SEG-D, Rev 3.0

SEG Field Tape Standards

November, 2009

Table of Contents

1.0 Introduction

For several years now there has been talk of the need for a new revision of the SEG-D standard for seismic field data. The last major revision of the standard, Rev 2.0, was published in 1996 with an incremental update Rev 2.1 in 2006. This new Revision 3.0 recognizes the significant developments in acquisition and computer technologies and brings the standard into line with current, and many envisioned future, industry techniques and practices. It also resolves longstanding ambiguities and corrects both typographic and factual errors that have been reported against the existing SEG-D standards. (It does not, we admit, contain the prescription for RODE encapsulation promised in the Rev 2.0 introduction. It has been decided to keep the mapping of seismic data to RODE separate from the SEG-D format.) While not fully 100% backwards compatible to prior revisions, the only significant changes required to existing software that creates SEG-D output is the generation of a third General Header block, and updating the channelset descriptors to 96 bytes. Most other additions are optional.

Among the major upgrades that can be encoded in new header and trailer blocks are microsecond accurate timestamps, detailed source and sensor (multicomponent included) description, extended recording modes that allow for nearly 9 years of continuous recording by permanent emplacements, electromagnetic survey support, post-acquisition edits, coordinate reference system datum and projection, microsecond sample rates, and negative start times.

At the SEG convention in Houston in 2005, the SEG Technical Standards Committee decided to revive the SEG-D Format Subcommittee, with Stewart A. Levin of Halliburton Energy Services, whose passions include maintenance and improvement of SEG-D input software packages, volunteering to chair the subcommittee and provide the necessary energy to drive it forward. Since then there has been considerable activity to ascertain what should be contained in the new revision, with email discussion lists, subcommittee activities, progress reports in The Leading Edge and First Break, and meetings of interested parties at the EAGE and SEG annual meetings. This updated standard explicitly incorporates items and codes from the industry standards groups such as Energistics and the International Association of Oil and Gas Producers (OGP).

The following individuals have been active participants in this effort:

Jill Lewis	Troika International	Technical Standards Chair
Stewart A. Levin	Halliburton	SEG-D Subcommittee Chair
Rune Hagelund	Roxar	Technical Standards Vice Chair
Barry D. Barrs	ExxonMobil	
Roger Lott	Consultant	
Tore Nilsen	Schlumberger WesternGeco	
Francois Daube	Schlumberger WesternGeco	
Paul Maton	Sillimanite Consultants	
Nils Aatland	Fugro Seismic Imaging	
Jacques Hamon	Sercel	
Angus Stott	Petroleum Geo-Services	
Peter Green	Saudi Aramco	
Joseph Cignoli	Saudi Aramco	
Bob Firth	Troika International	

Numerous other people have also participated at various times.

1.1 Controlling Organization

The SEG-D rev 3.0 is administered by the SEG Technical Standards Committee. Any questions, corrections or problems encountered in the format should be addressed to:

Society of Exploration Geophysicists
P.O. Box 702740
Tulsa, Ok 74170-2740

Attention: SEG Technical Standards Committee
Phone: (918) 497-5500
Fax: (918) 497-5557
Internet site: www.seg.org

2.0 Changes Introduced in Revision 3.0

Revision 3.0 introduces format changes to better handle the requirements from modern seismic acquisition and processing systems. Revision 3.0 attempts to be an improved format for information exchange between the data acquisition and processing system by including important metadata like equipment positions and trace edits.

The following list is a summary of the specific changes made in Revision 3.0 compared to 2.1:

1. Revision number changed from 2.1 to 3.0, see
    a. Chapter 4, field number 2, and
    b. Chapter 8.2 (General Header #2, byte 11 and 12).

2. The following headers are left mostly unchanged to allow e.g. pattern matching, i.e. searching for recognizable patterns in a stream of data. The minor changes are:
    a. Tape Label (updated the SEG-D version number),
    b. General Header #0 (allow FF_{16} as value of byte 23, to support different base scan intervals), and
    c. Demux Trace Header (allow $FFFF_{16}$ as value in bytes 5 and 6, to support extended trace number).

3. Rev 3.0 no longer requires a channel set to be present, making it possible to store a SEG-D record without any data. This supports creating a SEG-D record with only header data, e.g. only External, Extended, and/or General Trailer, which is useful for transferring metadata between systems.

4. Rev. 3.0 allows traces of zero length (no samples). This is done by setting start and end time to the same value, and setting number of samples to 0 (Byte 13–16 of Channel Set Descriptor). This supports transfer of Trace Header meta data between systems without any data attached.

5. Rev 3.0 is still a shot domain format; however a separate mode, Extended Recording Mode, has been added to partially support non-shot domain data.

6. A high-resolution timestamp is introduced to accurately determine the time in SEG-D. The time of first sample in record is entered in General Header #3 (Bytes 1–8). Position measurements, orientation measurements, source events, etc. all have a timestamp attached, which allows multiple measurements of the same type to be stored within the same record and trace header. This give the manufacturer a much more fine grained control over, for example, the movement of equipment during the record, and allows the use of modern, more powerful filtering techniques to be applied to the data traces.

7. A SEG-D positioning format has been defined, and Position blocks may be inserted into Trace Headers. The datum and projection information is inserted as part of the General Header. Only one datum/projection is supported in a SEG-D record, however multiple datums/projections may exist for a given storage unit (tape).

8. Rev 3.0 has an extended indexing structure, allowing more advanced logical addressing of traces, and better grouping of information. The following indexes are supported for sources and receivers: Line, Point, Point Index, Depth, Group and Re-shoot index.

9. An orientation header has been defined to properly support multi-component data. The format supports rotation to a global reference system as part of acquisition, or to be applied later in processing. The rotation specified in the orientation header may or may not be applied as part of acquisition. The Trace type and Line, Point and Group indexes are used to determine which traces should be rotated together.

10. Rev 3.0 supports an increased range of sampling intervals, record lengths, number of channels, filter resolution, etc. compared to Rev 2.x, to be able to handle the requirements of modern acquisition systems. In addition negative times for start of trace are now supported.

11. The size of header, data, record etc. are now explicitly stated in General Header to facilitate quick data access and facilitate error recovery.

12. All header blocks have an enumerated type attached (byte 32 of all blocks), both to increase flexibility of the format to and simplify decoding. This also allows all information to be optional, simplifies error recovery and increases the robustness of the format. In addition, it allows information to be inserted in the header and order deemed most useful by the recorder.

13. The Channel Set Header has been extended to 96 bytes to support extended sampling intervals, trace lengths, etc. All values are now explicitly stated, no complex calculation is required to determine, e.g., sampling interval or descale multiplier. In addition derived values, such as the number of samples per channel, are also explicitly stated to avoid ambiguities of prior versions of the SEG-D standard. A recorder-defined textual description has been added to the channel set descriptor to increase the flexibility of the storage format and increase the efficiency of information transfer between acquisition and processing.

14. The number of channels per channel set has been extended to 16,777,215.

15. General Header blocks for common acquisition information like Client, Job, Survey Area, Vessel/Crew and Line information have been defined. The information in these General Header blocks is intended to be a short, textual description of the specific item, not a complete, detailed description. To give a complete definition of the survey area, or store the complete client contract, the External or Extended header may be used. The General Header blocks #4–8 are basically defined to simplify data management and information identification.

16. The size of all recorder-defined headers (Extended, External and General Trailer) has been increased. The maximum size of an External/Extended header is now 512 MB and a General Trailer may contain up to 128 GB.

17. The maximum Trace Header size has been increased to 8180 bytes to allow multiple positions or other measurements to be inserted in each trace. The space may also be used by the recorder to store an increased amount of recorder-defined blocks.

18. The source information is far more comprehensive than in Rev 2.x. A different General Header block type exists to describe each source type (Vibrator, Airgun, etc.), replacing the General Header Block N of Rev 2.x. The information contained in the source description blocks basically aligns the information in SEG-D with SPS.

19. An additional source information block has been defined to allow specification of the actual firing time of the source (with microsecond accuracy), and the status of the source.

20. Compound sources can be created, i.e. sources containing other sources, allowing SEG-D to store information about, e.g., single airguns and the combined source or single vibrator trucks and the combined vibrator group.

21. Auxiliary trace reference blocks may be attached to the source information, listing which auxiliary traces contain relevant information for the specified source.

22. The source information may be inserted into the General Header or Trace Header depending on what is most useful. For records with multiple source events (like slip-sweep acquisition), storing the source information in the source related auxiliary traces (e.g. source reference signal), is recommended.

23. Allow recording data in 64 bit IEEE
    Additional Valid Format Code for bytes 3 & 4 of the General Header #1 is:

8080	64 bit IEEE float demultiplexed

24. To support acquisition systems where sensors are deployed for longer periods of time without external clock synchronization, a special Time drift header block have been added. This allows storing information about drifting clocks (for quality control and more advanced clock corrections post acquisition). It must be noted that even though SEGD supports logging of the information, SEGD does not support storing data from multiple drifting time reference systems. Traces must be clock drift corrected prior to record creation if data from multiple sensors are stored in one record.

25. Support for Electromagnetic surveys has been added, including Electromagnetic source description and Electromagnetic receiver description.

26. Sensor calibration information may now be added. SEGD supports an individual scaling factor, as part of the sensor sensitivity value, and a more complex calibration function in frequency or time domain. The function can either be specified as part of the trace header (frequency domain), or as a separate calibration "trace" (time domain).

27. A set of standard measurements may now be added to a special Measurement header block, which may be inserted into a trace header. Multiple measurement blocks may be used in any trace. Supported measurements are depth, temperature, pressure, wind speed, altitude, uphole time, etc. Note: The format of the Measurement block is still not finalized, discussions still ongoing with Energistics regarding the contents.

28. The General Trailer format has been completely changed from that of Rev 2.x. The new General Trailer format consists of a number of blocks of data, each a multiple of 32 bytes, and starting with a 32 byte description header. Any binary or ASCII block data may be stored unmodified as part of the General Trailer, as long as the block is padded with zeros (0x00 – for binary data) or spaces (0x20 – for ASCII data) until it is a multiple of 32 bytes long.

29. An edit format has been defined as part of the General Trailer, to simplify addition of post acquisition edit records (e.g. by quality control or processing systems), to standardize transfer of edit information between acquisition and processing. The format is based on the *SEG ADS Trace Edit format* (Norris et. al. 1999).

30. Storing positioning files like P1 and P2 as part of the General Trailer (for backup purposes), has been standardized.

31. Some other simple Trailer blocks, like Observer Log and Text Comments have also been defined.

32. Several examples have been added to Appendix E.

33. Revision 2.x Appendix D "Header Descriptors" has been removed.

34. Appendix F, which previously listed specific tape drives and corresponding maximum block sizes, has been removed. SEG-D revision 3.0 supports data on any fixed block, variable block and byte stream device including tapes, disks, DVDs, and network connections with record sizes up to any specific device, operating system, file system, or network limits. Information regarding specific devices, their block sizes and recommended usage can now be found on http://www.seg.org through the Technical Standards pages. This allows keeping the information up to date without modifying the standards document.

2.1 Changes Introduced in Revision 2.1

The following list discusses each of the specific changes made in Revision 2.1 compared to Revision 2.0.

1. Revision number changed from 2.0 to 2.1, see
   - Chapter 4, field number 2
   - Chapter 8.2 (General Header # 2, byte 11 and 12).
2. Since Rev 2.1 is intended to handle ultra high density tapes, acceptable media is expanded to include:

   STK 9940B, IBM 3592 (Jaguar-1) and IBM TS1120 (Jaguar-2).

   For further details, see Appendix F.
3. More than one production line per tape is allowed, as long as a unique combination of **field file number** and a new line **sequence number** are used per storage unit.

   The sequence number was added to General Header # 2, byte 21–22. Range is 1–65535 (Set to 0 if not valid).
4. Appendix A is updated (Manufacturers of Seismic Field Recorders).
5. Appendix C is updated (API Producer Organization Codes). Organization codes are now assigned by POSC which maintains the current list of codes (API in previous revision).
6. Producer organization code is no longer a required field.

2.2 Changes Introduced in Rev 2.0

The following list discusses each of the specific changes made in Revision 2.0 compared to Revision 1.0. Also mentioned are changes which were discussed as potential changes to be included in Rev 2.0, but were not included in Rev 2.0.

1. Since Rev 2.0 is intended to handle higher density tapes, acceptable media is expanded to include: 3490/3490E, 3590, D2, and D3.

