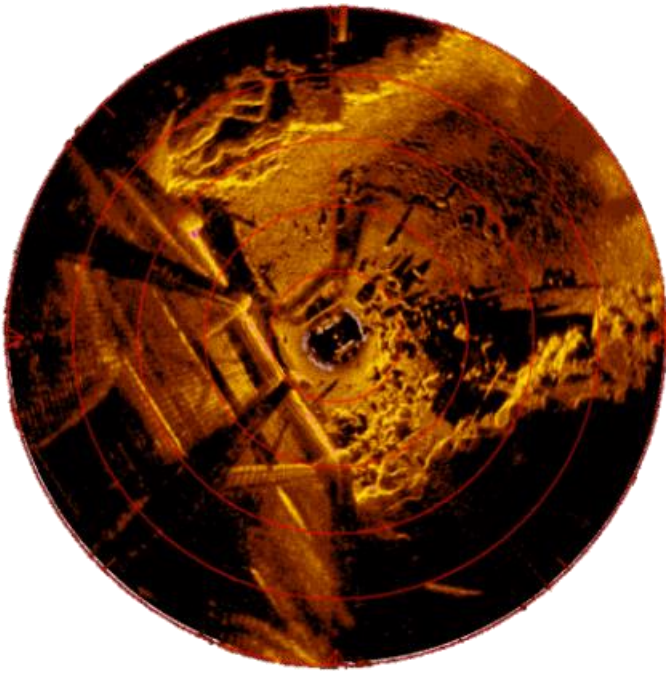
Operator manual

KONGSBERG

SMB File Format

Singlebeam Sonar



SMB File Format

Singlebeam Sonar

Revision

Issue	Date	Written by	Checked by	Approved by
2.1	31-Mar.-08	LL	AZ	AZ
2.2	18-Jan-12	AZ	MM	MM
2.3 Draft	07-May-15	AZ		

Document history

This document supersedes document number 900-00007904 and includes a description of the data associated with all singlebeam sonars. The information contained in this document is subject to change without prior notice. Kongsberg Mesotech Ltd. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this document.

Kongsberg Mesotech Ltd.

1598 Kebet Way
Port Coquitlam, BC
V3C 5M5 Canada

Telephone: +1 604 464 8144
Telefax: +1 604 941 5423
www.kongsberg-mesotech.com
Email: km.sales.vancouver@kongsberg.com

KONGSBERG

Table of Contents

1. OVERVIEW

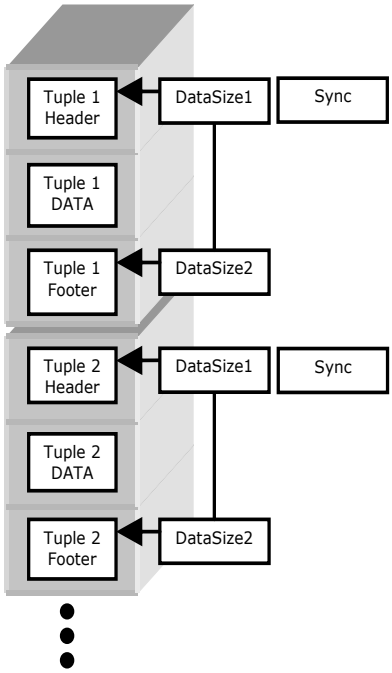
The SMB file format is sonar image file generated by the multiple beam sonar software of Kongsberg Mesotech Ltd. A SMB file is a binary file created by recording data and contains sonar image view and head settings information for playback. The extension of the first recorded file is.smb and the subsequent are .001, .002, and so on.

The entire SMB format is based on the concept of "Tuple". This document describes all the types of Tuples included in the singlebeam sonar SMB files and data type information within each Tuple.

1.1. The Tuple Concept

The following diagram shows the structure of the SMB file. A Tuple includes a Tuple header, a Tuple packet, and a Tuple footer . A SMB file is a group of different types of Tuples. A packet is a group of data between the header and footer.

The Tuple header and footer contain matching data size values to indicate how large the data portion is and what kind of Tuple it is. The header also contains a special 16-bit pattern of 0x8000, called Sync, which can be used in conjunction with DataSize1 and DataSize2 to find Tuples in a stream from any location..



1.2. Data Types

The data types used in this document are as follows:

Data Type	Description
byte	8-bit character
char	8-bit character.
int8	8-bit signed integer.
uint8	8-bit unsigned integer.
uint16	16-bit unsigned integer.
int16	16-bit signed integer.
short	16-bit signed integer
uint32	32-bit unsigned integer.
int32	32-bit signed integer.
float	32-bit floating point
double	64-bit floating point
bool	Boolean, 1 byte
string	Variable length ASCII strings

2. TUPLE HEADERS AND FOOTERS

2.1. BINARY_TUPLE_HEADER

This is the standard Tuple header used for most Tuples. It supports a data size up to 64KB.

Data Type	Member	Description
uint16	Sync	This is used in conjunction with DataSize1 and DataSize2 to find data packets. DataSize2 is a copy of DataSize1 but is located at the end of the Tuple within the BINARY_TUPLE_FOOTER
uint16	SourceType	Type of the Controller that is sending data
uint16	SourceID	ID of the Controller that is sending data
uint16	DataType	Type of data. One of the DATA_TYPE_XXX constants.
uint32	DataTime	Time at which data was collected in milliseconds since midnight GMT.
uint16	DataSize1	Size of data in bytes, excluding the header. This is used in conjunction with Sync and DataSize2 to find data packets.

2.2. BINARY_TUPLE_FOOTER

This is the standard Tuple footer used for most Tuples. It supports a data size up to 64KB.

Data Type	Member	Description
uint16	DataSize2	Size of data in bytes, excluding the header. This is a copy of DataSize1.
uint32	DataSize2	Size of data in bytes, excluding the header. This is a copy of DataSize1.

3. SENSOR TUPLE

This section describes the DATA_TYPE_XXX data types that pertain to NMEA sensor data.

