float follows. 2: Float - A 4 byte single precision float follows. 3: Lat Lon - What follows are 2 values: Longitude: 8 byte double precision float - units Degrees Longitude: 8 byte double precision float - units Degrees 4: X Y - What follows are 2 values: X: 8 byte double precision float - units Meters Y: 8 byte double precision float - units Meters 5: UTM Position - What follows are 3 values: X: 8 byte double precision float - units Meters Y: 8 byte double precision float - units Meters Zone: 4 byte integer representing the UTM Zone 6: XYZ - What follows are 3 values: X: 8 byte double precision float - units Meters Y: 8 byte double precision float - units Meters Z: 8 byte double precision float - units Meters 7: Range and Bearing - What follows are 2 values: Range: 8 byte double precision float - Meters. Bearing: 8 byte double precision float - Degrees.	UINT16
Variable	The data as described in bytes 6-7.	Variable

This example may help clarify the data structure. Assume that there are 3 situation IDs configured: Position, Altitude, and Heading, with Heading not being received. In this case, the situation data object may contain the following:

Bytes	Value
	Main Header
0-15	This is a long binary bit pattern - the GUID for the platform.
16-19	3 (3 Situation IDs to follow)
	Altitude Information
20-21	0 (Altitude follows)
22	16 (Total size of this item in bytes)
23	3 (State is interpolated)
24-25	0 (State flags for Altitude)
26-27	1 (Double precision float follows)
28-35	The actual altitude reading as a double.
	Heading Information
36-37	7 (Heading Follows)
38	4 (Total size of this item in bytes)
39	0 (State is not available)
	Nav Position Information
40-41	12 (Nav Position Follows)
42	24 (Total size of this item in bytes)
43	3 (State is interpolated)
44-45	0 (State flags for Position)
46-47	3 (Lat Lon Follows)
48-55	Double for Longitude
56-63	Double for Latitude

Message Type 9003: Discover-2 Acoustic Prefix Message

This is a prefix message with supplementary data for the acoustic (type 80) message which follows it. This message consists of a fixed header containing supplementary acoustic information, followed by the previously defined situation data object containing the situation information for that channel.

Acoustic Prefix Message:

Byte Offsets	Description	Size
0 – 15	Discover-2 General Prefix Data Structure (see Discover-2 General Prefix Message).	Structure
16 - 19	Ping Number (for redundant validation of the following messages ping number).	UINT32
20 - 23	Mixer Frequency in kHz if the data is base banded.	FLOAT32
24 - 27	Mixer Phase at first sample of ping measured in Turns. Range is 0.0 to 1.0 where 0.0 indicates a phase of 0 and 0.5 indicates a phase of 180 degrees.	FLOAT32
28 - 31	Sample Rate in kHz.	FLOAT32
32 - 35	Sample Offset in samples to the first sample in the acoustic message.	UINT32
36 - 37	Processing Flags - Reserved	UINT16
38 - 39	Pulse Index of active pulse in configuration	UINT16
40	Original Data Source - Reserved	UINT8
41	Data source: 1: Acquire (Diagnostic output only) 2: Acquire Windowed (AKA raw data) 3: Match Filtered (Diagnostic output only) 4: Math Filtered Windowed (AKA match filtered data) 5: Post Match Filter Processed. Note: It is possible to get the same data at multiple processing stages in the file. The normal output would be type 4 or type 5 - where type 5 would be present if there is a post match filter processor (e.g. MPX or Dynamic Focus blending).	UINT8
42	MPX Pulse Number (0 to 3)	UINT8
43	Packet Number (Only applies to diagnostic multi-packed data)	UINT8
44 – 47	Reserved	UINT32
48 – end	Discover-2 Situation Data Object (see description under message type 9002).	Varies