2. It is not anticipated that the higher density drives will be used to record multiplexed data. Rev 2.0 does not support multiplexed data.

3. No specific changes will be made to SEG-D to handle "non-shot domain" data. Either a new committee should be formed, or the charter of this committee should be extended to develop a new format for this application. It does not appear practical to extend SEG-D to fit this application.

4. No special arrangements will be made to provide a standard method of recording SPS in the SEG-D header. The relevant portions of SPS can be put into existing header extensions in user defined positions.

5. The MP factor description will be modified to clarify the meaning for fixed bit data (see MP discussion in section 7).

6. The description of byte 12 in the General Header is being clarified to clearly state that the byte defines the number of additional blocks. Figure 4 in the SEG-D Rev 1 document will be changed from # BLKS IN GEN HDR to "# Additional blks in Gen Hdr". Another correction will be made to correctly state, for byte 1 of the General Header, "File number of four digits (0–9999) set to FFFF (Hex) when the file number is greater than 9999.

7. The RECEIVER LINE NUMBER (bytes 1–3) and RECEIVER POINT NUMBER (Bytes 4–6) in the Trace Header Extension have been modified to include a fractional component. An all one's pattern ($FFFFFF_{16}$) in either of these fields, will serve as a flag to indicate that the complete five byte value will be located in newly defined locations in the Trace Header Extension. See Trace Header Extension table below.

8. The maximum number of Trace Header Extensions is now limited to 15.

9. Channels within the same Channel Set must now have the same number of Trace Header Extensions. Since all traces within a Channel Set will now contain the same number of Trace Header Extensions, the number of Trace Header Extensions will be indicated in the Channel Set Descriptor. The previously unused nibble of Byte 29 in the Channel Set Descriptor will now be defined to be a 4 bit binary parameter that defines the number of Trace Header Extensions for that Channel Set. Byte 29 of the Channel Set descriptor will now be:

0	1	2	3	4	5	6	7
EFH_3	EFH_2	EFH_1	EFH_0	THE_3	THE_2	THE_1	THE_0

   As a result of this limitation the Trace Header Extension field in Byte 10 of the Trace Header will also be redefined as a 4 bit value limited to a maximum of 15 Trace Header Extensions.

10. The length of each trace within a Channel Set is now restricted to be the same value. This limitation and the restricting the number of Trace Header Extensions to the same number within a Channel Set will result in each trace within a Channel Set being recorded with the same number of bytes.

11. A tape label will be required on each tape. The details of this label format are described in section 4.

12. Data may be recorded in large logical blocks to maximize the transfer rates with high density tape systems. 3 types of device structures are supported:

A) Variable block length devices.

Every shot record must be aligned on a block boundary (i.e. each block will contain data from only one shot record). Multiple channel sets may be included in each block. When the data to be recorded in a block contains less than the maximum number of bytes in the block, there will be no padding characters to fill the block.

Storage Unit Structure in field 3 in Storage Unit Label must contain the text "RECORD"

B) Byte stream devices

There is no concept of a block, even though there is a hidden underlying physical block structure. Within each file, one or more shot records are written consecutively without any gap.

Storage Unit Structure in field 3 in Storage Unit Label must contain the text "RECORD"

C) Fixed block length devices

Every shot record must be aligned on a block boundary (i.e. each block will contain data from only one shot record). Multiple channel sets may be included in each block. Typically the last block in a shot record will contain less data than the block size, the remaining part of this block will be padded with characters without any information.

Storage Unit Structure in field 3 in Storage Unit Label must contain the text "FIXREC" and the block size is found in field 5 in Storage Unit Label.

Note: Structure A can be mapped to a file directly but one can not re-generate the same interblock gaps and File Marks from data stored on a file. Structure B and C can be mapped to a file directly and the structure can be re-generated apart from the original position of the File Marks.

13. An appendix will be added to indicate the maximum allowable block size for accepted types of media. It is expected that this table will need to be updated approximately once per year.

14. Byte 12 of the Trace Header will have an additional option, TR= 03 Trace has been edited. This parameter will indicate the acquisition system has modified one or more samples of this trace. During data acquisition, if a telemetry error occurs, a sample may be corrupted. Some radio acquisition systems fill in this missing data with a copy of the previous sample, or interpolate to fill in the missing sample. Trace edit can also occur when a noise edit process is applied by the acquisition system. The TR=03 flag should be set for those traces which have been modified by the acquisition system.

15. The SEG-D, Rev 2.0 format treats data going to tape as a byte stream. File Marks are not required to separate shot records, however File Marks may be included in between shot records where appropriate to ease error recovery and/or to provide logical partitioning of the data. If used, File Marks may only be recorded at shot record boundaries. For field tapes, File Marks should be written as frequently as possible, preferably for every shot. If data is staged on disk, many shots can be stored in each file. When SEG-D, Rev 2.0 data is recorded on tape, an EOD mark must be recorded after the last valid record and prior to the end of tape

16. The time standard referenced by byte 14 of the General Header has been changed from GMT to UTC.

17. Partitioning of a tape or other type media volume is now allowed. Each partition, or each tape if not partitioned, constitutes one storage unit. The storage unit label shall consist of the first 128 bytes of the first user-writable tape record in the first user-writable physical block and may, optionally, be followed by a File Mark. No File Mark shall be written before the storage unit label.

18. Added a field in the Trace Header extension to indicate the type of sensor used for that trace (Byte 21).

2.3 Changes Introduced in Rev 1

In 1994, several changes were introduced to SEG-D to increase flexibility. These changes are listed below.

1. To allow for additional defined fields in SEG-D headers, additional blocks are allowed for the General Header and Demux Trace Header.

2. Added provision for an optional set of General Trailer blocks. This type header allows provisions for recording auxiliary seismic system and real-time navigation related data in the trailer. The trailer is optional and typically follows all other recorded data.

   The addition of the trailer will allow the accumulation of system faults, data QC information, real-time navigation position, and timing information on the same tape, and contiguous with, the shotpoint that it relates to. By recording this data after all of the other data, additional time is provided for collecting the data and transferring it to the recording system.

   The Trailer blocks take the same general form as the Channel Set Descriptor. Byte 11 uses the "Channel Type Identification" set to 1100 to indicate a Trailer block. Bytes 1 and 2 indicate the number of the General Trailer block, with the first block numbered as 1.

   All other information in the trailer is optional and may be formatted as desired by the manufacturer/user.

   The number of General Trailer blocks is indicated in bytes 13 and 14 of General Header Block #2.

3. Provide provision to include the revision of SEG-D format. Added to Bytes 11 and 12 of General Header Block #2 contain the SEG-D Revision Number. The revision number is a 16 bit unsigned binary number. The Revision number is 1 for the proposed version.

   In addition, in the General Header Block #1, nibble 1 of byte 12 contains the number of additional blocks in the general header. Nibble 1, byte 12 is an unsigned binary number. This number will be 1 or greater for SEG D Rev 1.

4. Added provision to include the source and receiver locations for each source and receiver location. Source locations are included in the General Header Blocks. Block #3 contains the position for Source Set #1. Additional General Header Blocks may be included to allow for additional Source Sets.

   Source positions are defined by a Source Line Number (three bytes integer and two bytes fraction), a Source Point Number (three bytes integer and two bytes fraction), and a Source Point Index (one byte). This index allows several locations for the source in the grid, the original value is 1 and that value is incremented by 1 every time the source is moved, even when it is moved back to a previous location).

   Receiver locations are included in Trace Header Extensions to be used with Demux Trace Headers. Receiver positions are defined by a Receiver Line Number (three integer bytes and two fraction bytes), a Receiver Point Number (three bytes integer and two bytes fraction), and a Receiver Point Index (one byte). This index allows for defining the receiver group in the grid, the original value is 1 and that value is incremented by 1 every time the receiver is moved, even when it is moved back to the previous location.

5. Provide for the use of File Numbers greater than 9999. Bytes 1, 2, and 3 in General Header Block #2 allow for a three byte, binary file number. When the file number is greater than 9999, bytes 1 and 2 in the General Header Block #1 must be set to FFFF.

6. Provide for Extended Channel Sets/Scan Types. General Header Block #2 allows for a two byte, binary number of Channel Sets/Scan Types in bytes 4 and 5. When using the Extended Channel Sets/Scan Types, byte 29 of General Header #1 must be set to FF.

7. Provide for additional Extended and External Header blocks. General Header Block #2 bytes 6 and 7 (for Extended Header blocks) and Bytes 8 and 9 (for External Header blocks) allow the use of a two byte, binary number to allow more than 99 blocks. When using these capabilities, General Header Block #1 byte 31 (for extended) and byte 32 (for external) must be set to FF.

8. Provide a mechanism for recording additional information about vibrator sources. Byte 15 of the General Header Block #N indicates the signal used to control vibrator phase. Byte 16 indicates the type of vibrator (P, Shear, Marine). Bytes 28 and 29 contain the phase angle between the pilot and the phase feedback signal.

   The additional vibrator information may be recorded for multiple sets of sources by using additional General Header blocks.

9. Provide for larger number of samples per trace. Using bytes 8, 9, and 10 of the Trace Header Extension.

10. Provide provisions for using 1/2" square tape cartridges. (ANSI X3.180 1989).

11. Allow recording data in IEEE and other new formats.
    Additional Valid Format Codes for bytes 3 & 4 of the General Header are:

0036	24 bit 2's compliment integer multiplexed
0038	32 bit 2's compliment integer multiplexed
0058	32 bit IEEE multiplexed
8036	24 bit 2's compliment integer demultiplexed
8038	32 bit 2's compliment integer demultiplexed
8058	32 bit IEEE demultiplexed

The IEEE format is fully documented in the IEEE standard,
"ANSI/IEEE Std 754 - l985", available from the IEEE.

The IEEE format is summarized as follows:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_7	C_6	C_5	C_4	C_3	C_2	C_1
Byte 2	C_0	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}	Q_{-6}	Q_{-7}
Byte 3	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}	Q_{-14}	Q_{-15}
Byte 4	Q_{-16}	Q_{-17}	Q_{-18}	Q_{-19}	Q_{-20}	Q_{-21}	Q_{-22}	Q_{-23} (see Note 1)

The value (v) of a floating-point number represented in this format is determined as follows:

if $e = 255$ & $f \neq 0$. .$v = \text{NaN}$	Not-a-Number (see Note 2)
if $e = 255$ & $f = 0$. .$v = (-1)^s \times \infty$	Overflow
if $0 < e < 255$. . . . .$v = (-1)^s \times 2^{e-127} \times (1.f)$	Normalized
if $e = 0$ & $f \neq 0$ . . . $v = (-1)^s \times 2^{e-126} \times (0.f)$	Denormalized
if $e = 0$ & $f = 0$ . . . $v = (-1)^s \times 0$	± zero

where e = binary value of all C's (exponent)
f = binary value of all Q's (fraction)

NOTES: 1. Bit 7 of byte 4 must be zero to guarantee uniqueness of the start of scan in the Multiplexed format (0058). It may be non zero in the demultiplexed format (8058).

2. A Not-a-Number (NaN) is interpreted as an invalid number. All other numbers are valid and interpreted as described above.

12. Allow for the use of blocked records. Allow blocked demultiplexed data (integral number of traces in a block). Headers will not be blocked. All records in a block will be the same size. Not all blocks will be the same size. Byte 20 in the general header (B1 = 1) will indicate blocked data. Blocks will be limited to 128 kilobytes. All traces in a block are in the same Channel Set.

13. Added the effective stack order (unsigned binary), in byte 30 in the Channel Set descriptor. Set to 0 if the trace data was intentionally set to real 0. Set to 1 if no stack. Set to the effective stack order if the data is the result of stacked data (with or without processing).

14. Improved definition of undefined fields. All undefined fields will be specified as: "This field is undefined by this format".

15. Added provisions for a Trace Edit byte (byte 10 of Demux Trace Header) to indicate traces zeroed for roll-on or roll-off and to indicate deliberately zeroed traces.

TR=0 No edit of this trace
TR=1 Trace part of dead channels for roll-on or roll-off spread; trace intentionally zeroed.
TR=2 Trace intentionally zeroed.