3.1. DATA_TYPE_GGA = 0

Global Positioning System Fix Data, Time, Position and fix related data for a GPS receiver.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	Utc	UTC time of position. Number of milliseconds since midnight.
double	Latitude	The latitude value.
char	cNS	Latitude: 'N' = North 'S' = South
double	Longitude	The longitude value.
char	cEW	Longitude 'E' = East 'W' = West
uint16	QualityIndicator	GPS quality indicator 0 = Fix Not Available 1 = GPS Fix 2 = Differential GPS fix
uint16	NumSatellites	Number of Satellites in use
double	HorizontalDilution	Horizontal dilution of precision
double	AntennaAltitude	Antenna altitude above / below sea level (geoid), in meters
double	GeoidalSeparation	Geoidal separation
double	DataAge	Age of Differential GPS data
uint16	StationID	Differential reference station ID
BINARY_TUPLE_FOOTER (see page 3)		

3.2. DATA_TYPE_GLC = 1

Geographic Position, Loran-C

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
uint32	Gri	GRI, microseconds/10
double	MasterTOA	Master TOA, microseconds
char	cTOA	Signal status
double	fTD1	Time Difference 1 Microseconds
char	cTD1	Time Difference 1 Signal Status
double	fTD2	Time Difference 2 Microseconds
char	cTD2	Time Difference 2 Signal Status
double	fTD3	Time Difference 3 Microseconds
char	cTD3	Time Difference 3 Signal Status
double	fTD4	Time Difference 4 Microseconds
char	cTD4	Time Difference 4 Signal Status
double	fTD5	Time Difference 5 Microseconds
char	cTD5	Time Difference 5 Signal Status
BINARY_TUPLE_FOOTER (see page 3)		

3.3. DATA_TYPE_GLL = 2

Geographic Position — Latitude / Longitude

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	Latitude	The latitude value.
char	cNS	Latitude: 'N' = North 'S' = South
double	longitude	The longitude value.
char	cEW	Longitude 'E' = East 'W' = West
double	utc	UTC time of position. Number of milliseconds since midnight.
char	cStatus	Status: 'A' = Data valid 'V' = Data not valid
BINARY_TUPLE_FOOTER (see page 3)		

3.4. DATA_TYPE_GSV = 3

Satellites in View

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
byte	totalMessages	Total number of messages, 1 to 3
byte	messageNumber	Message number, 1 to 3
byte	totalSatel	Total number of satellites in view
uint16	satPRN	Satellite PRN number
byte	elevation	Elevation, degrees, 90 max
uint16	azimuth	Azimuth, degrees True, 000 to 359
byte	SNR	SNR (C/No) 00-99 dB
BINARY_TUPLE_FOOTER (see page 3)		

3.5. DATA_TYPE_HDG = 4

Heading — Deviation and Variation

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	heading	Magnetic sensor heading in degrees
double	deviation	Magnetic deviation in degrees
char	cDeviationEW	Magnetic deviation direction: 'E' = Easterly = 'W' = Westerly
double	variation	Magnetic variation
char	cVariationEW	Magnetic variation direction: 'E' = Easterly 'W' = Westerly
BINARY_TUPLE_FOOTER (see page 3)		

3.6. DATA_TYPE_HDT = 5

True heading.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	heading	Heading, degrees True
char	cTrue	'T' = True
BINARY_TUPLE_FOOTER (see page 3)		

3.7. DATA_TYPE_ MTW = 6

Water Temperature.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	waterTemp	Temperature, degrees Celsius
BINARY_TUPLE_FOOTER (see page 3)		

3.8. DATA_TYPE_TLL = 7

Target Latitude and Longitude.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
byte	number	Target number
double	latitude	The latitude value.
char	cNS	Latitude: 'N' = North 'S' = South
double	longitude	The longitude value.
char	cEW	Longitude 'E' = East 'W' = West
char[20]	userData	
double	utc	UTC of data. Number of milliseconds since midnight.
char	targetStatus	Target status
char	reference	
BINARY_TUPLE_FOOTER (see page 3)		

3.9. DATA_TYPE_VHW = 8

Water speed and heading.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	trueHeading	True Heading in degrees
double	magHeading	Magnetic Heading in degrees
double	speedN	Speed in knots
double	speedK	Speed in km/h
BINARY_TUPLE_FOOTER (see page 3)		

3.10. DATA_TYPE_VLW = 9

Distance traveled through water.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	totalDistance	Total cumulative distance, in nautical miles.
double	sinceReset	Distance since reset, in nautical miles.
BINARY_TUPLE_FOOTER (see page 3)		

3.11. DATA_TYPE_TSS = 10

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
uint32	horizAccel	Horizontal acceleration
uint32	vertAccel	Vertical acceleration
double	heave	Heave
char	qualFlag	Quality flag
double	roll	Roll
double	pitch	Pitch
BINARY_TUPLE_FOOTER (see page 3)		

3.12. DATA_TYPE_USER_DEFINED = 11

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
char[…] + '\0'	Name	Null-terminated variable-length string specifying the name of the sensor.
char[…] + '\0'	Format	Null-terminated variable-length string specifying the format of the sensor's data, as follows: 'A' = ASCII char 'B' = BCD char 'F' = Search for the next field delimiter 'I' = Ignore character 'L' = Field is an integer 'E' = Field is a float number 'H' = Field is a hex number 'N' = Print as ASCII to the next field delimiter 'D' = Insert decimal point 'S' = Insert space
byte[…]	Data	The rest of the data portion contains binary data from the sensor, according to the above Format member.
BINARY_TUPLE_FOOTER (see page 3)		

3.13. DATA_TYPE_HEAD_SENSORS = 25

This Tuple contains information from sensors in the head.

