Fisheries Technology Committee

ICES CM 2002/B:04

Report of the

Planning Group on the HAC Data Exchange Format

Sète, France
7–8 June 2002

https://doi.org/10.17895/ices.pub.9608

International Council for the Exploration of the Sea

Conseil International pour l'Exploration de la Mer

Palægade 2–4 DK–1261 Copenhagen K Denmark

TABLE OF CONTENTS

1 OPENING OF THE MEETING

The meeting was Chaired by D. Reid, (UK, Scotland) who also acted as Rapporteur. A full list of participants is attached as Annex 1.

2 BACKGROUND

The terms of reference for the Planning Group on the HAC Data Exchange Format (PGHAC) as agreed at the WGFAST meeting (Seattle, USA April 2001) and approved at the ICES 89th Statutory Meeting, Oslo, Norway, September 2001 were:

a) continue to work on the HAC format in order to adapt it to the latest versions of equipment and to improve it;

b) provide information on the changes in the format and its evolution;

c) share information between manufacturers and users on the way acoustic data are processed and stored.

The Planning Group reported to the June 2002 meeting of WGFAST (Montpellier, France, 17 June 2002) and to the Fishing Technology Committee at the 2002 Annual Science Conference.

3 INTRODUCTION

In 1999 the WGFAST WG (meeting in St. John's, Newfoundland) adopted the ***HAC*** standard data format for raw and edited hydroacoustic data (Simard *et al.* 1997, 1999) as the common format for exchanging fisheries acoustics data and for comparing processing algorithms within the ICES community (ICES-CM 1999/B:2: Section 10.3, p. 12). A group of experts including FAST members and representatives of hardware and software manufacturers was assigned the responsibility of coordinating the development of the format. This included the examination of proposals to introduce new information in the ***HAC*** environment and the definition of a generic set of tuples for echosounders that were not covered by the already defined tuples[*] of this upgradable format. At the WGFAST in Haarlem, Netherlands, it was agreed that this was a major issue of importance to all members of the fisheries acoustic community and that a more permanent group should be set up. This was proposed at the ASC in Bruges, Belgium (September 2000) and was formally incorporated as an ICES Planning Group (PGHAC, ICES Annual Report for 2000. Part 3. p. 256).

4 SUBJECTS ADDRESSED

The PG discussed the following main issues:

- Review progress on development of new platform attitude tuples (41 and 10140) – assigned for developers in 2001.
- Review development of Single Target Information Tuples (4000 and 1009) – Assigned for developers in 2001.
- Decision on the use of Big or Little Endian format in files.
- Determine what format of "char" to use as standard.
- Determine desirability of variable v. fixed length tuples.
- Review progress on new set of tuples for the Simrad EK60 echosounder (210 and 2100).
- Review list of Tuples required for HAC compatibility.

> The new tuples and changes to existing tuples are detailed in the annex and are only described briefly here. For more detail please see Annex 2.

[*] Tuple: a labelled group of bytes encapsulating special type of information in the ***HAC*** format, which forms the basic structure of this format and that gives the format its upgradability and versatility property. Tuples belongs to tuple families or classes that groups the information by themes. Unique numbers, varying from 0 to 65535, identify each tuple. The ***HAC*** co-ordinating committee has to allocate these numbers to prevent any "collision" in the tuple usage by various groups around the world and to agree on the definition of the various fields of information they contain.

5 REVIEW PROGRESS ON DEVELOPMENT OF NEW PLATFORM ATTITUDE TUPLES (41 AND 10140) – ASSIGNED FOR DEVELOPERS IN 2001

At the 2001 meeting it was agreed that there were some problems with this tuple as it stood. In particular it was not possible to relate the tuple to a particular channel. It was agreed that there was a need for a "sub-channel" style tuple – containing the description and relations of the attitude sensor system, and a "ping style" tuple – containing the data flowing from that sensor. The new tuples and structures were presented at the meeting. One problem found was that there was no field for sensor height offset. It was agreed that this should be added. The completed tuples were approved subject to the standard two-month response period.

As a result of discussions, it was apparent that the new tuples did not allow for the use of attitude sensor systems to relate the position of a towed system to a vessel. It was agreed to define two new tuples – a "sub-channel" tuple (42) and a "ping style" tuple (10142) for the data flow. These were allocated to IFREMER for development and will be presented at the next meeting of PGHAC.

For format see Annex 2, Tables A.2.4a and b.