16. Increased precision of MP factor, using byte 7 of the Channel Set descriptor.

17. Since modern seismic vessels record more than one streamer at a time, a standard convention is required to identify which streamer recorded each channel of data. The Channel Set Descriptors are updated to handle this task. The definition of a channel set is expanded to include the following rules. A channel set is a group of channels that:

a) Use identical recording parameters. This includes the same record length and sampling interval.
b) Use identical processing parameters, including the same filter selection and array forming parameters. A field has been added to Channel Set Descriptor byte 32 to describe any array forming applied to data in that channel set.
c) Originates from the same streamer cable for marine data. The streamer cable number for each channel set has been added to Channel Set Descriptor byte 31.
d) Consists of channels with the same group spacing. For example, if one steamer has short group spacing close to the boat and longer groups spacing at long offsets, the data from that streamer would be recorded as two channel sets.

In addition, the first channel in each channel set will start with Trace number one.

18. Correct the MP factor calculation (refer to Appendix E7 in the SEG-D recording format description.)

MP CALCULATION
The calculation of **MP** for a data recording method is given by one of the following equations:
(1) **MP** = FS − PA − Cmax; for binary exponents,
(2) **MP** = FS − PA − 2 x Cmax; for quaternary exponents,
(3) **MP** = FS − PA − 4 x Cmax; for hexadecimal exponents (except the 4 byte excess 64 method),
(4) **MP** = FS − PA − 4 x (Cmax − 64); for excess 64 hexadecimal exponents,
(5) **MP** = FS − PA − (Cmax − 127); for 32 bit IEEE exponents,

where
2^{FS} = Converter full scale (millivolts),

2^{PA} = Minimum system gain,

and

Cmax = maximum value of the data exponent,
Cmax = 15 for binary exponents,
7 for quaternary exponents,
3 for hexadecimal exponents except excess 64,
127 for excess 64 exponents, and
255 for 32 bit IEEE exponents.

19. Added the option for using record lengths in millisecond increments (rather than the previous 0.5 second increments). The Extended Record Length is the record length, in unsigned binary milliseconds, and is recorded in bytes 15–17 in General Header Block #2. If this option is used, Record Length (R), in the General Header Block #1, bytes 26, 27 must be set to FFF.

3.0 Format Overview

A SEG-D record consists of three different parts stored consecutively on a storage media.

Record Header (Mandatory)	Trace Data (Optional)	Record Trailer (Optional)

Start or record

End or record

On blocked devices like tapes, each of these parts must start on a block boundary.

The SEG-D, Rev 3.0 format treats data going to physical storage devices as a byte stream which may be conveniently divided, if needed, into records and blocks.[1] Figure 1 illustrates a typical record structure.

Each SEG-D Rev 3.0 dataset must begin with a storage unit label, as detailed in section 4. Following the label, each SEG-D record is recorded in demultiplexed format. SEG-D, Rev 3.0 does not support multiplexed data records.

A tape or other media to be used for SEG-D, Rev 3 recording may be partitioned. Each partition, or each tape if not partitioned, constitutes one storage unit. A disk file is a partition of a disk, and hence constitutes a storage unit. If SEG-D data are stored on a raw disk device (no filesystem, i.e. a byte stream device), the entire device may be considered a storage unit. Transferring SEG-D data across networks is allowed. Each network port (e.g. TCP or UDP socket) is then considered a storage unit. A network port may be used to transfer an unlimited sequence of SEG-D field records. However each time the transfer is closed and (re)opened, the data must be prepended with a storage unit label. Using other types of media for storing or transferring SEG-D data is allowed. Some examples are USB memory sticks, flash devices, solid state disks, DVDs, serial ports, raw ethernet transfer. The same rule applies here: if the device is partitioned, e.g. placing each field file into a separate disk file, each partition is considered a storage unit. However if SEG-D data are read/written to the entire device as one unit, the entire device is considered a storage unit.

The storage unit label (tape label) shall consist of the first 128 bytes of the first user-writable tape record in the first user-writable physical block and may, optionally, be followed by a File Mark. No File Mark shall be written before the storage unit label. For field tapes we recommend the storage unit label be followed by a File Mark.

When blocked data are being recorded, all of the headers may be included in the same block with the initial channel set. If the header spans multiple blocks, the remaining part of the header may be stored in the same block as the initial channel set. Each channel set may be split across block boundaries. A trace may be split across several blocks, and a trace does not have to start on at a block boundary. The trace header may also be split across blocks.

Data may be recorded in large blocks to maximize the transfer rates with high density tape systems. Three types of device structures are supported:

A) Variable block length devices.

Every shot record must be aligned on a block boundary (i.e. each block will contain data from only one shot record). Multiple channel sets may be included in each block. Blocks should not be padded to make their length up to the maximum block size specified in the Storage Unit Label.

Storage Unit Structure in field 3 in Storage Unit Label must contain the text “RECORD”

[1] Each new field file must begin at a record boundary.

B) Byte stream devices

There is no concept of a block, even though there is a hidden underlying physical block structure. Within each file, one or more shot records are written consecutively without any gap.

Storage Unit Structure in field 3 in Storage Unit Label must contain the text “RECORD”, and Maximum Block Size should be set to 1.

C) Fixed block length devices

Every shot record must be aligned on a block boundary (i.e. each block will contain data from only one shot record). Multiple channel sets may be included in each block. Typically the last block in a shot record will contain less data than the block size, the remaining part of this block will be padded with characters without any information.

Storage Unit Structure in Field 3 in Storage Unit Label must contain the text “FIXREC” and the block size is found in Field 5 in Storage Unit Label.

Note: Structure A can be mapped to a file directly but one can not re-generate the same inter-block gaps (if present) and File Marks from data stored on a file. Structure B and C can be mapped to a file directly and the structure can be re-generated apart from the original position of the File Marks.

The SEG-D, Rev 3.0 format treats data going to tape as a byte stream. File Marks are not required to separate shot records, however File Marks may be included between shot records where appropriate to ease error recovery and/or to provide logical partitioning of the data. If used, File Marks may only be recorded at shot record boundaries. For field tapes, File Marks should be written as frequently as possible, preferably for every shot. If data are staged on disk, many shots can be stored in each file. When SEG-D, Rev 3.0 data are recorded on tape, an EOD mark must be recorded after that last valid record and prior to the end of tape.

If the tape media supports multiple partitions, SEG-D data may be written to any of the partitions of the tape, each beginning with a Storage Unit Label. Data from one partition cannot “run-over” into a subsequent partition, each partition must be capable of being decoded in isolation.

On one tape, it is allowable to mix partitions containing SEG-D data with partitions containing non SEG-D formatted information.

The headers of SEG-D Rev 3.0 can be very large compared to previous versions of the format. The maximum for each of the main header types are shown below:

- General Header: 2,097,152 bytes= 2 MB
- Skew headers: 2,097,120 bytes ~ 2 MB
- Scan type/Channel Set descriptors: 6,291,360 bytes ~ 6 MB
- Extended Header: 536,870,912 bytes ~ 512 MB
- External Header: 536,870,912 bytes ~ 512 MB
- General Trailer: 137,438,953,472 bytes ~ 128 GB
- Trace Header: 8,180 bytes ~ 8 kB

In addition, the maximum trace size is

- Trace: 8,120 (header) + 34,359,738,360 (data) bytes ~ 32 GB

The headers and traces may therefore span multiple tape blocks.

SEGD Rev 3.0 disconnects tape blocks from data blocks completely, the start of a header block/start of trace may be anywhere in a tape block. A header may also be split between two tape blocks, e.g. a trace header may have just one byte in one tape block, and the rest continuing in the next block.

The maximum block size used on the storage media is listed in the Tape Label, bytes 20-29. The maximum block size is limited to about 4 GB in SEG-D Rev 3.0. If the storage device supports less than 4GB blocks, the maximum block size is determined by the manufacturer of the device. The following values should be used as the maximum block size in the media Tape Label:

- For fixed block size media, this will be the block size of the media.
- For variable block media, this will be the maximum block size used on the tape. The size is determined by the data recorder, and may be less than the maximum value determined by the device manufacturer. The size should be selected to allow efficient reading and writing data to/from the media.
- For byte stream media, e.g. tape devices like D2, disk, USB memory sticks, network transfer, etc., the maximum block size is always set to 1.

SEG-D data is streamed to/from the storage device as a stream of bytes, split into blocks as defined by the device manufacturer and the maximum block size in the Tape Label, with the following modifications:

- The Tape Label must be located at the beginning of the media, with no preceding data.
- It is strongly recommended to write an end-of-file mark after the Tape Label.
- The General Header Block #1 (start of record), the first Trace Header (start of data) and the first General Trailer block (start of trailer) will always be located at the beginning of a device block. For fixed block size media, this means the previous block may have to be padded with zeros, and for variable block size media, the previous block may be smaller than the maximum block size indicated in the Tape Label.