BINARY_TUPLE_HEADER (see page 3)

Data Type	Member	Description
double	DepthBelowTransducer	Depth below transducer, in meters.
double	Depth	Depth in meters.
double	Temperature	Temperature in degrees Celsius.
double	Pitch	Pitch in degrees.
double	Roll	Roll in degrees.
double	Compass	Compass value in degrees, or 0xFFFF to indicate invalid value.
double	CableVoltage1	Cable 1 voltage.
double	CableVoltage2	Cable 2 voltage.
byte	Catch1	Status of catch sensor 1: 1 = On 2 = Active 3 = Triggered
byte	Catch2	Status of catch sensor 2: 1 = On 2 = Active 3 = Triggered
byte	Catch3	Status of catch sensor 3: 1 = On 2 = Active 3 = Triggered
byte	Catch4	Status of catch sensor 4: 1 = On 2 = Active 3 = Triggered
uint16	SensorReadFlags	Flags indicating which sensor has a new read value
uint16	wCompassSque	Compass sequence number
uint16	wDepthSque	Depth sequence number
uint16	wPitchRollSque	Pitch/Roll sequence number
byte[56]	Reserved	

BINARY_TUPLE_FOOTER (see page 3)

3.14. DATA_TYPE_DBT = 1001

Depth below transducer.

BINARY_TUPLE_HEADER (see page 3)

Data Type	Member	Description
double	depthFeet	Depth in feet
char	cf	‘f’
double	depthM	Depth in meters
char	cM	‘M’
double	depthFathoms	Depth in fathoms
char	cF	‘F’

BINARY_TUPLE_FOOTER (see page 3)

3.15. DATA_TYPE_DPT = 1002

Depth below transducer.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	depth	Water depth relative to the transducer, meters
double	offset	Offset from transducer, meters
BINARY_TUPLE_FOOTER (see page 3)		

3.16. DATA_TYPE_HDM = 1003

Magnetic heading.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	headingMag	Degrees magnetic
char	cM	'M'
BINARY_TUPLE_FOOTER (see page 3)		

3.17. DATA_TYPE_RMA = 1004

Recommended minimum navigation information.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
char	cStatus	Status: 'A' = Data valid 'V' = Blink, Cycle or SNR warning
double	latitude	The latitude value.
char	cNS	Latitude: 'N' = North 'S' = South
double	longitude	The longitude value.
char	cEW	Longitude 'E' = East 'W' = West
double	timeDifA	Time difference A, in microseconds.
double	timeDifB	Time difference B, in microseconds
double	speedOG	Speed over ground, in knots.
double	courseOG	Course over ground, in degrees True.
double	magVar	Magnetic variation, in degrees E/W.
char	cMagVarEW	Magnetic variation direction: 'E' = East 'W' = West
BINARY_TUPLE_FOOTER (see page 3)		

3.18. DATA_TYPE_RMC = 1005

Recommended minimum navigation information.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	utc	UTC of position fix. Number of milliseconds since midnight.
char	cStatus	Status: 'A' = Data valid 'V' = Navigation receiver warning
double	latitude	The latitude value.
char	cNS	Latitude: 'N' = North 'S' = South
double	longitude	The longitude value.
char	cEW	Longitude 'E' = East 'W' = West
double	speedOG	Speed over ground, in knots
double	courseOG	Course over ground, in degrees True
uint32	dwDate	Date: dd|mm|yy
double	magVar	Magnetic variation, in degrees E/W
char	cMagVarEW	Magnetic variation direction: 'E' = East 'W' = West
BINARY_TUPLE_FOOTER (see page 3)		

3.19. DATA_TYPE_VBW = 1006

Dual Ground / Water Speed.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	longWaterSpeed	Longitudinal water speed, in knots. Negative means astern.
double	transWaterSpeed	Transverse water speed, in knots. Negative means port.
char	cWSStatus	Water speed status: 'A' = Data valid
double	longGroundSpeed	Longitudinal ground speed, in knots. Negative means astern.
double	transGroundSpeed	Transverse ground speed, knots. Negative means port.
char	cGSStatus	Ground speed status: 'A' = Data valid 'V' = Data invalid
BINARY_TUPLE_FOOTER (see page 3)		

3.20. DATA_TYPE_VTG = 1007

Track made good and ground speed.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	courseOGTrue	Course over ground, in degrees True.
char	cTrue	'T' = True
double	courseOGMag	Course over ground, in degrees Magnetic.
char	cMag	'M' = Magnetic
double	speedOGN	Speed over ground, in knots.
char	cKnots	'N' = Knots.
double	speedOGKm	Speed over ground, in km/hr
char	cKm	'K' = Kilometres per hour.
BINARY_TUPLE_FOOTER (see page 3)		

3.21. DATA_TYPE_ZDA = 1008

Time & Date — UTC, Day, Month, Year, and Local Time Zone

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
double	utc	UTC. Number of milliseconds since midnight.
uint16	wDay	Day, 01 to 31.
uint16	wMonth	Month, 01 to 12.
uint16	wYear	Year : 1999.
int16	zoneHour	Local zone description, 00 to +/-13 hrs.
int16	zoneMin	Local zone minutes description.
BINARY_TUPLE_FOOTER (see page 3)		

4. DATA SET USED IN SONAR / PROFILE TUPLE

This section describes each data set used inside of the sonar or profile Tuple.

4.1. HC1000_DATA_HEADER

This is a common header used within the various sonar and profile data Tuples.

Data Type	Member	Description
uint16	HeaderSizeBytes	The size in bytes of this packet.
uint16	DataType	One of the following values: 1 = Sonar Data 2 = Sonar Data 3 = Profile Data 4 = Profile Data
uint16	DataNumber	Number of pings in this Tuple.
uint16	DataSizeBytes	The size in bytes of the data packet following this header packet.
uint16	DataFormat	One of the following values: 0 = character 1 = 8-bit signed integer 2 = 8-bit unsigned integer 3 = 4-bit signed integer 4 = 4-bit unsigned integer 5 = 16-bit signed integer 6 = 16-bit unsigned integer
uint16	Timestamp1	Timestamp relative to command (in milliseconds).
uint16	Timestamp2	Timestamp relative to command (in milliseconds).
int16	Direction1a	First stop angle.
int16	Direction1b	Last stop angle.
int16	Direction2a	First stop angle.
int16	Direction2b	Last stop angle.