6 REVIEW DEVELOPMENT OF SINGLE TARGET INFORMATION TUPLES (4000 AND 10090) – ASSIGNED FOR DEVELOPERS IN 2001

The latest state of development of these tuples was approved by PGHAC. However, it was determined that there was no field available for the inclusion of the TS Gain figure available from SIMRAD systems. It was agreed that this should be included in the channel tuple 2001. However, in line with the new policy of using "patches" (see Section 9 below) rather than modifying complete tuples, the group assigned such a patch tuple (2002) to include both TS and S_v gains. It was also agreed that this set of tuples should be designed as "generic" and be suitable for all sources of single target information, not just SIMRAD EK500. The tuples should have "not available" and "not applicable" codes permissible in each field to allow this flexibility. These codes were agreed on at the 2001 meeting of PGHAC.

For format see Annex 2, Tables A.2.5a and b.

7 DECISION ON THE USE OF BIG OR LITTLE ENDIAN FORMAT IN FILES

The issue of big versus little "endian" data formats was discussed at the 2001 meeting of PGHAC.

> These are both permissible in the format to enhance platform independence. It was proposed and agreed that this was probably not necessary given that all known software developed for the HAC format are on the PC platform. Therefore it should only be necessary for software to read or write in little endian format (PC platform) to be ***HAC*** compliant. DFO and IFREMER will investigate consequences of this change, and this will be discussed at PGHAC in 2002.

Following investigation it was agreed that there was no need for both formats to be used and that software could be coded in little Endian only. It was agreed by the group that the HAC standard be modified to allow only data files written in little Endian format only.

8 DETERMINE WHAT FORMAT OF "CHAR" TO USE AS STANDARD

A number of formats are described in the literature for "char". The earliest is 7 bit. However, this does not allow for the coding of any non-English letters. 8 bit char does allow for these, and a variety of other symbols. However, there is no single agreed international structure for 8-bit char, which could result in confusion between countries, developers and software. The group agreed that in the light of this only 7-bit char would be specified.

It should be noted that this decision was made by a panel which, included members from a number of countries whose characters would not be available.

9 DETERMINE DESIRABILITY OF VARIABLE V. FIXED LENGTH TUPLES

The potential for using variable rather than fixed length tuples was discussed at the 2001 meeting of PGHAC.

Variable length tuples – It was proposed that at the next revision ALL tuples should have a variable rather than a fixed length. This would allow new fields to be added to existing tuples without major revision updates. Older software would simply not read the extra fields. The size of the tuple would remain in the first and last fields. This will be examined by DFO and others and discussed at PGHAC in 2002.

The PG agreed that, in principle, the use of variable length tuples would be desirable. However, it was recognised that this would be a major re-coding task. It was therefore agreed that:

- There would be no rewrite for existing tuples.
- New tuples (after this meeting) would be variable in length and any software should read the tuple for length prior to use.
- Where small modifications needed to be made to existing, fixed length, tuples, this would carried out using "patch tuples". These would be small, variable length, tuples. The original tuple to which the patch refers will indicate the presence of the patch using an entry in the "attribute" field.

This proposal is subject to the standard two-month approval period.

As there has been some confusion about the exact specification of the attribute field, a description of this field and the permissible entries have been included in this report (Table A.2.6). Developers and users should note that this may be modified at the 2003 meeting of PGHAC.

10 REVIEW PROGRESS ON NEW SET OF TUPLES FOR THE SIMRAD EK60 ECHOSOUNDER (210 AND 2100)

The new EK60 tuples were presented by the developers at SIMRAD. The new tuples are not finalised as yet and the PG addressed a number of questions raised by the developers.

- SIMRAD developers asked if it was possible for a single, up-to-date version of the HAC format could be compiled and made available. At present, developers etc. must consult a range of separate documents to find the full specification as it evolves. The PG agreed in principle, that a single authoritative version would be desirable, but it was not clear who would do this work.
- SIMRAD programmers asked to be allowed to use FLOAT and DOUBLE style floating point variables, as these could be more flexible. The PG felt that the known inaccuracies of floating point variables precluded their use and that specified structures using LONG should be retained.
- SIMRAD wanted "Practical ranges" as well as limit ranges to be printed in the limit range field. Bearing in mind that in some cases these can be the same, the PG agreed.
- SIMRAD wanted to provide a mechanism for archiving complex samples from the four quadrants of a split beam transducer. This was agreed and the PG assigned two new tuples: 4010 as a sub channel tuple to describe the system, and 10002 to store complex ping data. It was further agreed that these should be designed for ANY complex ping data not just EK60.
- Other general improvements were suggested e.g., too many system details were being put into the remark field. These should be given their own specified fields. Some changes were suggested in the data types used e.g., in char field length and blanking etc.

11 REVIEW LIST OF TUPLES REQUIRED FOR HAC COMPATABILITY

The PG reviewed the list of tuples required for ***HAC*** compliance. The new list is presented below (see Section 16).