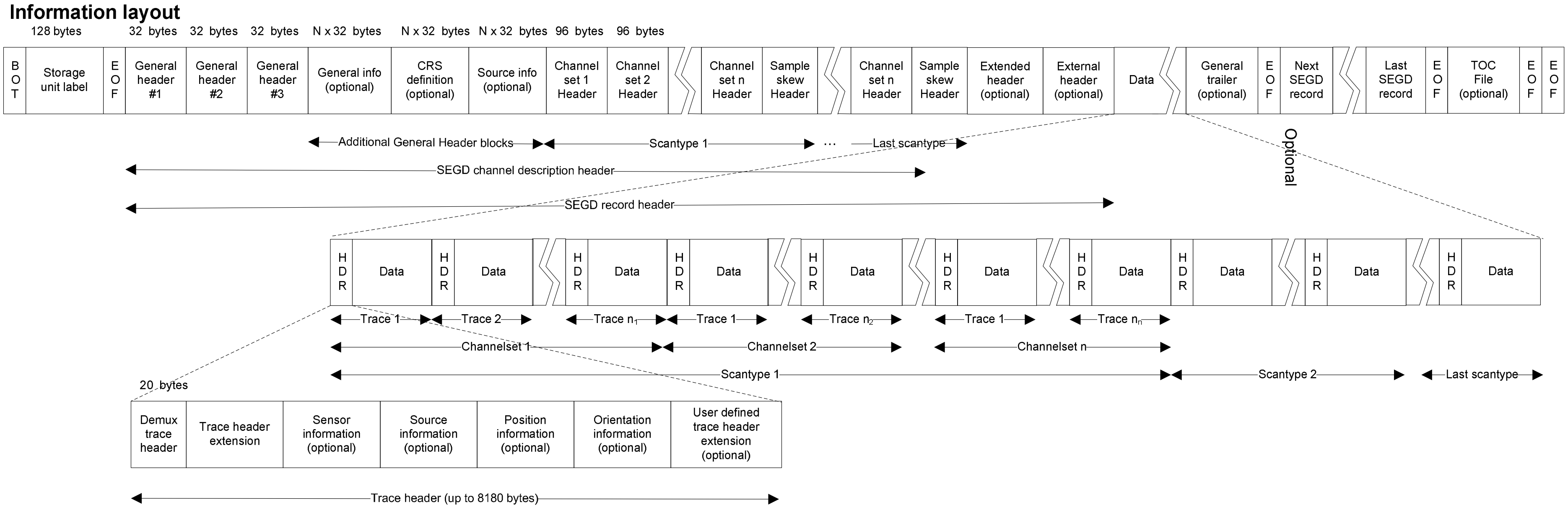


Figure 1. Record Format

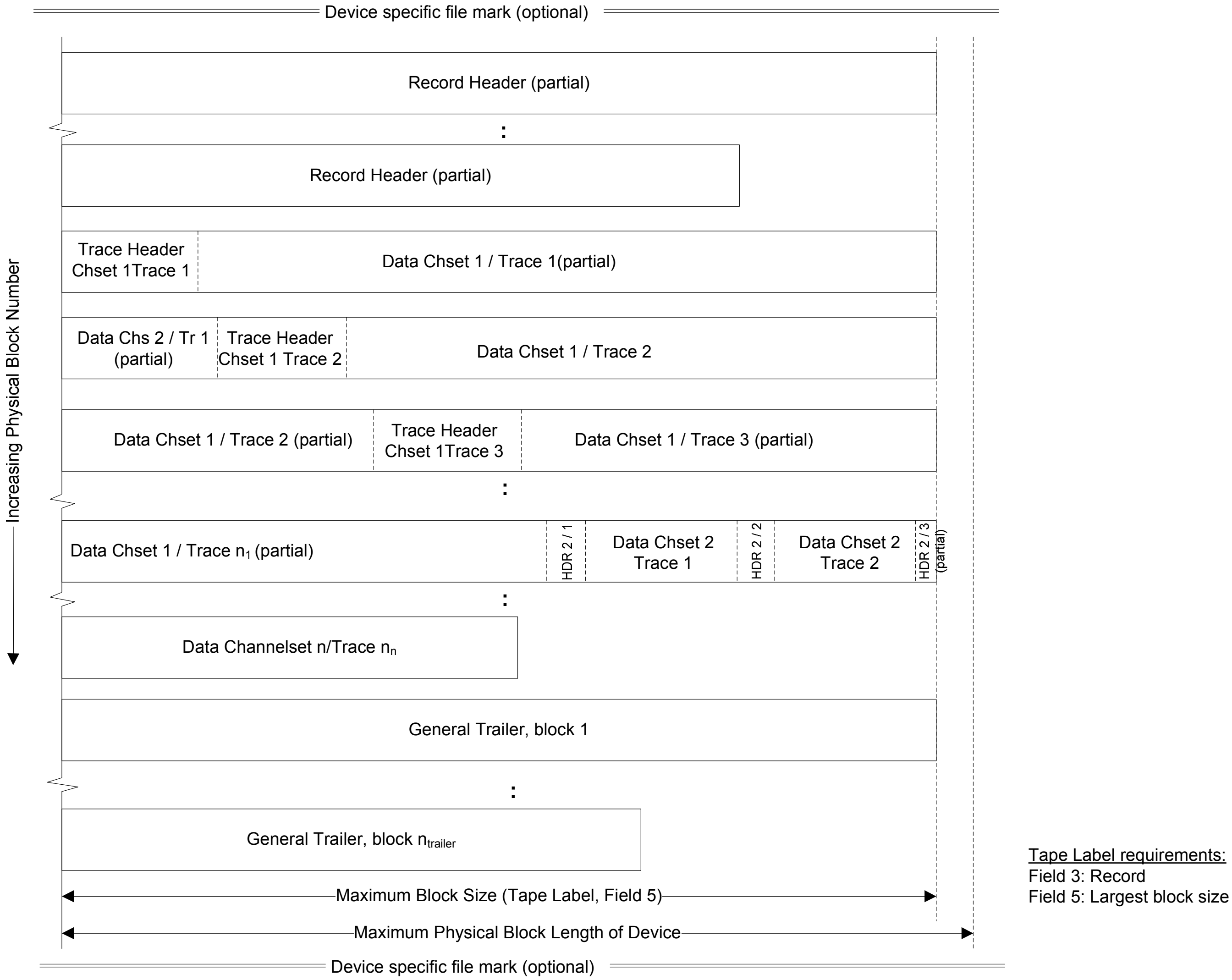


Figure 2. Variable Block Length Format Example

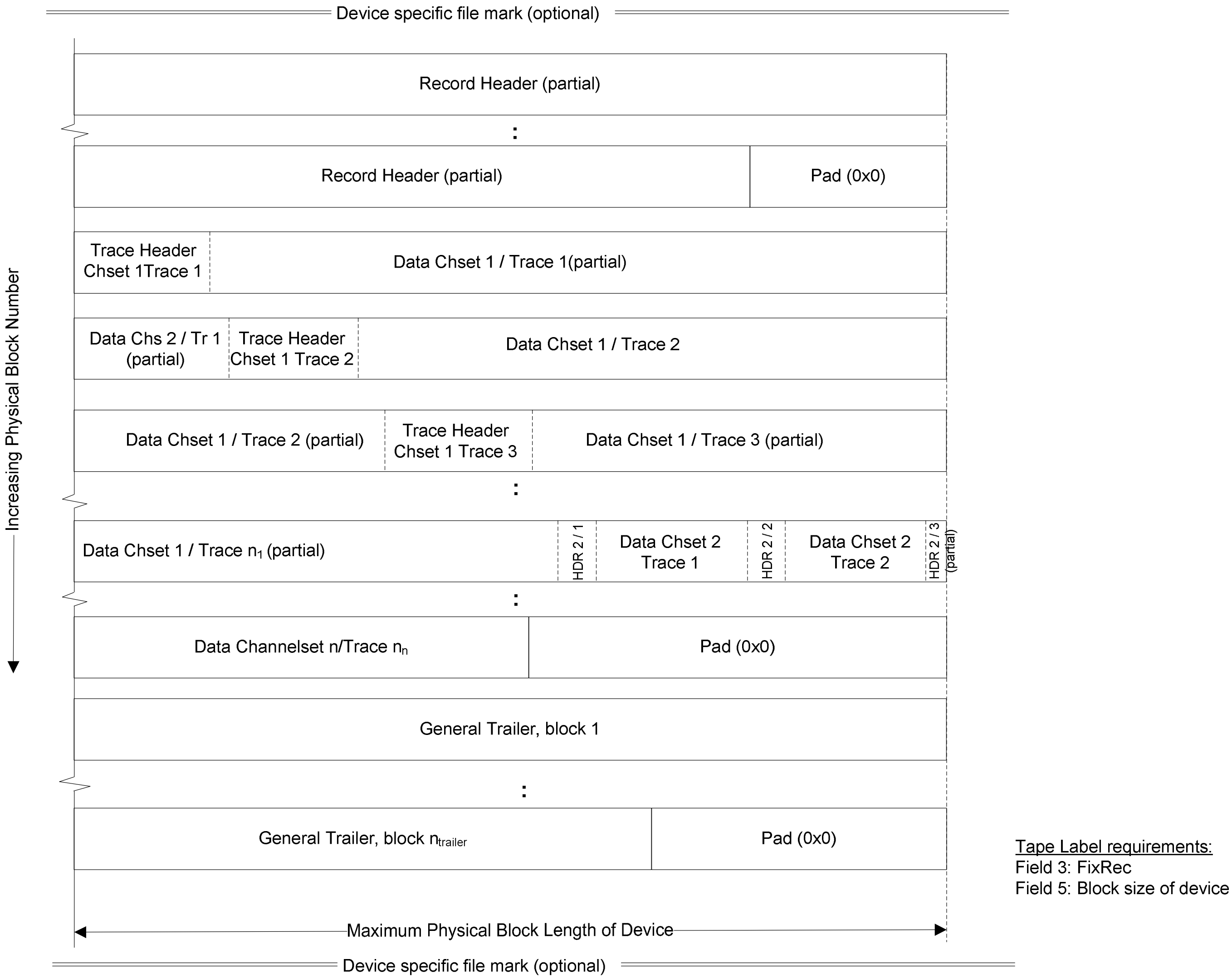
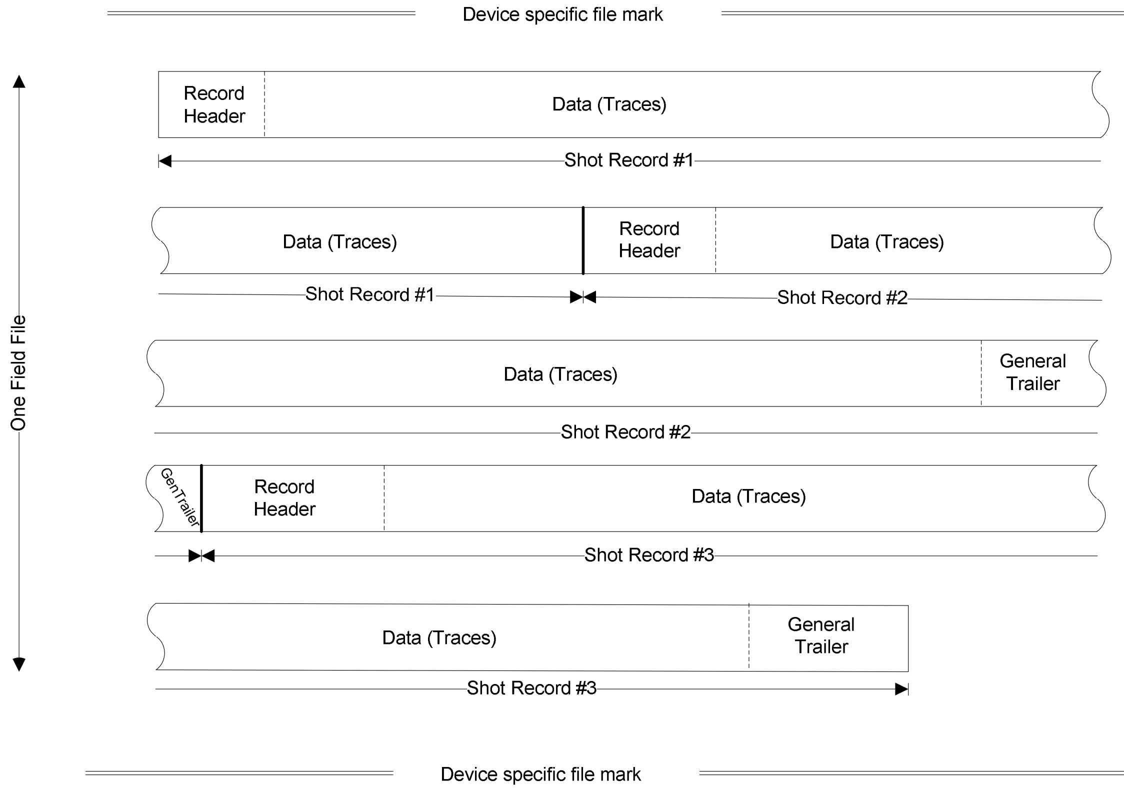
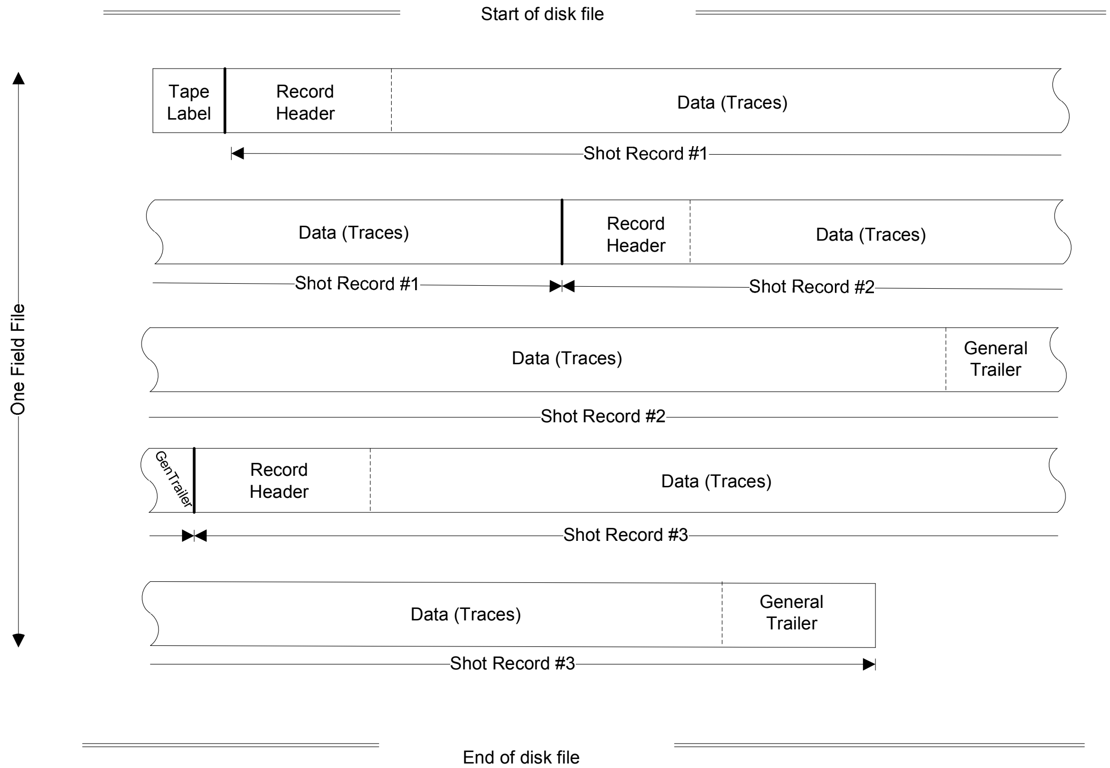


Figure 3. Fixed Block Length Format Example



<u>Tape Label requirements:</u>
Field 3: Record
Field 5: Largest Block Size set to 1
Field 10: Max shots/file set to number of records between file marks (in this example 3)

Figure 4. Byte Stream Device Example (3 records with 6 Channel Sets, and changing General Trailer)



Disk file
Format is similar to Byte Stream device, however each disk file is a storage unit starting with a Tape Label, and may contain any number of SEG-D records.