4.2. EXTRA_PROCESSED_PROFILE_TUPLE_DATA

This is a common header used within profile Tuples, where the profile was processed in the head.

Data Type	Member	Description
uint16	BytesInPacket	The size in bytes of this packet.
uint16	StepSize	Step size of head in 0.225^{ths} of a degree.
uint16	StartAngle	Start angle in 0.225^{ths} of a degree steps.
uint16	TimeStamp1	Timestamp relative to command (in milliseconds).
uint16	TimeStamp2	Timestamp relative to command (in milliseconds).
byte	ScanDirection	Clockwise = 0 Counterclockwise = 1
byte	ZoomFactor	True zoom factor (1, 2, 4, etc)
byte	NumProfilePoints	Number of profile points per ping.
byte[7]	Reserved	Reserved

4.3. EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA

This is a common header used within the various sonar and profile data Tuples.

Data Type	Member	Description
uint16	BytesInPacket	size of the following DATA packet
byte	DisplayMode	One of the HC_DISP_XXX constants specifying the display mode.
byte	ScanDirection	0 = Clockwise 1 = Counterclockwise
uint16	StepSize	Step size of head in 0.225^{ths} of a degree.
uint16	StartAngle	Start angle of scan in 0.225^{ths} of a degree steps.
uint16	SectorHeading	Sector heading in 0.225^{ths} of a degree steps from head's 0 degree heading.
byte	Upsample	Number of times to upsample sonar data
byte	BitsPerSample	Number of bits per each sample.
byte	ZoomFactor	True zoom factor (1, 2, 4, etc)
byte	Version	The version. Matches HeadControllerVersion in HC_OPERATION_INFO.
uint32	SamplingRate	Sampling rate in hertz.
uint16	SamplingDelay	Sampling delay in samples.

4.4. HC_OPERATION_INFO

This data set includes incrementing values for the HeadControllerVersion member. Any other members that are present only in a certain range of version are marked with their range in *italics.*

Data Type	**Member**	**Description**
int16	ScanStopsLeft	Scan axis limit, at left, in 0.225^{ths} of a degree. Applies only to non continuous heads with mechanical stops.
int16	ScanStopsRight	Scan axis limit, at right, in 0.225^{ths} of a degree. Applies only to non continuous heads with mechanical stops.
int16	SectorLeft	Left sector limit of the sweep, in 0.225^{ths} of a degree.
int16	SectorRight	Right sector limit of the sweep, in 0.225^{ths} of a degree.
uint16	ProfileNumReturns	Number of profile points returned per ping.
uint16	ProfileMinThreshold	Profiling minimum threshold, 0 to 255.
uint16	ProfileMaxThreshold	Profiling maximum threshold, 0 to 255.
uint16	ProfileMinRange	Profiling minimum range, in units of 2 cm (0 -> 1300 meters).
uint16	ProfileMaxRange	Profiling maximum range, in units of 2 cm (0 -> 1300 meters).
uint16	ProfileMinWidth	Profiling minimum width, in units of 2 cm (0 -> 1300 meters).
uint16	ProfileMinGap	Profiling minimum gap between profile points found in same ping, in units of 2 cm (0 -> 1300 meters).
uint16	ProfileReserved	Reserved
uint32	BaudRate	Uplink baud rate, in bits per second.
uint16	SoundSpeedUser	The user selected sound speed (m/s). For old recorded files, should not be used — SoundSpeedDefault should be used instead.

		HeadControllerVersion ≥ 11.
byte[16]	FutureExpansion1	Reserved.
byte	AutoTvg	Whether automatic Time Varying Gain is enabled: 0 = Disabled 1 = Enabled *HeadControllerVersion ≥ 10.*
byte	TestTone	Indicate if using test tone to do cable calibration 0 = Not using 1 = Using *HeadControllerVersion ≥ 9.*
int16	MinCableGain	Minimum cable gain, 0 to 499. *HeadControllerVersion ≥ 9.*
int16	MaxCableGain	Maximum cable gain, 0 to 499. *HeadControllerVersion ≥ 9.*
char	DualTransmitRatio	Primary / auxiliary transmit ratio (x = DualTransmitRatio): < 0 = 1 ping aux for each x pings pri 0 = aux only > 0 = x ping pri for each x pings aux ≥ 51 = pri only *HeadControllerVersion ≥ 9.*
uint16	TiltSectorHeading	Tilt axis scanning sector heading, in 0.225^{ths} of a degree. *HeadControllerVersion ≥ 9.*
byte	GainShiftFactor	Possible values: 4 = High 5 = Medium 6 = Low *HeadControllerVersion ≥ 9.*
byte	OversamplingFrequencyFactor	Oversampling frequency factor = Oversampling frequency / sample rate. *HeadControllerVersion ≥ 8.*
uint16	AssociationFlag1	Each bit indicates a type of property of the Head. *HeadControllerVersion ≥ 7.*
uint16	AssociationFlag2	Each bit indicates a type of property of the Head. *HeadControllerVersion ≥ 7.*
byte	HiResSampAcquisFactor	Multiplication factor used to increase the number of samples acquired from the head associated with standard display modes. *HeadControllerVersion ≥ 5.*
byte	HardwareSyncMode	Hardware Synchronization: 0 = Disabled 1 = Input 2 = Output 3 = IO
byte	Bandwidth	Bandwidth control: For TTM: 0 = Maximum (default) 1 = Wide 2 = Medium 3 = Narrow For 1071 newer digital heads 0 = Automatic (default) 1 = Wide