12 DATA VOLUME/DATA COMPRESSION

The problems of increasing amounts and volumes of data to be archived was discussed. One solution to this problem was deemed to be the use of common sense restrictions on the amount of data stored, e.g., do not store data below the noise threshold. This could involve using passive calculated noise as a cut-off. Another solution was to store the data in 16 rather than 32 bit form. To this end it was agreed that the C16 tuple for storing compressed 16 bit data (10040) should be included in the standard from now on. This tuple will also be modified in line with the change made in the

2001 report. Essentially this simply involved a clarification of the description of the sample value field (offset byte 28). This was believed to be unclear and possibly unusable.

For format see Annex 2, Table A.2.2.

13 GENERAL ITEMS

13.1 General sub-channel assignments

The Planning Group decided that the general philosophy for these should be as follows:

- Tuples 4000 – 4009 sub channels for TS parameters
- Tuples 4010 – 4019 sub channels for parameters describing the collection of complex data e.g., EK quadrant data
- Tuples 4020 – 4029 sub channels for parameters of bottom classification systems.

13.2 Proposed new tuples for use by developers:

- Tuple 30 was assigned for the collection of through the water speed of the vessel
- Tuple 31 was assigned for recording of the vessel log pulses.

13.3 Quester Tangent

Quester Tangent of Canada has made a request for assignment of tuples for use in the QT substrate discrimination system. The PG recognised that this was an appropriate request, as it would facilitate data exchange. However, no delegate from QT was able to attend so discussion was truncated. The group agreed that QT should be able to develop the use of HAC for their approach and could use the Private Tuple (65937 – QTC code – 7 in the third field) for the purposes of initial developments and submit these to the PG next year for consideration.

The following items are reprinted from the 2001 report to provide guidance for users.

- **Temporary and provisional tuples** – These are assigned specifically to allow developers to produce new software. They should not be used in shipped software until approval and adoption is finalised by the Committee.
- **New tuples** – Unless stated otherwise new tuples will be considered provisional for two months after the meeting to allow suggestions and objections to be made. Following that, and if no objections are raised, they will be accepted.
- **Allowable items in fields** – All programmers should be aware that new values might be added at any revision to individual fields. The current lists of options may be added to and code should be developed with the expectation of ANY permissible values.
- **Version Numbering** – The current release will be entitled HAC 1.4. two months after the Montpellier meeting (i.e., end of August 2002). New releases will be incremented in the first decimal after each annual meeting as required.
- **Precision and "unavailable/not known" data** – From now on, and where possible, LONG format data fields will be used to allow a four decimal place precision, and a larger range of permitted values. Where possible, "unavailable/not known" will generally use the maximum permissible value for unsigned fields and the minimum permissible value for signed fields.

14 TUPLE ALLOCATION RULES

The following is reprinted from the last report and is included here for the guidance of users and developers.

"The rules for allocating tuple numbers and accepting new tuple definitions: the basic tuples and the optional tuples of the common data format.

*To ease the use of the **HAC** format by various software developers requiring the addition of new tuples, and to facilitate the work of the coordinating Committee, the tuple classes were divided in two groups. A first group is the basic tuples classes for which any tuple addition will require a thorough examination and a unanimous agreement by the coordinating committee. Tuple numbers will be allocated temporarily to the applicants during their definition and* *debugging period for a maximum of 14 months, after which they will be retired if the committee has not accepted their description. (See below; the Committee will meet annually to resolve outstanding issues). A second group is the optional tuple classes that concern auxiliary information or secondary level of data analysis. For these classes, the committee will allocate tuple numbers at the request of the users, on presentation of a short justification and objectives of the tuple by the applicant. In addition there is a need to define the minimum tuples required to define the minimum needs of a* ***HAC*** *compliant file.*

The Basic tuple classes are: *Position tuples, Navigation tuples, Platform attitude tuples, Echosounder tuples, Channel tuples, Ping tuples, Threshold tuples, Environmental tuples for sound speed profiles, Opening and closing file tuples, End of file tuples and the HAC signature tuple.*

The Optional tuple classes are: *Mission and project tuples, Event marker tuples, Edition tuples, Classification tuples, Environmental tuples except sound speed profiles, Private tuples, and Index tuples.*

The minimum tuples in a HAC file are: *Position tuples, an Echosounder tuple, a Channel tuple, Ping tuples, a Threshold tuple, an End of file tuple and the* ***HAC*** *signature tuple."*

15 NEW OR RECENTLY ADDED TUPLE NUMBERS

Since the initial definition of the ***HAC*** version 1.0, the following tuples numbers were added to the list of defined tuples or in use: 39, **41, 210**, 300, 301, **901, 1001, 2001, 2100**, 3000, 3001, **4000,** 5000, 5001, **9001**, **10011,** 10039, **10090,** 10119, 12000, 12005, 12010, 12050, 12051, 12052, 12053, 12100, 13000, 13500, 14000, 65397, 65406. *Numbers in bold represent those tuples added at the 2000 meeting.*

Modified tuples included at the 2002 meeting were:

- 10040 Change in content field at offset byte 28 – compression description (Table A.2.2)
- 2001 Change in attribute filed to recognise patch 2002. (Table A.2.3a)
- 2002 Patch tuple for 2001 with TS and S_v gains (Table A.2.3b)

New tuples included at the 2002 meeting were:

- 30 Tuple for vessel speed through the water (temporary – for development)
- 31 Tuple for recording vessel log pulse (temporary – for development)
- 41 Platform attitude parameter tuple. (Table A.2.4a)
- 10140 Platform attitude sensor data tuple (Table A.2.4b)
- 42 Towed body position sub channel tuple (temporary – for development)
- 10142 Ping style tuple for towed body position data (temporary – for development)
- 4000 Split-beam detected single target parameters sub-channel tuple. (Table A.2.5a)
- 10090 Split-beam detected single-target tuple. (Table A.2.5b)
- 4010 Complex data parameter sub-channel tuple (temporary – for development)
- 10002 Complex data ping data (temporary – for development)

16 *HAC* COMPLIANCE AND *HAC* COMPATIBILITY

A data file is defined as ***HAC*** compliant if it conforms to the ***HAC*** syntax rules, contains the minimum required ***HAC*** tuples described above using the exact tuple format described (Simard *et al.* 1997 or subsequent updates).

A software application tool is defined as ***HAC*** compatible if it can read and/or write, and use a minimum number of commonly used basic tuples, in the little endian format used by PC platforms. These tuple numbers are: 20, 100, 200, 901, 1000, 2000, 2001, 9001, 10000, 10001, 10010, 10011, 10100, 65534 and 65535. Tuples 900 and 9000 were replaced by 901 and 9001, respectively in 2001. Tuples 65516 and 65517 (start and end of run) are no longer required for ***HAC*** compatibility.

The PG added two new tuples to the standard this year:
2002 – A patch for 2001 to include TS and S_v gains
10040 – C16 – 16 bit compressed format for ping data.

The above paragraph has been left in the report to allow comparison with previous reports. Following these changes, the current list of tuples for HAC compatibility is:

- **20, 100, 200, 901, 1000, 2000, 2001, 2002, 9001, 10000, 10001, 10010, 10011, 10040, 10100, 65534 and 65535**

This represents an update on the published definitions in last year's report.

The following table represents the ability of some of the currently available data acquisition software to read and write the above list of tuples and therefore their HAC compatibility.

	Data Acquisition/Processing Software			
Tuple number	**CH1(ver. 3.1)**	**CH2(ver 2.1)**	**Echoview (ver 2.1)**	**Movies+ (ver. 3.2)**
20	W	R	RW	RW
100	W	R	R*	
200	W	R	R	RW
901	N/A	R*W*	R*W*	RW
1000	W	R	R*	
2000	W	R	R	R
2001	W	R	R*	RW
2002	W*	R*		
9001	N/A	R*W*	R*W*	RW
10000	W	R	RW	R
10001	W	R	R*W*	
10010	W	RW		R
10011	W	RW		
10040				RW
10100	WW	R	W*	RW
65534	W	R	RW	RW
65535		R	RW	RW

* Represents implementation planned for 2002/2003 – **Bold** represents new to standard.

17 RECOMENDATIONS

It was agreed that the group should meet again at the same time as the FAST meeting in April 2003 under the Chairship of D. Reid. The Terms of Reference should remain the same as for the present meeting.

18 REFERENCES

Simard, Y., McQuinn, I., Diner, N., and Marchalot, C. 1999. The world according to ***HAC***: summary of this hydroacoustic standard data format and examples of its application under diverse configurations with various echosounders and data acquisition software. ICES-Fisheries Acoustics Sciences and Technology meeting, St. John's, Newfoundland, Canada, 20–22 April 1999, Working paper. 14 pp.

Simard, Y., McQuinn, I., Montminy, M., Lang, C., Miller, D., Stevens, C., Wiggins, D., and Marchalot, C. 1997. Description of the ***HAC*** standard format for raw and edited hydroacoustic data, version 1.0. Can. Tech. Rep. Fish. Aquat. Sci. 2174: vii + 65 pp.

ANNEX 1: LIST OF PARTICIPANTS

Name	Organisation	Phone	Fax	Email
Lars Andersen	Simrad AS, Norway	47 33034462	47 33042987	Lars.nonboe.andersen@simrad.no
Laurent Berger	IFREMER, Brest, France	33 298 224986	33 298224650	Laurent.berger@ifremer.fr
Noel Diner	IFREMER, Brest, France	33 298224177	33 298224650	Noel.diner@ifremer.fr
Ian Higginbottom	SonarData, Australia	61 362315588	61 362341822	Dave@sonardata.com
Ian McQuinn	DFO, IML, Canada	418 775 0627	418 775 0740	Mcquinni@dfo-mpo.gc.ca
Dave Reid	FRS, Aberdeen, UK	44 01224 295363	44 1224 295511	Reiddg@marlab.ac.uk

ANNEX 2: MODIFICATIONS TO EXISTING TUPLES AND PROPOSED NEW TUPLES

The following tables have been adopted as a standard component of this report and are designed outline any changes to existing tuples and define new provisional tuples. Where a tuple has been modified by the PG the changes are in **bold** and only those fields which have been changed are included, all other fields remain as previously described. The new tuples are provisional and will be reviewed by PGHAC in 2003 for acceptance into the standard. Software developers are reminded that these should not be shipped in any new software prior to this approval.