Network transfer (e.g. Ethernet)
Similar format is used when transferring SEG-D records across networks. Every time a network port is opened/transfer initiated, it is viewed as starting a new storage unit, hence a Tape Label must be sent prior to any SEG-D records. If number of records to be sent is unknown at the initiation of transfer, Field 10 of the Tape Label is set to 0.

Tape Label requirements:
Field 3: Record
Field 5: Largest Block Size set to 1
Field 10: Max shots/file set to number of records in file (in this example 3). If unknown at the time of Tape Label creation, set to 0 (undefined/unknown)

Figure 5. Disk File Example (3 records with 6 Channel Sets, and changing General Trailer)

3.1 SEG-D timestamp

SEG-D Rev 3.0 supports a more accurate time stamp than provided by the earlier versions. (Rev. 2.x had only a timestamp accurate to one second stored in General Header Block #1.) There is need to be able to give each sample an accurate timestamp, to be able to tie seismic data to external events, externally acquired data, or similar. Previously every manufacturer had their own proprietary time format, e.g. by storing a timestamp as part of the Extended or External Header. As of SEG-D Rev 3.0, a timestamp is stored in General Header Block #3, bytes 1 to 8. It is still permissible to store timestamps as part of the External or Extended Header, however the official time of the SEG-D record is now found in General Header Block #3.

Definition

> *A SEG-D Rev 3.0 timestamp is an 8 byte, signed, big-endian integer counting the number of microseconds since 6 Jan 1980 00:00:00 (GPS epoch). The timestamp is equal to GPS time converted to microseconds.*

This 8 byte timestamp is heretofore referred to as a *SEG-D timestamp*.

The number range an 8 byte microsecond count supports is approximately 9223372036854 seconds or 292471 years.

Dates before 6 Jan 1980 00:00:00 will have negative timestamps. The earliest timestamp allowed in prior revisions of SEG-D was 1 Jan 1970 00:00:00 (SEG-D epoch). See below for more details regarding conversion between timestamp and UTC.

There are several reasons why GPS time has been selected as basis of SEG-D timestamp:

- Most seismic acquisition systems use GPS for positioning, and hence have GPS time easily available
- GPS time does not contain leap seconds, which would cause a problem for processing of any record crossing leap second boundaries (remember, in Rev 3.0 measurements at several different times during the record may exist). For this reason, UTC was not chosen as basis for timestamp.
- There are other time formats without leap seconds, such as International Atomic Time (TAI) which tracks proper time on the Earth's geoid; however GPS is more commonly used in seismic acquisition systems.
- There are proposals for redefinition of UTC (to eliminate leap seconds). This would cause problems for handling historical data (i.e. tapes), as there would be a difference in definition before and after the change.
- Leap seconds are unpredictable (the Earth rotation speed changes at an unpredictable rate), so it is not possible to create a mathematical formula to calculate leap second.

<u>Relative Time Mode record</u>

For recording systems without absolute time, the timestamp in General Header Block #3 (bytes 1 to 8) must be set to 0, and the Relative Time Mode in General Header Block #3 (byte 30) set to 1. All timestamps in the record must then be relative to start of record (time zero). If reference to absolute time is required for such a record, the manufacturer is responsible for providing the required information separately.

The time stamp in General Header Block #1 can in Relative Time Mode either be set to 1 Jan 1970, an approximate value of the actual time, or a relative time compared to other records in the survey, whatever the manufacturer determines would best help interpretation of the data.

SEG-D timestamp and UTC conversion

There are a few problems with choosing GPS time as foundation for a SEG-D timestamp, mainly related to the timestamp in General Header Block #1, which is UTC (Coordinated Universal Time) or GMT (Greenwich Mean Time). This has not been changed in Rev. 3.0 for backward compatibility. However, when reading and writing SEG-D Rev 3.0 records, it is important to handle the differences, i.e. by adding the proper number of leap seconds.

GPS was aligned with UTC on 6 Jan 1980, but due to the variation in Earth's rotation, the definition of seconds, etc., leap seconds have to be added to UTC. As of April 2009, *UTC time is 15 seconds behind GPS time*.

The sequence of dates of the UTC second markers during leap second are:

31 December 2005, 23h 59m 59s
31 December 2005, 23h 59m 60s
01 January 2006, 0h 0m 0s

Note: *The interpretation of UTC timestamps needs to allow 60 as a value of seconds! This was previously not accounted for in the Rev 2.x standard.*

As GPS time is not defined prior to 6 Jan 1980, leap seconds are added to UTC as defined by International Atomic Time (TAI) such that the SEG-D timestamp equals TAI time plus 19 seconds at the TAI epoch (1 Jan 1972 00:00:00). (The reason for the 19 seconds difference is the TAI choice to start off with an initial difference to UTC of 10 seconds (TAI was synchronized with UTC at the start of 1958, and has drifted ever since), then 9 leap seconds were added during the 1970s. As GPS time was synchronized to UTC 6 Jan 1980, GPS time is now running at a fixed 19 second difference to TAI.)

Prior to 1970 no official leap seconds existed, so SEG-D defines for the purpose of timestamp/UTC conversions that no leap seconds exist for 1970 and 1971 (this should have very few practical consequences).

UTC to SEG-D timestamp difference:

01 MAR 2008: UTC 14 seconds behind timestamp

06 JAN 1980: UTC equal to timestamp

01 JAN 1970: UTC 9 seconds ahead of timestamp

See Appendix E for examples of conversions between UTC and SEG-D timestamp.

The table below shows leap seconds added to UTC time since 1970. Note the values for 1970 and 1971 (in italics) are not according to international standards; they are just values used to allow conversion of dates back to 1 Jan 1970 (SEG-D epoch):

Year	**30 June**	**31 Dec**
1970	*none*	*none*
1971	*none*	*none*
1972	+1 second	+1 second
1973 thru 1979	none	+1 second each year
1980	none	none

1981	+1 second	none
1982	+1 second	none
1983	+1 second	none
1984	none	none
1985	+1 second	none
1986	none	none
1987	none	+1 second
1988	none	none
1989	none	+1 second
1990	none	+1 second
1991	none	none
1992	+1 second	none
1993	+1 second	none
1994	+1 second	none
1995	none	+1 second
1996	none	none
1997	+1 second	none
1998	none	+1 second
1999 thru 2004	none	none
2005	none	+1 second
2006	none	none
2007	none	none
2008	none	+1 second
2009	0	

3.2 Multi-component data

Rev 2.x had limited support for multi-component data, but with several deficiencies. Rev 3.0 introduces several new sensor types, and Position/Orientation header blocks to better cope with multi-component acquisition.

Multi-component data should be recorded with each component in separate channel sets, i.e. the X component in one channel set, the Y component in another, etc. The component type must be clearly indicated in the Channel Set Header and Trace Header Extension. The indices in Trace Header Extension #1 (i.e. receiver line/point/point index/group index/depth index/reshoot index) will indicate which traces have to be processed together as a unit, e.g. for rotation. The group index will indicate the component number. The sensor type will indicate which type of data each trace contains. Component numbering must be made such that

$$\begin{bmatrix} C_{north} \\ C_{east} \\ C_{vertical} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} * \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

C is a component as indicated, R is the rotation matrix described by the values found in the Orientation Header. C_1, C_2, C_3 indicates the component numbers (group index) that must be used in the trace extension header. C_{north}, C_{east}, $C_{vertical}$ are components in the indicated direction after rotation. Please refer to the drawing in the Orientation Header section for a better description of the different components.

3.3 Extended recording mode

To be able to support different types of advanced recording systems requiring very long record times, SEG-D supports an *Extended Recording Mode.*

Extended Recording Mode is turned on by setting the Extended Recording Mode flag of General Header Block #3 to 1 (byte 29). The trace will then start at the time indicated by the Timestamp Block in the Trace Header, not as calculated by using the Start Time (byte 5–8) of the Channel Set Header. As a consequence of this, the trace in Extended Recording mode can utilize the entire 4,294,967,296 microseconds (or approximately 4294 seconds) trace length.

3.4 Permanent recording systems

SEG-D Rev 3.0 is also designed to support permanent system recording, though its primary usage is online acquisition of land/marine/seabed/transition zone surveys.

A permanent system consists of units with typically a few sensors, and recording can last for days or even weeks. SEG-D 3.0 supports traces of up to 2147 seconds in length. To record data for longer than this, multiple channel sets must be used, i.e. the same traces continue into the next channel set. See *Extended Recording Mode* above. Multiple traces from the same sensor are not allowed in one channel set, but it is not necessary to make a new scan type for the next channel set. This allows recording of up to 4294 seconds (time per trace) * 65535 (max number of channel sets) = 281,407,290 seconds or approximately 3257 days. If longer recording is required, multiple SEG-D records must be created. (Note: The example above assumes only one channel type is recorded. If multiple channeltypes are needed, i.e. multiple channelsets must be created, the maximum record length will be reduced accordingly.)

The sizes of headers have been extended to support the extended amount of meta information required for permanent surveys. Metadata recording incidents occurring throughout the record may appear as part of the External or Extended Headers, inserted into the Trace Header Extension, or appended to the General Trailer. For devices with limited amount of memory and CPU, the General Trailer option might be preferable. All metadata should be tagged with the SEG-D timestamp (if time related), and/or scantype#/channel set#/trace# (if trace related, and information not recorded as part of the Trace Header).

3.5 Time drift

Some acquisition systems deploy single sensor unit for long periods of time without synchronization with other parts of the acquisition system or an external clock reference (e.g. GPS). Time drift (i.e. the clock running at a faster or slower rate than GPS) is then a problem. SEGD supports recording the time drift for each individual trace in the Time Drift Header block by storing the time of deployment and retrieval of the sensor, and the corresponding time offsets.

The data can then be time drift corrected using these values directly (linear time correction). If a more advanced time drift correction is desired, the manufacturer is free to do so. The manufacturer must then provide the time drift correction information in manufacturer specific header blocks or possibly other external data.

Trace data from a sensor unit may be recorded in SEGD format using a drifting clock, both in normal and extended recording mode. However all channels within the record must have the same time drift (i.e. use the same time reference system).

Note:
Trace data ***must*** be time drift corrected before merging trace data from multiple drifting sensor units into one single SEGD record, i.e. start of trace must be aligned, and sample rate must be corrected. SEGD does not support irregular (drifting) sample intervals or time skew between channels other than specified in the skew blocks and trace sample skew values.

See Appendix E, EM survey, for more details regarding the time drift correction.

3.6 Positions in SEG-D

SEG-D allows for multiple coordinate tuples for any location. A coordinate tuple is an ordered set of coordinates describing a position. For redundancy, two coordinate tuples can be given in different coordinate reference types for any position, e.g. geographic and projected. Coordinate reference system types are:

CRS type	**Coordinate Tuple**		
	Coord 1	**Coord 2**	**Coord 3**
projected (see note 1)	easting or northing (see note 2)	northing or easting (see note 2)	(not used)
geographic2D	latitude	longitude	(not used)
geographic3D	latitude	longitude	ellipsoidal height
vertical	(not used)	(not used)	gravity-related height or depth (see note 3)
geocentric	geocentric X	geocentric Y	geocentric Z
compound geographic+vertical	latitude	longitude	gravity-related height or depth (see note 3)
compound projected+vertical	easting or northing (see note 2)	northing or easting (see note 2)	gravity-related height or depth (see note 3)

Notes:
1) Sometimes called "map grid".
2) There is significant variation worldwide in the convention used for projected CRS axis order and abbreviation. In some cases the easting will be given before the northing and in other cases the order will be northing before easting. In both of these scenarios the axes *may* be labeled X and Y; in such instances the first coordinate will be labeled X regardless of whether easting or northing and the second coordinate labeled Y.
3) Whether vertical coordinates are heights (positive up) or depths (positive down) will be defined in the CRS definition.

These three fields need to hold the following values:

	Coord 1		**Coord 2**		**Coord 3**	
	min	**max**	**min**	**max**	**min**	**max**
Value range:	−9,999,999	60,800,000	−9,999,999	60,800,000	−6,400,000	6,400,000
Resolution:	10 significant figures		10 significant figures		9 significant figures	
All coordinate fields are to be capable of being signed.						

Every coordinate tuple must be time-stamped, allowing multiple coordinate tuples to be given for any source or receiver location in the record. Coordinates for each location may be given for various phases of a project – planning, execution, etc.

Coordinates only define location unambiguously when the Coordinate Reference System (CRS) to which they are referenced has been identified. This is done through the Location Data and/or Location Data EPSG Reference extended textual stanza. That Location Data stanza may be supplemented with a Location Data Transformation extended textual stanza. These stanzas are defined in Appendix D and stored in the General Header in multiple Coordinate Reference System Identification blocks.