		2 = Narrow 3 = Narrow
uint16	SectorHeading	Sector heading, in 0.225^{ths} of a degree.
uint16	SectorWidth	Sector width, in 0.225^{ths} of a degree.
uint32	HighFrequencySampRate	Altimeter has a much higher sampling rate limit *HeadControllerVersion = 3, 4, 5 (for altimeter only)* *HeadControllerVersion ≥ 6 (for all heads).*
uint16	StepSize	Angular size of each step, in 0.225^{ths} of a degree.
uint16	HeadAngle	Current head angle, in 0.225^{ths} of a degree.
uint16	SampRate	Sampling rate in Hz *HeadControllerVersion = 1, 2 (for all heads).* *HeadControllerVersion = 3, 4, 5 (for all heads except altimeter).*
uint16	SampCount	Number of samples in each ping.
uint16	SampDelay	Delay before the start of sampling in samples at current rate.
uint16	Range	Operating range in 10^{ths} of a metre.
uint16	Gain	Gain applied to the signal, 0 to 255.
uint16	PulseLength	Transmit pulse length in microseconds.
uint16	MultiPingCount	MultiPingCount & ~0x8000: Number of pings in each multiping command. MultiPingCount & 0x8000: 0 = Multiping disabled 1 = Multiping is enabled
byte	StepAck	This is the opto state returned by head: 0 = top right quadrant 1 = top left quadrant 2 = bottom right quadrant 3 = bottom left quadrant
byte	CommID	Assigned communication ID for this Controller
byte	Paused	0 = Head controller not on hold 1 = Head controller on hold
byte	ImageProfMode	Bit 0: 0 = Image not enabled 1 = Image enabled Bit 1: 0 = Profile disabled 1 = Profile enabled
byte	TransmitterOn	0 = Transmitter off 1 = Transmitter on
byte	XDCRorientation	Transducer mounting orientation: 1 = Up 0 = Down
byte	DisplayMode	Display mode for head: 1 = Polar 2 = Sector 3 = Sidescan 4 = Echosounder 5 = Cylindrical 6 = Sounder 7 = Net Opening 8 = Horizontal 9 = Vertical 10 = Multibeam

		11 = History 12 = Vertical Fan 13 = Net Opening 14 = True Ratio Echosounder
byte	ZoomFactor	True zoom factor (1, 2, 4, etc)
byte	ScanDirection	Direction of sweep in each multiping command: 0 = Clockwise 1 = Counterclockwise
byte	BitsPerSample	Number of bits per sample.
byte	UpSample	Upsampling rate factor due to head sampling rate limitation.
byte	BeamType	Switchable fan cone: 0 = Fan transmit beam 1 = Cone transmit beam
uint16	HidPartNumber1	Head id part number 1.
uint16	HidPartNumber2	Head id part number 2.
uint32	HidSerialNumber	Head serial number.
uint32	HidType	Head type: 0x0440 = 971 scanning heads 0x0971 = 971 scanning heads 0x1071 = 1071 scanning heads 0x1007 = 1007 altimeter heads 0x1001 = 1001 telemetry translation module 0x2000 = multibeam heads 0x2180 = multibeam heads (180 degree, 160 elements)
uint16	HidVersion1	Head version part 1.
uint16	HidVersion2	Head version part 2.
uint32	HidData1	Head data 1.
uint32	HidData2	Head data 2.
uint16	HidMaxBaudrateIn300bps	Maximum baud rate in 300bps increments.
uint16	HidMaxMessageDataBytes	Maximum bytes of data per message.
uint16	HidReserved	Reserved.
uint16	HidTelemetryVersion	Telemetry version.
char[36]	HeadContName	Name assigned to the controller.
uint16	ZoomOffset	The pixel offset of the zoom origin before applying the zoom, along the new heading.
uint16	SoundSpeedDefault	The default sound speed for this head, in meters per second.
byte	HeadControllerVersion	The version of this Tuple. Must be checked to see which members are specified.
byte	SyncState	Bit-0: 0 = Synchronization disabled 1 = Synchronization enabled Bit-1: 0 = Slave 1 = Master Controller
byte	SyncMask	Bit mask of the CommID in head for synchronization
byte	OperationMode	Bit-0: 0 = Auto tilt correction off 1 = Auto tilt correction on Bit-1: 0 = Load ROM time varying gain

		1 = Load RAM time varying gain Bit-2: 0 = Fan beam 1 = Cone beam Bit-3: 0 = High power 1 = Low power Bit-4: 0 = Activate azimuth axis 1 = Activate tilt axis Bit-5: 0 = Xmit on 1 = Xmit off
byte	TvgType	Time varying gain type: 0 = 20LOG 1 = 30LOG 2 = 40LOG 3 = TEST 4 = USER
byte	TvgAFactor	Time varying gain A factor.
uint16	TvgBFactor	Time varying gain B factor.
int16	TvgCFactor	Time varying gain C factor.
byte	TvgLimit	Time varying gain curve limit, 0 to 100 dB.
int16	TiltAngle	Tilt axis angle, in 0.225^{ths} of a degree.
uint16	CableCalibration	
byte	TiltMode	0 = Auto tilt step on 1 = Auto tilt step on
byte	TiltStepsize	Angular size of each step on tilt axis, in 0.225^{ths} of a degree.
int16	MaxTilt	Maximum angle on tilt axis, in 0.225^{ths} of a degree.
int16	MinTilt	Minimum angle on tilt axis, in 0.225^{ths} of a degree.
int16	TiltStopsLeft	Tilt axis limit, at left, in 0.225^{ths} of a degree. Applies only to non continuous heads with mechanical stops.
int16	TiltStopsRight	Tilt axis limit, at right, in 0.225^{ths} of a degree. Applies only to non continuous heads with mechanical stops.
byte	HeadJumperSetting	Jumper setting of the analog 1071 heads.
uint16	HeadSoftwareVersion	Software version of the analog 1071 heads.
byte	AvailableSensors	Bitwise definition of available head sensors: Bit-1: Proll type from HSTS Bit-2: Nmea Compass Bit-3: Ksi TX / sensor module
byte	TrawlSensorReadInterval	This is the time interval between each Trawl Sensor read in seconds. If set to 0, then we are in sensor test mode

4.5. OVERLOAD COUNTER

This is a special uint16 appended to newer sonar data. It is not part of the image.