Table A.2.1. Definitions.

Data Type	Size	Range
DOUBLE	64 bit	Floating point
FLOAT	32 bit	Floating point
LONG	32 bit	Integer -2147483647 to 2147483647
ULONG	32 bit	Integer 0 to 4294967295
SHORT	16 bit	Integer -32767 to 32767
USHORT	16 bit	Integer 0 to 65535
CHAR	7 bit	Microsoft ASCII table for PC

Integer values are used to represent the encoded units presented in the tables.

Table A.2.2. Compressed 16-bit ping data tuple 10040.

In the ping tuple C-16, the sample value is encoded using a compressed 16 bit format.

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit range
0	Tuple size	4	ULONG	Tuple data size: variable	byte	[30 – 4 giga]
4	Tuple type	2	USHORT	Tuple type code: **10040**	unitless	10040
6	Time fraction	2	USHORT	Time of the transmitted pulse. Fraction of a second to add to the CPU ANSI C time (next field) to get a time precision of 0.0001 s.	0.0001 s	[0 – 6.5535 s] Practical range: [0 – 0.9999 s]
8	Time CPU ANSI C Standard time	4	ULONG	Time of the transmitted pulse. ANSI C time given by the CPU clock, in seconds. Usually the CPU clock is set to local time.	s	[0 – 4294967295 s] (= up to year 2106)
12	Software channel identifier	2	USHORT	Unique identifier for this software data channel to which the ping data is associated.	unitless	[0 – 65535]
14	Transceiver mode	2	USHORT	Operating mode of the transceiver: 0 = active: the transceiver is transmitting and receiving a monotone pulse 1 = passive: the transceiver is receiving but not transmitting 2 = test: a calibration signal is injected in the sounder 3 = eavesdropping: the transceiver is receiving while another transceiver is transmitting … 65535 = not available	unitless	[0 – 65535] Presently: [0, 1, 2, 3]
16	Ping number	4	ULONG	Ping sequence number since the beginning of the file. This should be a permanent label of the pings that should not be altered in further processing steps, namely the edition steps.	unitless	[0 – 4294967295]
20	Detected bottom range	4	LONG	Positive values indicate the range from the transducer face where the bottom detection criteria were encountered, under the above active transmitter mode. The sound speed field of the echosounder tuple is used for conversion of time to space. 2147483.647 = not detected Negative values are reserved for future use.	0.001 m	[-2147483.648 m to 2147483.647 m] Practical range: [0.000 – 15000.000 m]
24	No. of samples (>threshold) in this ping	4	ULONG	No. of samples (>threshold) in this ping (This information can also be computed from the tuple size).	unitless	[0 – 4294967295]

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit range
28	Sample value	2	SHORT	Sample value on 15 bit or zero series (< threshold) compressed into RLE samples (the upper bit is set to 1 and the lower 15 bits indicate the number of zeros + 1; 32768 below threshold values can then be compressed into one RLE sample; no value smaller than -16384 or larger than 16383 can be encoded).	depending on the “type of data sample” of the channel tuple: 0.001 volts or 0.01 dB, for S_v or TS	For volts: [-16.384 to 16.383 volts] Practical range: [0.000 to 15.000 volts]; For S_v and TS: [-163.84 to 163.83 dB] Practical range: [-150.000000 to 0.000000 dB]
...	continued					
...	Sample value	2	SHORT	Idem		
...	Space	2	USHORT	Space to allow the next field to be aligned on an address that is a multiple of 4.	unitless	0
...	Tuple attribute	4	LONG	Attribute of the tuple: 0 = original tuple, e.g., nothing special to mention 1 = edited tuple … Other attributes could be labeled by a code (e.g., tuple data quality). Negative codes should be used for special cases.	unitless	[-2147483648 to 2147483647]
	Backlink	4	ULONG	Tuple size: variable	byte	[... – 4 giga]

Table A.2.3a Change in the attribute filed of tuple 2001 to detect patch:
One more possible value is defined in the tuple attribute field.