All coordinates in a SEG-D file should preferably reference the same coordinate reference system. One CRS definition and Location Data stanza then applies to all locations. Should there be a requirement to store coordinates in one of a number of CRSs, then each CRS requires identification through separate Location Data and/or Location Data EPSG Reference extended textual stanzas. Each coordinate set must be related to the relevant CRS through a reference to the relevant Location Data Stanza ID value, a single integer number between 1 and 65535.

4.0 SEG-D, Rev 3.0 Structure

4.1 Rev 3.0 Storage Unit Label (Tape Label)

The first 128 bytes of data on a Rev 3.0 tape must consist of ASCII characters and will constitute a Storage Unit Label. This label is very similar to the RP-66 storage unit label. The Storage Unit Label is also often referred to as the "Tape Label" for historical reasons. The label format is summarized in the table below.

If the tape media supports multiple partitions, SEG-D data may be written to any of the partitions of the tape, each beginning with a Storage Unit Label. Data from one partition can not "run over" into a subsequent partition, each partition must be capable of being decoded in isolation.

On one tape, it is allowed to mix partitions containing SEG-D data with partitions containing non SEG-D formatted information.

Table 1: Label

Field	Description	Bytes	Start – end byte
1	Storage unit sequence number	4	1 – 4
2	SEG-D Revision	5	5 – 9
3	Storage unit structure (fixed or variable)	6	10 – 15
4	Binding edition	4	16 – 19
5	Maximum block size	10	20 – 29
6	API Producer organization code	10	30 – 39
7	Creation date	11	40 – 50
8	Serial number	12	51 – 62
9	Reserved	6	63 – 68
10	Storage set identifier	60	
11	External Label Name	12	69 – 80
12	Recording Entity Name	24	81 – 104
13	User defined	14	105 – 118
14	Max shot records per field record	10	119 – 128

Field 1 The **Storage Unit Sequence Number** is an integer in the range 1 to 9999 that indicates the order in which the current storage unit occurs in the storage set. The first storage unit of a storage set has sequence number 1, the second 2, and so on. This number is represented using the characters 0 to 9, right justified with leading blanks if needed to fill out the field (No leading zeros). The rightmost character is in byte 4 of the label. *This field is optional*. If not used, it must be blank (filled with blank characters). This implies that this is the only storage unit within the storage set. Separate Storage Sets should be used for different data types.

Field 2 The **SEG-D Revision** field indicates which revision of SEG-D was used to record the data on this tape. SD3.0 indicates that the data was recorded using SEG-D, Revision 3.0 (SD2.1 in previous revision) - *This field is required.*

Field 3 **Storage Unit Structure** is a name indicating the record structure of the storage unit. This name is left justified with trailing blanks if needed to fill out the field. The leftmost character is in byte 10 of the label. For SEG-D, Rev 3 tapes, this field must contain "RECORD" or "FIXREC". *This field is required*.

"RECORD" — Records may be of variable length, ranging up to the Blocksize length specified in the maximum Block size field of the storage unit label (if not zero). If the maximum Block size specified is zero, then records may be of any length.

"FIXREC" — All records in the storage unit have the same length, namely that specified in the maximum Block size field of the storage unit label. Although all storage units in the same storage set must have a FIXREC structure, the maximum record length may be different in different storage units. When the FIXREC option is used, then the maximum record length field shall not be 0 (zero).

Field 4 **Binding edition** is the character B in byte 16 of the label followed by a positive integer in the range 1 to 999 (no leading zeros), left justified with trailing blanks if needed to fill out the field. The integer value corresponds to the edition of the Part 3 of the API, RP66 standard used to describe the physical binding of the logical format to the storage unit. *This field is required.*

Field 5 **Maximum Block Size** is an integer in the range of 0 to 4,294,967,295 ($2^{32}-1$), indicating the maximum block length for the storage unit, or 0 (zero) if undeclared. This number is represented using the characters 0 to 9, right justified, with leading blanks if necessary to fill out the field (no leading zeros). The rightmost character is byte 29 of the label. A valid value or 0 (zero) must be recorded. A value of 1 means byte stream device (e.g. disk). *It is highly recommended to indicate the maximum block length where possible, as this will determine the size buffer needed to read the data, and will enable reading the data at a more optimal speed (using multi-block read).*

Field 6 **Producer organization code** is an integer in the range of 0 to 4,294,967,295 ($2^{32}-1$) indicating the organization code of the storage unit producer. This number is represented using the characters 0 to 9, right justified, with leading blanks if necessary to fill out the field (NO leading zeros). The rightmost character is byte 39 of the label. *This field may be empty, i.e. may contain all blanks, in which case no storage unit producer is specified (i.e. same as RP-66 V2).*

Organization codes are assigned by Energistics, formerly the Petrotechnical Open Standards Consortium (POSC), which maintains the current list of codes. To request a new organization code, contact:

Energistics
24 E. Greenway Plaza
Suite 1315
Houston, TX 77046-2414 USA
+1 713 784-1880 telephone
+1 713 784-9219 fax
info@energistics.org

Field 7 **Creation date** is the earliest date that any current information was recorded on the storage unit. The date is represented in the form dd-MMM-yyyy, where yyyy is the year (e.g. 1996), MMM is one of (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC), and dd is the day of the month in the range 1 to 31. Days 1 to 9 may have one leading blank. The separator is a hyphen (code 45_{10}). *This field is required.*

Field 8 **Serial number** is an ID used to distinguish the storage unit from other storage units in an archive of an enterprise. The specification and management of serial numbers is delegated to organizations using this standard. If an external label is used the name/number must be a subset of the serial number or the External Label Name in Field 10, and must occupy the rightmost characters in the serial number (or External Label Name). *This field is required.*

Field 9 This field is reserved and should be recorded as all blanks (code 32_{10}).

Field 10–12 The **Storage set identifier** is a descriptive name for the storage set. Every storage unit in the same storage set shall have the same value for the **user defined portion** of the storage set identifier in its storage unit label. Included in the **Storage Set Identifier** is the **External Label Name**. The characters in this field are right justified with leading blank characters as required. If the tape does not have a physical label, then this field must be blank. A physical label is optional, but if it exists, then this field is required only if the external label is different from the lower 6 characters of the Serial Number in Field 8. The next field in the Storage set identifier is the **Recording Entity Name**. This must contain the crew number or name, or some other unique identifier which will differentiate the recording entity which recorded this data from any other recording entity within the organization (as included in field 6). The 24 bytes may by any alphanumeric characters. If multiple recording systems are used on a vessel or crew, then data recorded on each system must be clearly distinguished. For example, an ABC Geophysical crew (party 13), on the M/V Gopher, recording data on two Zip 6000 recording systems might have a Recording Entity Name on tapes recording on the first recording system of:

ABC, Gopher, P13, Zip#1

On the second system, the Recording Entity Name might be:

ABC, Gopher, P13, Zip#2

The Recording Entity Name field is required.

Field 13 **USER DEFINED**

The next 14 bytes in this field may contain any other user input information. The only restriction is that the data must be in ASCII.

Field 14 **Max Number of shot records per field record**. Field Records are data between File Marks (10 bytes).

It is not acceptable to use an ANSI label (or any other label or data) prior to the Storage Unit Label.

An external, physical label is not required.

4.2 Rev 3.0 Table of Contents (TOC) File (optional)

The SEG-D TOC FILE is a table of contents for a specific tape/file/storage unit. The TOC file consists of a 448 byte header defining general survey information, followed by a list containing one 64 byte entry for each SEG-D record on tape. The purpose of this list is to provide fast location of data, and give an overview of the storage media without scanning through the entire e.g. tape.

The TOC file is optional, however for tapes and other large media it is highly recommended to create a TOC file.

Note: The recorder is responsible for allocating enough space at the end of the storage media to allow storage of the TOC file. The size of the TOC file is not large, please refer to the end of this section for an example of TOC file size calculation.

The SEG-D TOC file may be stored on disk to provide an overview of, and fast access to, the data on the media. It is critical that the Serial Number in the Tape Label, and the SEG-D TOC Header matches to facilitate this functionality. The naming convention of these files should include the Serial Number, Storage Unit Sequence Number etc. from the Tape Label, to ease the matching of TOC file to tape media.

On tape the TOC file must be separated from the other data with a File Mark before and after. The TOC file is most commonly stored at the end of tape (after last SEG-D record, before any double File Mark or end of media), though it may also be stored at the beginning, directly following the Tape Label.

On disk (and similar devices), where no File Marks exists, the TOC file will not be separated from the rest of the data, hence the reader will need to recognize the TOC File Header *TOC identifier* `"SEGD TOC FILE".`

The TOC file consists of a header, followed by one TOC entry for each SEG-D record on the storage media.

4.2.1 TOC Header

Field	Description	Data type	Start – end byte
1	TOC identifier.	ASCII text	1 – 13
2	SEG-D Revision	ASCII text	14 – 19
3	Serial number	ASCII text	20 – 31
4	Client Identification	ASCII text	32 – 63
5	Vessel/Crew Identification	ASCII text	64 – 95
6	Country	ASCII text	96 – 127
7	Region	ASCII text	128 – 159
8	Block	ASCII text	160 – 191
9	Survey Area Name	ASCII text	192 – 223
10	Contract Number	ASCII text	224 – 255
11	License ID	ASCII text	256 – 287
12	Survey Type	ASCII text	288 – 307
13	Acquisition Contractor	ASCII text	308 – 339
14	Acquisition Dates (from-to)	ASCII text	340 – 364
15	Job Identification	ASCII text	365 – 396
16	Media Sequence Within Record Set	ASCII text	397 – 402
17	Data Type Identifier	ASCII text	403 – 434
18	Undefined, fill with spaces (ASCII code 32_{10})	ASCII text	435 – 438
19	Number Of Entries In TOC File	ASCII text	439 – 448

The ASCII text fields are left justified, space (‘ ‘, ASCII 32_{10}) padded, unless otherwise specified. ASCII fields may be left blank, i.e. contain only space (‘ ‘, ASCII 32_{10}) if they are not relevant.

If data from multiple surveys are present on the same SEG-D media, most fields of the header may be left blank. If some fields are common to all data, they may be filled in. Note: Storing data from e.g. multiple surveys onto the same tape is not recommended.

Field 1 **TOC identifier**. Text indicating this is a TOC file. The field should be set to `"SEGD TOC FILE",` *This field is required,*

Field 2 **SEG-D Revision.** Indicates which SEG-D revision data is stored on the media. The field should be set to SD3.0 to indicate SEG-D Revision 3.0. Must be identical to the SEG-D Revision (Field 2) of the Tape Label of this storage unit (i.e. tape/disk file). *This field is required*.

Field 3 **Serial number.** Must be identical to the Serial Number (Field 8) of the Tape Label of this storage unit (i.e. tape/disk file). *This field is required*.

Field 4 **Client Identification**. The name of the client. Must match General Header #6 of the SEG-D records (if present).

Field 5 **Vessel/Crew Identification**. The name of the vessel or crew acquiring the data. Must match General Header #4 of the SEG-D records (if present).

Field 6 **Country**. The name of the country the data was acquired in. If multiple countries are involved, they should all be listed here.

Field 7 **Region**. The region of the country the data was acquired in.

Field 8 **Block**. The block name of the survey area. If multiple blocks are involved, they should all be listed.

Field 9 **Survey Area Name**. The name of the survey area. Must match General Header #5 of the SEG-D records (if present).

Field 10 **Contract Number**. The number/name of the survey contract.

Field 11 **License ID**. The license identification of the

Field 12 **Survey Type**. The type of survey, the presently recommended are:

```
“TOWED MARINE SEISMIC“
“LAND SEISMIC        “
“TRANSITION ZONE     “
“SEABED SEISMIC      “
“VERTICAL SEISMIC    “
“ELECTROMAGNETIC     “
“COMBINATION         “
“OTHER               “
```

The COMBINATION survey is any combination of the survey types, e.g. combined TOWED MARINE and ELECTROMAGNETIC. Other strings may be used if this increases the usability of the TOC file, however it is recommended to use the standardized strings whenever possible.

Field 13 **Acquisition Contractor**. The (legal) name of the acquisition contractor. Must uniquely identify the acquisition contractor.

Field 14 **Acquisition Dates (To-From)**. The range of dates of the data on this media. If data from only one day is stored, to and from date is set to same date. The format is "DD-MMM-YYYY – DD-MMM-YYYY", e.g. "03-JAN-2009 – 04-JAN-2009"

Field 15 **Job Identification**. The acquisition contractor job name/number/identification for the survey. Must match General Header #7 of the SEG-D records (if present).