Data Type	Member	Description
uint16	OverloadCounter	Indicates signal saturation in the head.

To check whether or not the overload counter is present, you need to check the DATA_TYPE_SETTINGS Tuple for that head. Either of the following cases will result in a overload counter being present:

1. If the HidPartNumber2 == 0x1001 and the HidVersion1 ≥ 0x0220.
2. If the HidPartNumber2 == 0x1071 and the HidVersion1 ≥ 0x0220.

5. SONAR / PROFILE TUPLE

This section describes each sonar or profile Tuple.

5.1. DATA_TYPE_SONAR = 12

This Tuple contains a single ping of singlebeam sonar data.

BINARY_TUPLE_HEADER (see page 3)		
EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA (see page 14)		
Data Type	**Member**	**Description**
byte[…]	SonarData	The sonar data. Each BitsPerSample bits represents a single sample. For example, if BitsPerSample is 8, every byte specifies a value between 0 and 255, where 255 is the strongest return and 0 is no return at all.
OVERLOAD COUNTER (see page 19)		
BINARY_TUPLE_FOOTER (see page 3)		

5.2. DATA_TYPE_PROFILE = 13

This Tuple contains a single ping of singlebeam profile data (surface processed).

BINARY_TUPLE_HEADER (see page 3)		
EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA (see page 14)		
HC1000_DATA_HEADER (see page 13)		
PROFILE DATA		
Data Type	**Member**	**Description**
byte[…]	ProfileData	The profile data. The length of this data is HC1000_DATA_HEADER.DataNumber * HC1000_DATA_HEADER.DataSizeBytes. Each entry specifies the sample number at which the profile point was found (there may be more than one point per ping).
BINARY_TUPLE_FOOTER (see page 3)		

5.3. DATA_TYPE_SONAR_PROFILE = 14

This Tuple contains a single ping of singlebeam sonar data combined with its profile data. There are two cases here:

1. Sonar data first, then profile data.

(HC1000_DATA_HEADER 1).DataType == 1 (Profile Data 1)
or
(HC1000_DATA_HEADER 1).DataType == 2 (Profile Data 2)

BINARY_TUPLE_HEADER (see page 3)

EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA (see page 14)

HC1000_DATA_HEADER 1 (see page 13)

Data Type	Member	Description
byte[…]	SonarData	The sonar data. Each EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA.BitsPerSample bits represents a single sample. For example, if BitsPerSample is 8, every byte specifies a value between 0 and 255, where 255 is the strongest return and 0 is no return at all.

OVERLOAD COUNTER (see page 19)

HC1000_DATA_HEADER 2 (see page 13)

Data Type	Member	Description
byte[…]	ProfileData	The profile data. The length of this data is HC1000_DATA_HEADER.DataNumber * HC1000_DATA_HEADER.DataSizeBytes. Each entry specifies the sample number at which the profile point was found (there may be more than one point per ping).

BINARY_TUPLE_FOOTER (see page 3)

2. Profile data first, then sonar data.

(HC1000_DATA_HEADER 1).DataType == 3 (Sonar Data 1)
or
(HC1000_DATA_HEADER 1).DataType == 4 (Sonar Data 2)

BINARY_TUPLE_HEADER (see page 3)

EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA (see page 14)

HC1000_DATA_HEADER 1 (see page 13)

Data Type	Member	Description
byte[…]	ProfileData	The profile data. The length of this data is HC1000_DATA_HEADER.DataNumber * HC1000_DATA_HEADER.DataSizeBytes. Each entry specifies the sample number at which the profile point was found (there may be more than one point per ping).

HC1000_DATA_HEADER 2 (see page 13)

Data Type	Member	Description
byte[…]	SonarData	The sonar data. Each

		EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA.BitsPerSample bits represents a single sample. For example, if BitsPerSample is 8, every byte specifies a value between 0 and 255, where 255 is the strongest return and 0 is no return at all.

OVERLOAD COUNTER (see page 19)

BINARY_TUPLE_FOOTER (see page 3)

5.4. DATA_TYPE_SETTINGS = 15

This Tuple contains the settings for the controller. These are sent out periodically or whenever they change (such as a change in range, etc).

BINARY_TUPLE_HEADER (see page 3)

EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA (see page 14)

HC_OPERATION_INFO (see page 14)

BINARY_TUPLE_FOOTER (see page 3)

5.5. DATA_TYPE_MULTI_SONAR = 21

This Tuple contains multiple pings of singlebeam profile data. The number of pings is specified by HC1000_DATA_HEADER.data_number.

BINARY_TUPLE_HEADER (see page 3)

EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA (see page 14)

HC1000_DATA_HEADER (see page 13)

Ping 1		
Data Type	**Member**	**Description**
byte[…]	SonarData	Each EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA.BitsPerSample bits represents a single sample. For example, if BitsPerSample is 8, every byte specifies a value between 0 and 255, where 255 is the strongest return and 0 is no return at all. The length of each ping can be determined by HC1000_DATA_HEADER.DataSizeBytes.

Ping 2. Same as above **.** **.**

OVERLOAD COUNTER 1 (see page 19) **OVERLOAD COUNTER 2 (see page 19)** **.**

BINARY_TUPLE_FOOTER (see page 3)

5.6. DATA_TYPE_PROCESSED_PROF = 24

This Tuple contains a single ping of singlebeam profile data (processed in the head).

BINARY_TUPLE_HEADER (see page 3)

EXTRA_PROCESSED_PROFILE_TUPLE_DATA (see page 13)

Ping 1		
Data Type	**Member**	**Description**
byte[…]	ProfileData	The profile data. The length of this data is EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA.BytesInPacket. Each entry specifies the sample number at which the profile point was found (there may be more than one point per ping).