Offset (byte)	**Field**	**Length (bytes)**	**Format**	**Content**	**Encoded units**	**Limit range**
108	Tuple attribute	4	LONG	Attribute of the tuple: 0 = original tuple, e.g., nothing special to mention 1 = edited tuple 2 = temporary 4 = patched Other attributes could be labelled by additional bits (e.g., tuple data quality). Negative codes should be used for special cases.	unitless	[-2147483648 to 2147483647]

Table A.2.3b Patch for the channel tuple for the Simrad model EK-500:
This tuple patches tuple 2000 or 2001 in order to get both transducer gains for EK500 for one transducer without defining two tuple channels.

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit range
0	Tuple size	4	ULONG	Tuple data size: variable	byte	[... – 4 giga]
4	Tuple type	2	USHORT	Tuple type code: 2002.	unitless	2002
6	Software channel identifier	2	USHORT	Unique identifier for this software data channel This identifier must be unique for the whole file in order to associate the pings to their proper parent channel N.B. This is not the hardware channel number.	unitless	[0 – 65535]
8	Echosounder document identifier	4	ULONG	Identification number for the parent echosounder document (i.e., the group of channels) to which this data channel belongs. It is the echosounder document identifier field of the echosounder tuple.	unitless	[0 – 4294967295]
12	S_v transducer gain	2	USHORT	Peak transducer gain used during computation of the S_v data sample.	0.01 dB	[0.00 dB – 655.35 dB] EK500 range: [0.00 to 99.90 dB]
14	TS transducer gain	2	USHORT	Peak transducer gain used during computation of the TS data sample.	0.01 dB	[0.00 dB – 655.35 dB] EK500 range: [0.00 to 99.90 dB]
16	Remarks	X4	CHAR	Character string comment. The string must be space filled to the 4 byte boundary. The Remarks field can be missing if there are no comments.	ASCII char.	variable
...	Tuple attribute	4	LONG	Attribute of the tuple: 0 = original tuple, e.g., nothing special to mention 1 = edited tuple 2 = temporary Other attributes could be labelled by a code (e.g., tuple data quality). Negative codes should be used for special cases.	unitless	[-2147483648 to 2147483647]
...	Backlink	4	ULONG	Tuple size: variable	byte	[... – 4 giga]

Table A.2.4a. New platform attitude parameter tuple - 41.

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit ange
0	Tuple size	4	ULONG	Tuple data size: 54 bytes	byte	54
4	Tuple type	2	USHORT	Tuple type code: **41**	unitless	41
6	Time fraction	2	USHORT	Fraction of a second to add to the following CPU ANSI C time to get a time precision of 0.0001 s (Local time at which the platform attitude reading was taken).	0.0001 s	[0 – 6.5535 s] Practical range: [0 – 0.9999 s]
8	Time CPU ANSI C Standard time	4	ULONG	Local time at which the platform attitude reading was taken. ANSI C time given by the CPU clock, in seconds. Usually the CPU clock is set to local time.	s	[0 – 4294967295 s] (up to year 2106)
12	Dependent attitude sensor identifier	2	USHORT	Unique identifier of the dependent attitude sensor	unitless	[0 – 65535]
14	Transceiver channel number	2	USHORT	Transceiver channel number from the parent channel tuple to which these parameters refer.	unitless	[0 – 65535]
16	Platform type	2	USHORT	The platform type in which the attitude sensor is installed: 0 = ship 1 = towed body 1 2 = towed body 2 3 = AUV 4 = ROV 5 = pelagic trawl 6 = bottom trawl …	unitless	[0 – 65535] Presently: [0; 1; 2; 3; 4; 5; 6]
18	Alongship offset	2	SHORT	Distance between the transducer and the reference point of the attitude sensor in the fore and aft direction. Negative values are on the aft side of the reference point of the attitude sensor. -327.68 = not available	0.01 m	[-327.68 to 327.67 m]
20	Athwartship offset	2	SHORT	Distance between the transducer and the reference point of the attitude sensor in the starboard and port direction. Negative values are on the port side of the reference point of the attitude sensor. -327.68 = not available	0.01 m	[-327.68 to 327.67 m]

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit ange
22	Elevation offset	2	SHORT	Distance between the transducer and the reference point of the attitude sensor in the vertical direction. Negative values are below the reference point of the attitude sensor. -327.68 = not available	0.01 m	[-327.68 to 327.67 m]
24	Remarks	30	CHAR	Character string comment, up to 30 characters. This field could be used to store the sensor type and serial number.	ASCII char.	32 characters
54	Space	2	USHORT	Space to allow the next field to be aligned on an address that is a multiple of 4.	unitless	0
56	Tuple attribute	4	LONG	Attribute of the tuple: 0 = original tuple, e.g., nothing special to mention 1 = edited tuple …	unitless	[-2147483648 to 2147483647]
60	Backlink	4	ULONG	Tuple size: 64 bytes	byte	64

Table A.2.4b. New attitude sensor tuple – 10140.