Field 16 **Media Sequence Within Record Set**. The sequence of this storage media within this record set. The sequence starts counting at 1, and is increased with 1 for each storage media (e.g. tape) for the record set (e.g. sequence or swath). For storage media with data from multiple record sets, this field is set to "1".

Field 17 **Data Type Identifier**. The data type identifier states which type of data is present on this storage media. The acquisition system can freely define the contents. Example: "RAW SEISMIC ORIGINAL DATASET 1", "ROT/FILT DATASET COPY 2".

Field 19 **Number Of Entries In TOC File**. The number of TOC entries following this header. Field format is right justified, space (' ', ASCII 32_{10}) padded integer value. *This field is required.*

4.2.2 TOC Record Entry

Field	Description	Data type	Start – end byte
1	Media File Number	ASCII text	1 – 8
2	Record Within File	ASCII text	9 – 15
3	Time of shot	ASCII text	16 – 35
4	SEG-D File Number	ASCII text	36 – 45
5	Record ID	ASCII text	46 - 50
6	Record Set Number	ASCII text	61 – 70
7	Line identification (name/number)	ASCII text	71 – 94
8	User defined.	ASCII text	95 – 128

The ASCII text fields are right justified, space (' ', ASCII 32_{10}) padded, unless otherwise specified. ASCII fields may be left blank, i.e. contain only space (' ', ASCII 32_{10}) if they are not relevant.

Field 1 **Media File Number**. This field is the media file number containing the SEG-D record. Field format: Unsigned integer. The number starts counting from 1. The Tape Label has Media File Number 1, the first SEG-D record is usually contained in Media File Number 2. *This field is required.*

Field 2 **Record Within File**. Multiple SEG-D records may be stored within one single media file (though this is not recommended for field tapes). Record Within File starts counting from 1, i.e. the first SEG-D record has Record Within File = 1 (and starts at byte 0 of the media file). Field format: Unsigned integer. *This field is required.*

Field 3 **Time Of Shot**. This field is identical to the Time Zero of the SEG-D record (General Header Block #3, bytes 1–8), and matches the timestamp in General Header Block #1 (bytes 11–16). Field format: Signed integer. *This field is required*.

Field 4 **SEG-D File Number**. The SEG-D file number as listed in the General Header of the SEG-D record (bytes 1–2 of General Header Block #1, or bytes 1–3 of General Header Block #2). Field format: Unsigned integer. *This field is required*.

Field 5 **Record ID**. The record ID identifies the record within the record set. Field format: Signed floating point value, some examples: “4” “4.0” “4.000” “-4.0512”. This may be the shot point or source point number if available, but can be any number that identify this record within the record set.

Field 6 **Record set number**. The record set this SEG-D record belongs to. Must be the same as bytes 21-22 of General Header #2. Field format: Unsigned integer.

Field 7 **Line identification (name/number)**. The line number or name of the line the record belongs to. For surveys where both receiver and source line numbers exists, it is recommended to set this field to the source line number whenever possible. Set to empty (all spaces, ASCII 32_{10}) if not valid. Must match General Header #8 Line Identification field of the SEG-D record (if present). Field is left justified space (‘ ‘, ASCII 32_{10}) padded ASCII string.

Field 8 **User defined**. 34 bytes of user defined ASCII text. The field can be freely utilized by the recorder, but the contents must be ASCII text. Record as empty (fill with space (‘ ‘, ASCII 32_{10}) if not used).

TOC file size calculation example:

Size of shot:
5000 channels, 10 seismic seconds (i.e. 10240 milliseconds) record length, SEG-D 8036 format (3 byte per sample) and 2 millisecond sampling rate. All traces recorded for the full record time, each trace has five 32 byte Trace Header Extension blocks. (For simplicity we ignore the SEG-D record header, it is assumed to be small compared to the data).

5000 * (10240 / 2 * 3 + 5 * 32 + 20) = 77700000 bytes = 74.1 MB

Size of tape:

300 000 MB (or ~300 GB)

Size of table of contents file:

448 + 300000/74.1 * 128 = 507 kB (approximately)

The recorder of this dataset will need to leave 507 kilobytes of space at the end of a 300 GB tape for storing the TOC file.

4.2.3 Using the TOC with SEG-D on disk

The Table of Contents is designed to aid in rapid access to data stored on tape and other sequential record-oriented media where random access to arbitrary byte locations on the media is unavailable. For SEG-D datasets created as one huge contiguous byte stream disk file, the TOC would be trivial and of quite modest utility, with just a list of TOC Record entries showing the SEG-D File Numbers in that file, but providing no way of locating those records short of reading through the disk file until the desired record is encountered (note that the reading is quite simple though as GeneralHeader

#3 contains size of record allowing fast jump to next record). For this reason, as well as to limit the loss of data should a disk file be corrupted, we recommend that SEG-D data be created directly on disk in the following manner:

- The SEG-D Storage Unit Label and each Record should be written to a separate disk file.

- The naming convention of these files should include the numeric Media File Number from the TOC Record Entry field. We also recommend the disk file name include the Record Set Number from that same TOC Record Entry and the Media Sequence Within Record Set from the TOC Header prepended before the Media File Number.

For SEG-D records copied to disk from tape, the existing TOC is copied to a disk file of its own. In order for this copied TOC to be immediately valid for the disk copy, we recommend the data be copied to disk in the following manner:

- Each individual tape file (which may include multiple SEG-D File Numbers) should be copied to a separate disk file. A dummy SEG-D Storage Unit Label must be prepended the data. It is recommended that the dummy label is based on the tape Storage Unit Label, but modifications done such that it will not be confused with the real tape data. It is however strongly recommended to include a reference to the original tape media in the dummy label, e.g. by utilizing the "User defined" portion of the Storage Unit Label.

- The naming convention of these files should include the numeric Record Set Number, the Media Sequence Within Record Set and Media File Number from the existing TOC.

For ease of listing and sorting, the numeric fields embedded in the file names should each be of a fixed length, with leading zeros padded instead of blanks, so as to accommodate the largest possible value for that field. So, for example, if we use the template FILE_[rsn]_[mseq]_[mfn].SGD for file names, the Record Set Number (rsn) should be 5 digits long, the Media Sequence Within File (mseq) should be 6 digits long, and the Media File Number 8 digits long. This results in a file name of 29 characters, quite manageable on all modern computer systems.

Note: There are a lot of other ways of storing SEG-D data on disk, for example storing each individual tape file as a disk file in a filesystem directory. This is perfectly legal, and may be very useful for internal data storage, but it is not considered proper SEG-D data as each disk file does not start with a Storage Unit Label and the data can not be treated as a stream of bytes (each disk file must be read separately).

5.0 Header blocks

The headers are blocks of data prior to the seismic data which contain auxiliary information about the seismic data, the acquisition parameters, acquisition geometry, and user-defined information. The header block includes at least three General Header Blocks, zero or more Scan Type headers, and optional Extended and External headers. Trace Headers are included in conjunction with each seismic data trace. Sections 7 and 8 detail the content of each type header.

In addition to header blocks which are recorded prior to the seismic data traces, an optional General Trailer is allowed following the seismic data. This allows recording other auxiliary information which is not available at the beginning of the record. Sections 7 and 8 include detailed description of the allowed fields of the General Trailer.

5.1 General Headers (General Header Block #1, #2 and #3 are required)

General Header Block #1 is 32 bytes long and contains information similar to SEG A, B, C, and the original SEG-D headers. Abbreviations are as close as possible to those used in previous formats.

SEG-D, Rev 3.0 requires the use of General Header Block #1, General Header Block #2 (as was also required in SEG-D, Rev 2.x), and General Header #3 (new with SEG-D, Rev 3.0). The General Headers define basic parameters for the record such as file number, time of record, number of channel sets, sizes of meta and auxiliary data, etc. General Header Block #2 allows extended values exceeding the limits of General Header Block #1; the values of General Header Block #1 are then set to FF_{16} to indicate the actual value will be found elsewhere. Note that the Rev 2.1 Sequence Number has been renamed Record Set Number to be applicable for non-towed marine operations. For towed marine operations, the sequence number should be entered in that field. General Header Block #3 contains an accurate timestamp for the record in addition to size information for the record to allow quick searching through the record.

Bytes 1–3 in General Header Block #2 allow for a three byte, binary file number. To use this extended file number, bytes 1 and 2 in General Header Block #1 must be set to $FFFF_{16}$.

General Header Block #2 also allows for a two byte, binary number of Channel Sets/Scan Type in bytes 4 and 5. When using the Extended Channel Sets/Scan Type, byte 29 of the General Header Block #1 must be set to FF_{16}.

Additional blocks may be added as needed by the manufacturer or user.

Rev 3.0 has a much more flexible information structure in the General Header than Rev 2.x. All General Header blocks except General Header Blocks #1–3 are optional, and can appear in any order, and a lot more information can be entered. (Rev. 3.0 allows up to 65535 General Header blocks, compared to the 15 allowed in Rev. 2.x). To achieve this, byte 32 of all header blocks have been assigned an ID depending on its contents. The following are currently defined:

ID (Byte 32)	Contents (Block Name)
02_{16}	General Header 2
03_{16}	General Header 3 (Timestamp and size header)
$0F_{16}$	General Trailer description block
10_{16}	Vessel/Crew identification
11_{16}	Survey Area Name
12_{16}	Client Name
13_{16}	Job identification
14_{16}	Line Identification
15_{16}	Vibrator Source Information
16_{16}	Explosive source information
17_{16}	Airgun source information
18_{16}	Watergun source information
19_{16}	Electromagnetic source

$1F_{16}$	Other source type information	
20_{16}	Additional source information	
21_{16}	Source Auxiliary Channel Reference	
30_{16}	Channel set description block 1	*
31_{16}	Channel set description block 2	*
32_{16}	Channel set description block 3	*
40_{16}	Demux Trace header extension 1	*
41_{16}	Sensor Info trace header extension block	*
42_{16}	Time Stamp Header block	
43_{16}	Sensor calibration block	
44_{16}	Time drift block	
45_{16}	Electromagnetic Src/Recv desc block	
50_{16}	Position block 1	
51_{16}	Position block 2	
52_{16}	Position block 3	
55_{16}	Coordinate Reference System Identification block	
56_{16}	Relative position block	
60_{16}	Orientation Header block	
61_{16}	Measurement block	
70_{16}	General Trailer Description Block	
$B0_{16} – FF_{16}$	User defined header block	

Table 2: Header block types defined in SEG-D Rev 3.0

Note the table contains all standard defined header types, including those used in trace headers. This is done to allow the same information to be inserted into a General Header or a Trace Header, depending on what is most useful. However, there are a few types that are not allowed in a General Header; these are marked with '*' in the table. (There are also blocks, like Survey Area Name (11_{16}), which do not make much sense to put into the trace header, though it is allowed.)

Note:

Though there is support for user-defined block types, we recommend using only standard header blocks in the General Header, and placing user-defined information into an Extended or External header. These have much more available space available and allow a more flexible format. User-defined blocks are really intended to be used in the Trace Header.

The General Header is a sequence of 32 byte blocks of different types, identified by the ID in byte 32, describing general information about the entire record.

Information blocks commonly inserted into a General Header are Vessel/Crew Identification (10_{16}), Survey Area Name (11_{16}), Client Name (12_{16}), Job Identification (13_{16}), and Line Identification (14_{16}).

If the record is related to a single source event (shot), we recommend inserting information regarding the source (IDs 15_{16}–$1A_{16}$ depending on source type) into a General Header. If multiple source events occur in the record, e.g. in land slip-sweep operations, storing the source information in the relevant auxiliary source trace headers can be done.

5.2 Scan Type Headers

The Scan Type Header is used to describe the information about the recorded channels (filters, sampling intervals, sample skew, etc.). The Scan Type Header is composed of one or more channel set descriptors followed by skew information. The channel set descriptors must appear in the same order as their respective channel sets will appear within a base scan interval. A channel set, which is part of a scan type, is defined as a group of channels all recorded with identical recording parameters. One or more channel sets can be recorded concurrently within one scan type. In addition, there can be multiple scan types to permit dynamic scan type changes during the record (e.g., 12 channels at 1/2 ms switched at about 1 second to 48 channels at 2 ms). Where there are dynamic changes, Scan Type Header #1 describes the first part of the record, Scan Type Header #2 the second part, etc. Within the Scan Type Header, each channel set descriptor is composed of a 96 byte field, and up to 65535 channel set descriptors may be present. In addition, up to 99 scan type headers may be utilized in a record.