BINARY_TUPLE_FOOTER (see page 3)

5.7. DATA_TYPE_TTM_DEF_HEAD_SETTINGS = 29

Settings for Telemetry Translation Module.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
byte	TvgType	Time varying gain type: 0 = 20LOG 1 = 30LOG 2 = 40LOG 3 = TEST 4 = USER
byte	TvgAFactor	Time varying gain A factor.
uint16	TvgBFactor	Time varying gain B factor.
int16	TvgOffset	DC offset added to the time varying gain curve (can be negative).
byte	TvgLimit	Time varying gain curve limit, 0 to 100 dB.
byte	TvgUiOffsetLimit	Low nibble (UI setting of the Offset): 0 = Default 1 = High 2 = Low High nibble (UI setting of the Limit): 0 = Default 1 = High 2 = Low
byte	OperationMode	Bit-0: 0 = Auto tilt correction off 1 = Auto tilt correction on Bit-1: 0 = Load ROM time varying gain 1 = Load RAM time varying gain Bit-2: 0 = Fan beam 1 = Cone beam Bit-3: 0 = High power 1 = Low power Bit-4:

		0 = Activate azimuth axis 1 = Activate tilt axis Bit-5: 0 = Xmit on 1 = Xmit off
byte	BeamType	Bit-0: 1 = Single beam Bit-1: 0 = Fan beam not available 1 = Fan beam available Bit-2: 0 = Cone beam not available 1 = Cone beam available
uint16	PulseLength	Pulse length in microseconds.
uint16	TransducerFrequency	Transducer frequency in hertz.
uint16	HighPowerDutyCycle	
uint16	LowPowerDutyCycle	
uint16	MaxPulseLength	Maximum pulse length in microseconds.
uint16	MinPulseLength	Minimum pulse length in microseconds.
byte[34]	Reserved	Reserved.

BINARY_TUPLE_FOOTER (see page 3)

5.8. DATA_TYPE_AUX_SETTINGS = 39

This Tuple contains head settings for an auxiliary transducer.

BINARY_TUPLE_HEADER (see page 3)

Data Type	Member	Description
byte[20]	Reserved	Reserved.

HC_OPERATION_INFO (see page 14)

BINARY_TUPLE_FOOTER (see page 3)

5.9. DATA_TYPE_AUX_SONAR = 40

This Tuple contains a single ping of singlebeam sonar data from an auxiliary transducer.

BINARY_TUPLE_HEADER (see page 3)

EXTRA_HIGH_FREQ_SONAR_TUPLE_DATA (see page 14)

Data Type	Member	Description
byte[…]	SonarData	The sonar data. Each BitsPerSample bits represents a single sample. For example, if BitsPerSample is 8, every byte specifies a value between 0 and 255, where 255 is the strongest return and 0 is no return at all.

OVERLOAD COUNTER (see page 19)

BINARY_TUPLE_FOOTER (see page 3)

5.10. DATA_TYPE_TTM_DEF_AUX_HEAD_SETTINGS = 41

Settings for Telemetry Translation Module for an auxiliary transducer.

Same as **DATA_TYPE_TTM_DEF_HEAD_SETTINGS = 29 (see page 23)**.

6. OTHER TUPLE TYPES

6.1. DATA_TYPE_DATE_VERSION = 22

This Tuple is recorded at the start of each volume to give an absolute date (time relative to this date is contained within the Tuple headers themselves).

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
uint32	TimeUTC	UTC time. The number of seconds elapsed since midnight (00:00:00), January 1, 1970, coordinated universal time, according to the system clock.
uint16	Version	Version of file format
byte[26]	FutureExpansion	Reserved
BINARY_TUPLE_FOOTER (see page 3)		

6.2. DATA_TYPE_OFFSET = 23

Mounting offsets of a source with respect to a ship.

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
float	XOffset	X offset in meters.
float	YOffset	Y offset in meters.
float	ZOffset	Z offset in meters.
float	XRotOffset	X offset in degrees.
float	YRotOffset	X offset in degrees.
float	ZRotOffset	X offset in degrees.
uint32	Mounting	0 = Down / Fore 1 = Up / Aft
BINARY_TUPLE_FOOTER (see page 3)		

6.3. DATA_TYPE_EVENT = 30

BINARY_TUPLE_HEADER (see page 3)		
Data Type	**Member**	**Description**
uint16	EventId	The id of the event.
uint32	TimeUtc	UTC time. The number of seconds elapsed since midnight (00:00:00), January 1, 1970, coordinated universal time, according to the system clock.
UINT	EventType	The event type: 34014 = Marker 0 34018 = Marker 1 34019 = Marker 2 34020 = Marker 3 34021 = Marker 4 34022 = Marker 5 34023 = Marker 6 34024 = Marker 7 34025 = Marker 8

		34026 = Marker 9 34027 = Marker 10 34028 = Marker 11 34029 = Marker 12 34030 = Marker 13 34031 = Marker 14 34017 = Marker 15 34015 = Marker 16 34016 = Marker 17 34145 = Marker 18 34146 = Marker 19 34183 = Reference marker 34211 = Dash marker 34211 = External event marker 34212 = Boat position marker 34213 = Start of survey line marker 34214 = End of survey line marker
double	Latitude	The latitude position of the marker.
double	Longitude	The longitude position of the marker.
byte	Flags	Bit 0: 0 = Latitude / longitude data invalid 1 = Latitude / longitude data valid
double	X	The position of the marker with respect to vessel coordinate system, in meters.
double	Y	The position of the marker with respect to vessel coordinate system, in meters.
double	Z	The position of the marker with respect to vessel coordinate system, in meters.

BINARY_TUPLE_FOOTER (see page 3)

6.4. DATA_TYPE_ASCII_STRING = 32

BINARY_TUPLE_HEADER

Data Type	Member	Description
Char[]	ASCII string	Variable length of ASCII string, null terminated.