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit ange
0	Tuple size	4	ULONG	Tuple data size: 22 bytes	byte	22
4	Tuple type	2	USHORT	Tuple type code: **10140**	unitless	10140
6	Time fraction	2	USHORT	Fraction of a second to add to the following CPU ANSI C time to get a time precision of 0.0001 s (Local time at which the platform attitude reading was taken).	0.0001 s	[0 – 6.5535 s] Practical range: [0 – 0.9999 s]
8	Time CPU ANSI C Standard time	4	ULONG	Local time at which the platform attitude reading was taken. ANSI C time given by the CPU clock, in seconds. Usually the CPU clock is set to local time.	s	[0 – 4294967295 s] (= up to year 2106)
12	Attitude sensor identifier	2	USHORT	**Unique identifier for the attitude sensor providing the attitude information**	unitless	[0 – 65535]
14	Pitch	2	SHORT	Inclination of the platform relative to the horizontal plane in the fore-and-aft direction. Negative angles are below the horizontal and positive above. 3276.7 = unavailable	0.1 degree	[-3276.8 to 3276.7 degree] Practical range: [-90.0 to 90.0 degree]
16	Roll	2	SHORT	Inclination of the platform relative to the horizontal plane in the starboard-and-port direction. Negative angles are below the horizontal and positive above. 3276.7 = unavailable	0.1 degree	[-3276.8 to 3276.7 degree] Practical range: [-90.0 to 90.0 degree]
18	Heave	2	SHORT	Heave of the platform Positive heave is upwards and negative heave is downwards. 3276.7 = unavailable	0.01 m	[-327.68 to 327.67 m]
20	Yaw	2	SHORT	Yaw of the platform. Negative yaw angles are clockwise (to starboard) and positive angles are counter-clockwise (to port). 3276.7 = unavailable	0.1 degree	[-3276.8 to 3276.7 degree] Practical range: [-180.0 to 180.0 degree]
24	Tuple attribute	4	LONG	Attribute of the tuple: 0 = original tuple, e.g., nothing special to mention 1 = edited tuple … Negative codes should be used for special cases.	unitless	[-2147483648 to 2147483647]
28	Backlink	4	ULONG	Tuple size: 32 bytes	byte	32

Table A.2.5a. Simrad EK500 split-beam detected single target parameters sub-channel tuple (tuple code 4000).

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit ange
0	Tuple size	4	ULONG	Tuple data size: 54 bytes	byte	54
4	Tuple type	2	USHORT	Tuple type code: **4000**	unitless	4000
6	Time fraction	2	USHORT	Fraction of a second to add to the following CPU ANSI C time for a time precision of 0.0001 s (Local time at which the single-target TS logging was initiated).	0.0001 s	[0 – 6.5535 s] Practical range: [0 – 0.9999 s]
8	Time CPU ANSI C Standard time	4	ULONG	Local time at which the single-target TS logging was initiated. ANSI C time given by the CPU clock, in seconds. Usually the CPU clock is set to local time.	1 s	[0 – 4294967295 s] (= up to year 2106)
12	Parent software channel identifier	2	USHORT	Unique EK500 channel (tuple no. 2000 and 2001) identifier to which this TS parameter information applies. **N.B.** This is not the hardware channel number.	unitless	[0 – 65535]
14	Detected single target parameters sub-channel identifier	2	USHORT	Unique identifier for this software sub-channel tuple. **N.B.** This is not the hardware channel number.	unitless	[0 – 65535]
16	Minimum value	2	SHORT	Threshold value (see EK500 TS-Detection menu)	1 dB	[-327.68 – 327.68] Presently: [-100.00 – 0.00]
18	Minimum echo length	2	USHORT	Minimum normalized echo length (see EK500 TS-Detection menu)	0.01 steps	[0.00 – 655.35] Presently: [0.00 – 10.00]
20	Maximum echo length	2	USHORT	Maximum normalized echo length (see EK500 TS-Detection menu)	0.01 steps	[0.00 – 655.35] Presently: [0.00 – 10.00]
22	Maximum gain compensation	2	USHORT	Maximum one-way gain compensation (see EK500 TS-Detection menu)	0.01 dB	[0.00 – 655.35] Presently: [0.00 – 6.00]
24	Maximum phase compensation	2	USHORT	Maximum standard phase deviation (see EK500 TS-Detection menu)	0.01 steps	[0.00 – 655.35] Presently: [0.00 – 10.00]
26	Remark	30	CHAR	Character string comment up to 30 characters.	ASCII char.	30 characters

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit ange
56	Tuple attribute	4	LONG	Attribute of the Tuple 0 = original 1 = edited …	unitless	[-2147483648 – 2147483648]
60	Backlink	4	ULONG	Tuple size: 64 bytes.	byte	64

Table A.2.5b. Split-beam detected single-target tuple (tuple code 10090).