BINARY_TUPLE_FOOTER

6.5. DATA_TYPE_SONAR_GEO_INFO = 54

BINARY_TUPLE_HEADER

Data Type	Member	Description
double	Longitude	Sonar position Longitude in decimal degrees
double	Latitude	Sonar position Latitude in decimal degrees
double	Magnetic heading	Sonar magnetic heading in decimal degrees
double	Magnetic variation	Magnetic variation in decimal degrees
byte[120]	Reserved	120 bytes reserved field
bool	On tripod	Sonar on tripod (true, false)
bool	Display north up	Sonar display north up (true, false)
Byte[124]	Reserved	124 bytes reserved field

BINARY_TUPLE_FOOTER

6.6. DATA_TYPE_FISH_SENSOR_SETUP = 49

BINARY_TUPLE_HEADER		
Data Type	**Member**	**Description**
uint16	nNumSensors	Number of enabled sensors
uint16	nNumMeasures	Number of defined measures
uint16	nMaxMPS	Maximum number of measures per sensor
uint16	nMaxSensors	Maximum number of supported sensors
bool	bCatchAlarm	True if the catch alarm is active
bool	bIntFilter	True if interference filter is enabled
uint16	iIntFilterLevel	0 – 9, default 9, highest filter level
uint16	iReceverFilterLevel	0 – 3, sensor receiver filter level, 3, highest
bool	bAGC	True if AGC (Auto Gain Control) is enabled
uint16	iGainIndB	Manual gain in dB, default 0
bool	bManualPathFilter	True if manual path filter is enabled. Default False
uint16	iMultipathFilterIndB	0 – 99, multipath filter level in dB. Default 42
uint16	iDetThresholdIndB	3 – 99, receiver detection threshold in dB. Default 8
float	fMaxShootingSpeed	1 – 5, maximum shooting speed in knots. Default 5
byte[234]	Reserved field	234 reserved bytes
Start of repeating "nNumSensors" entries of:		
byte	updateRate	0 – 2, update rate slow, medium, fast
byte	sensorPosition	0 – 2. 0: Port, 1: Starboard, 2: Other
byte[255]	reserved field	255 reserved bytes
End of repeating cycle "nNumSensors"		
Start of repeating "nNumMeasures" entries of:		
byte[5]	Reserved field	5 reserved bytes
uint16	measureType	1: catch 2: depth (300m) 3: depth (600m) 4: depth (1000m) 5: temperature 6: spread 7: spread extended 8: bottom contact 9: Geometry 10: Geometry extended 11: height 12: pitch 13: roll
unit16	sensorNumber	Specified which sensor this measure belongs to.
unit16	channel Number	The channel that this measure is configured to.
byte[64]	Reserved field	64 reserved bytes
bool	bEnabled	True if this measure is enabled.
unit16	sensorVariant	For depth sensor: 0: 300m 1: 600m 2: 1000m For spread sensor: 0: 350 m 1: 700 m For geometry: 0: 300m 1: 600m
float	fReadingOffset	Reading offset added to the measure reading

bool	SensorMount	True: fore, False: aft
byte[248]	Reserved field	248 reserved bytes
End of repeating cycle "nNumMeasures"		

BINARY_TUPLE_FOOTER

6.7. DATA_TYPE_FISH_MEASURE_STATUS = 50

BINARY_TUPLE_HEADER

Data Type	Member	Description
uint16	nNumMeasures	Number of defined measures
uint32	iTimeMasterTimer	When the master timer was started in milliseconds since midnight GMT.
bool	bMasterTimer	True is master timer has been started
byte[251]	Reserved field	251 reserved bytes
Start of repeating "nNumMeasures" entries of:		
uint16	measureType	1: catch 2: depth (300m) 3: depth (600m) 4: depth (1000m) 5: temperature 6: spread 7: spread extended 8: bottom contact 9: Geometry 10: Geometry extended 11: height 12: pitch 13: roll
unit16	sensorNumber	Specified which sensor this measure belongs to.
unit16	channel Number	The channel that this measure is configured to.
uint16	iIntFilterLevel	0 – 9, default 9, highest filter level
byte[64]	Reserved field	64 reserved bytes
bool	bTimerStarted	True if the timer has been started
uint32	iTimeTimer	When the timer was started in milliseconds since midnight GMT.
float	fReading	The reading of the measurement
float	fRateChange	The rate which the reading is changing, in the same unit as the specific measure, per minute.
byte	sensorPosition	0 – 2. 0: Port, 1: Starboard, 2: Other
byte[468]	reserved field	468 reserved bytes
End of repeat cycle "nNumMeasures"		

BINARY_TUPLE_FOOTER

6.8. PI sensor ASCII Sting data types

```
DATA_TYPE_PI_SENSOR_STRING =51;
DATA_TYPE_PI_SENSOR_GEOMETRY =52;
DATA_TYPE_PI_SENSOR_GEOMETRY_XT =53;
DATA_TYPE_PI_SENSOR_SPREAD =54;
DATA_TYPE_PI_SENSOR_SPREAD_XT =55;
DATA_TYPE_PI_SENSOR_PITCH =56;
DATA_TYPE_PI_SENSOR_ROLL =57;
DATA_TYPE_PI_SENSOR_CATCH =58;
```

```
DATA_TYPE_PI_SENSOR_DEPTH_300M =59;
DATA_TYPE_PI_SENSOR_DEPTH_600M =60;
DATA_TYPE_PI_SENSOR_DEPTH_1000M =61;
DATA_TYPE_PI_SENSOR_TEMPERATURE =62;
DATA_TYPE_PI_SENSOR_BOTTOM_CONTACT =63;
DATA_TYPE_PI_SENSOR_HEIGHT =64;
```

BINARY_TUPLE_HEADER		
Data Type	**Member**	**Description**
string	PIString	PI sensor sentences. Variable length

BINARY_TUPLE_FOOTER

KONGSBERG

1598 Kebet Way, Port Coquitlam, BC, V3C 5M5 Canada
Telephone +1 604 464 8144 Telefax +1 604 941 5423
www.km.kongsberg.com/mesotech