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit ange
0	Tuple size	4	ULONG	Tuple data size: variable	byte	[58 – 4294967295]
4	Tuple type	2	USHORT	Tuple type code: **10090**	unitless	10090
6	Time fraction	2	USHORT	Fraction of a second to add to the following CPU ANSI C time for a time precision of 0.0001 s (Local time at which the single target detection was made). This should correspond to the raw ping tuple time fraction.	0.0001 s	[0 – 6.5535 s] Practical range: [0 – 0.9999 s]
8	Time CPU ANSI C Standard time	4	ULONG	Local time at which the single-target detection was made. ANSI C time given by the CPU clock, in seconds. Usually the CPU clock is set to local time. This should correspond to the raw ping tuple time.	1 s	[0 – 4294967295 s] (= up to year 2106)
12	Parent sub-channel identifier	2	USHORT	Split-beam detected single-target parameter sub-channel identifier to which this TS information applies.	unitless	[0 – 65535]
14	Space	2	USHORT	Space to allow the next field to be aligned on an address that is a multiple of 4.	unitless	0
16	Ping number	4	ULONG	Ping sequence number since the beginning of logging. This should be a permanent label, corresponding to the raw ping tuple, and should not be altered in subsequent processing.	unitless	[0 – 4294967295]
20	Search start range	4	ULONG	Range at which search for single targets started for this ping	0.0001 m	[0 – 429496.7295]
24	Search end range	4	ULONG	Range at which search for single targets ended for this ping	0.0001 m	[0 – 429496.7295]
28	Detected bottom range	4	LONG	Positive values indicate the range from the transducer face where the bottom detection criteria were encountered, under the above active transmitter mode. The sound speed field of the echosounder tuple is used for conversion of time to space. 214748.3647 = bottom not detected.	0.0001 m	[-214748.3648 to 214748.3647 m]
32	Number of detected single targets	4	ULONG	Number of single targets detected in this ping.	unitless	[1 – 4294967295]
36	Range (target #1)	4	LONG	Range of the first detected single target	0.0001 m	[-214748.3648 – 214748.3648]
40	Compensated TS (target #1)	2	SHORT	Target strength of detected single target after compensation for off-axis angle.	0.01 dB	[-327.68 to 327.67 dB] EK500 range: [-100.00 to 0.00 dB]
42	Uncompensated TS (target #1)	2	SHORT	Raw target strength of detected single target uncompensated for off-axis angle.	0.01 dB	[-327.68 to 327.67 dB] EK500 range: [-100.00 to 0.00 dB]
44	Alongship angle (target #1)	2	SHORT	Fore-and-aft off-axis angle of the detected single target.	0.01 deg.	[-327.68 to 327.67 dB]

Offset (byte)	Field	Length (bytes)	Format	Content	Encoded units	Limit ange
46	Athwartship angle (target #1)	2	SHORT	Athwartship off-axis angle of the detected single target.	0.01 deg.	[-327.68 to 327.67 dB]
48	Range (target #2)	4	LONG	Range of the second detected single target	0.0001 m	[-214748.3648 – 214748.3648]
52	Compensated TS (target #2)	2	SHORT	Target strength of detected single target after compensation for off-axis angle.	0.01 dB	[-327.68 to 327.67 dB] EK500 range: [-100.00 to 0.00 dB]
54	Uncompensated TS (target #2)	2	SHORT	Raw target strength of detected single target uncompensated for off-axis angle.	0.01 dB	[-327.68 to 327.67 dB] EK500 range: [-100.00 to 0.00 dB]
56	Alongship angle (target #2)	2	SHORT	Fore-and-aft off-axis angle of the detected single target.	0.01 deg.	[-327.68 to 327.67 dB]
58	Athwartship angle (target #2)	2	SHORT	Athwartship off-axis angle of the detected single target.	0.01 deg.	[-327.68 to 327.67 dB]
…	Continued					
…	Tuple attribute	4	LONG	Attribute of the Tuple 0 = original 1 = edited	unitless	[-2147483648 – 2147483648]
…	Backlink	4	ULONG	Tuple size: variable (multiple of 4 bytes).	byte	[68 – 4294967295]

Table A.2.6. Specification of attribute field to include patch tuples.

Tuple attribute (ULONG)																																	
Content	Undefined																												Patched	Temporary	Edited	Value
Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	Code
Original	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Edited	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Temporary	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
Temporary + Edited	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3
Patched	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4
Patched + Edited	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5
…																																	…
b0 (1)	Tuple has been edited (0 = original tuple)																																
b1 (2)	Temporary tuple (unofficial).																																
b2 (4)	Tuple has been patched. Therefore another tuple must refer to this tuple																																
b3 (8)	Undefined																																
…	…																																
b31 (2147483648)	Undefined																																