Following the channel set descriptors of a scan type are a number of 32 byte fields (SK, specified in byte 30 of General Header #1) that specify sample skew. Sample skew (SS) is recorded in a single byte for each sample of each subscan of each channel set, in the same order as the samples are recorded in the scan. Each byte represents a fractional part of the base scan interval (Byte 23 of General Header #1). The resolution is 1/256 of this interval. For instance, if the base scan interval is 2 msec, the least significant bit in the sample skew byte is 1/256 of 2 msec or 7.8125 microseconds.

A channel set is a group of channels that:

a) Use identical recording parameters. This includes the same record length and sampling interval.
b) Use identical processing parameters, including the same filter selection and array forming parameters.
c) Originates from the same streamer cable for marine data. The streamer cable number for each channel set is included in the channel set descriptor byte 31.
d) Consists of channels with the same group spacing. For example, if one streamer has short group spacing close to the boat and longer group spacing at long offsets, the data from the streamer would be recorded as two channel sets. The first channel in each channel set will start with trace number one.

The following is a list of ground rules for the scan type header:

1. The order in which channel sets are described in the header will be the same as the order in which the data are recorded for each channel set.

2. In a Scan Type Header containing multiple channel set descriptors with different sampling intervals, each channel set descriptor will appear only once in each scan type header. Within the data block, however, shorter sampling interval data are recorded more frequently.

3. In the case of multiple scan type records, such as the dynamically switched sampling interval case, each scan type will contain the same number of channel sets. Any unused channel sets needed in a scan type must be so indicated by setting bytes 21 to 23 (channels per channel set) to zero in the channel set descriptor.

4. In multiple scan type records, the number of bytes per base scan interval must remain a constant for all scan types recorded.

5. Channels within the same Channel Set must now have the same number of Trace Header Extensions. Since all traces within a Channel Set will contain the same number of Trace Header Extensions, the number of Trace Header Extensions will be indicated in the Channel Set Descriptor. Byte 28 of the Channel Set Descriptor contains the number of Trace Extension Headers in the channel set. This must match the byte 10 of the Demux Trace Header of traces in the channel set. Each trace is hence limited to a maximum of 255 Trace Header Extension blocks.

6. The length of each trace within a Channel Set is restricted to be the same value. This limitation and the restriction of the number of Trace Header Extensions to the same number within a Channel Set will result in each trace within a Channel Set being recorded with the same number of bytes.

5.3 Demux Trace Header (required)

The Demux Trace Header length is 20 bytes and is an identifier that precedes each channel's data. The trace header and the trace data are recorded as one block of data. A trace is restricted to one channel of data from one channel set of one scan type. Some of the information in the trace header is taken directly from the general header and the scan type header.

Bytes 7, 8, and 9 comprise the timing word that would accompany the first sample if these data were written in multiplex format. To obtain the exact sample time, the actual sample skew time (Byte 11 multiplied by the base scan interval) must be added to the time recorded in Bytes 7, 8, and 9.

The timing word is in milliseconds and has the following bit weight assignments:

Timing word								
Bit	0	1	2	3	4	5	6	7
Byte 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}
Byte 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
Byte 9	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}

The timing word LSB (2^{-8}) is equal to 1/256 msec, and the MSB (2^{15}) is equal to 32,768 msec. The timing word for each scan is equal to the elapsed time from zero time to the start of that scan. Timing words from 0 to 65535.9961 msec can be encoded. For longer recordings the timing word may overflow to zero and then continue.

The first scan of data has typically started with timing word zero. However, this is not a requirement. In a sampling system, it is not always practical to resynchronize the system even though most seismic data acquisition systems have to date. Possible reasons for not wanting to resynchronize could be digital filtering, communication restrictions, etc.

Whether the system is resynchronized or not, the timing word will contain the time from the energy source event to the start of scan of interest. For example, assume the sampling interval is 2 msec, the system does not resynchronize, and the energy source event occurs 1 + 9/256 msec before the next normal start of scan. The timing word values would be:

First timing word	0 + 1 + 9/256 msec
Second	2 + 1 + 9/256 msec
Third	4 + 1 + 9/256 msec
Fourth	6 + 1 + 9/256 msec
…	…
One-thousandth timing word	1998 + 1 + 9/256 msec

Byte 11 contains sample skew of the first sample of this trace. This is identical to the first byte of sample skew for this channel in the Scan Type Header.

Bytes 13, 14, 15 are included as an integrity check on time break. They comprise the timing word of the scan in which Time Break Window Indicator (TWI) changed to a one. Thus, it represents the time from the time break to the end of the time break window. Random variations in this time indicate a problem in the fire control system. The presence of a value less than the base scan interval indicates that time break was not detected and recording commenced at the end of the time break window.

The definition of the timing word is kept the same between Rev 2.1 and Rev 3.0, even though the possible sample rates have decreased to 1 microsec. The reason is that timing words are needed for historical reasons, and modern acquisition systems supporting high sample rate data are assumed to use synchronization techniques not requiring the use of timing word.

A Trace Header Extension may be added to include the receiver location for that trace. Receiver locations are defined by a receiver line number (three integer bytes and two fraction bytes), a receiver point number (three bytes integer and two bytes fraction) and a receiver point index (one byte). This index allows one to define the receiver group in the grid, the original value being 1 and that value is incremented by 1 every time the receiver is moved, even when it is moved back to the previous location. The reshoot index should be used to indicate a reacquisition of a previously acquired trace, counting from 0, an increasing with 1 every time the trace is re-acquired. The group index is used to indicate that the trace is part of a group that must be processed as a unit, e.g. rotation for multi-component data. The depth index is used to support sensors in different vertical positions. Please refer to appendix E.6 for examples of use of indexes. The Sensor Type (vertical geophone, hydrophone, etc.) may be indicated in Byte 21.

Additional trace header blocks (e.g. source information, positions, orientation blocks, or manufacturer defined blocks) may be added as needed by the manufacturer or user. The maximum number of Trace Header Extensions is limited to 255.

A larger number of samples per trace may be recorded using bytes 25–28 of the first Trace Header Extension.

5.4 Extended Header (optional)

The Extended Header provides additional areas to be used by equipment manufacturers to interface directly with their equipment. Since the nature of this data will depend heavily on the equipment and processes being applied, it will be the responsibility of the equipment manufacturer to establish a format and document this area. Byte 31 of General Header Block #1 contains the number of 32 byte fields in the Extended Header. If more than 99 Extended Header blocks are used, then General Header Block #1, byte 31 is set to FF_{16} and bytes 6 to 8 in the General Header Block #2 indicate the number of Extended Header Blocks.

5.5 External Header (optional)

The External Header provides a means of recording special user-desired information in the header block. This data format will be defined and documented by the end user. The means of putting this information into the header has usually been provided by the equipment manufacturer. Byte 32 of General Header Block #1 contains the number of 32 byte fields in the external header. If more than 99 External Header blocks are used, then General Header Block #1, Byte 32 is set to FF_{16} and Bytes 27 to 29 of General Header Block #2 indicates the number of External Header blocks.

5.6 General Trailer (optional)

Following the seismic data, a General Trailer may be recorded. This provides for recording auxiliary system and navigation related data. The addition of the trailer will allow the accumulation of system faults, data QC information, real-time navigation position, and timing information on the same record and contiguous with the shotpoint to which it relates. By recording this data after all of the other data, additional time is provided for collecting the data and transferring it to the recording system. The General Trailer consists of a set of blocks consisting of a short header followed by a number of data blocks. All information in the General Trailer is optional, and each block may be formatted as desired by the manufacturer or user. The number of General Trailer blocks is indicated in bytes 13 to 16 of General Header Block #2.

To aid with the transfer of information between acquisition and processing, a few types of trailer blocks have been defined:

1. Edit
2. Positions
3. Positions raw measurements

4. Text comments
5. Observer log

We recommended using these standard formats when inserting this type of information in the trailer.

Note: All types of trailer records are optional.

5.6.1 Edit (SEG-D Trace Edit v1.0)

Block Type (byte 1) is set to 01_{16}, ASCII/binary (byte 2) set to 02_{16} and Description (byte 9–24) set to `"TRACE EDIT v1.0 "`.

The edits trailer allows the recorder to modify the trace information provided by the trace header in the record. This is typically used to indicate problems with certain traces. Edits trailer blocks are typically generated post acquisition, during the QC and processing stages. SEG-D Rev 3 allows multiple Edit trailer blocks to be defined. The information in later Edit blocks overrides information in earlier blocks.

The format presented here is similar to the one described in *[SEG ADS Trace Edit, Norris, Hares, Faichney, 2001, SEG-UKOOA Ancillary Data Standard* **-** *ADS Trace Edit: Geophysics, 66, no. 06, 2040–2054]*, though some modifications/simplifications have been done. The SEG ADS Trace Edit format is very flexible, and some adaptation must be done to make it fit the SEG-D record structure. In addition, the format of some fields had to be more strictly defined to enable automatic parsing of the format.

It is recommended that the recording system creates Edit records to indicate which traces are bad, even though this is also indicated in the trace headers. This will give a quick overview of the shot without the need for scanning through all the data, and will simplify importing of data into processing/QC systems.

Multiple Edit records may exist. The latter will override prior definitions.

The edit format is ASCII, to ease human readability.

The format consists of multiple lines, each line terminated with a newline terminator consisting of a Line Feed ('\n', $0A_{16}$) or Carriage Return/LineFeed ("\r\n", $0D0A_{16}$).

From here on, <LF> is used to indicate newline, <space> to indicate the space character (20_{16}).

Each line (record) consists of a record description letter followed by space (20_{16}), then some ASCII text, and terminated by linefeed.
<Record function><space>ASCII text><LF>

The Edit block is padded with <space> until a multiple of 32 byte is reached.

Class	Record Function Character	Description
Header	V	SEG-D Trace Edit version (required)
Header	C	Comment
Header	S	Timestamp, UTC (required)
Header	W	System/person to generate this edit

Class	Record Function Character	Description
		(required)
Header	A	Attribute definition and range (at least one A record is required)
Header	P	Parameters for attribute, if one line is not enough to describe the attribute (optional)
Header	H	Header, description of a test (required)
Header	R	Range of traces tested. A test may not be applicable to all traces. If not present, all traces are tested. (optional)
Header	Y	Severity of exclusion. If not present, trace is to be removed (not to be processed) (Optional)
Primary Key Record	X	Exclude these trace ranges. May be several for each header.
Primary Key Record	I	Include these trace ranges. May be several for each header. I records may be used to override X record settings (though this is considered bad practice).
Terminator	E	End of Header/Primary Key record pair (at least one E record is required)
Terminator	T	End of SEG-D Trace Edit dataset (one and no more than one T record is required)

An edit record will consist of a header (V/S/W/C records), then for each test H/A/P/Y/R/X/I/E/C and optionally S/W if the test was performed in another system/at another time.

The format of all fields are defined by the recorder, except V/S/Y/R/X/I. However there are three requirements:

- *System recognition*: All systems must be able to determine which traces have been edited, when, by who, and how serious the edit is.
- *System regeneration*: The system creating the edit must be capable of parsing the edit, and recreate the test based on the information in the edit records. No other information must be required.
- *Human readability*: The edits should be human readable. Use C records if the system generated text is not enough.

Format of fields

SEG-D Trace Edit version

Format:

```
V<space>SEGD Trace Edit v<major>.<minor><LF>
```

This version of trace edit is 1.0.

Example:

```
V SEGD Trace Edit v1.0
```

Time stamp

This is the time of the edit Timestamp in UTC, accuracy 1 second.

Format:

```
S<space><hour>:<minute>:<second> <monthday>-<month three letter
abbreviation>-<year 4 digits><LF>
```

Example:

```
S 13:35:00 23-MAR-2007
```

Severity of exclusion

What to do with the trace failing the test described in the A record.

```
Y<space><severity><LF>
```

The following severities exist:

- EXCLUDE - The traces listed in X record should be excluded from further processing (default)
- WARNING - The traces listed in X records are suspicious, and further investigation is recommended.

Example:

```
Y WARNING
Y EXCLUDE
```

Trace range definition

Used to define ranges of traces to test (R), include (I) or exclude (X).

```
Format:
<record letter><space><scanset NUMBER LIST;<channelset NUMBER LIST>;<trace
NUMBER LIST><LF>
```

NUMBER LIST is a comma separated list of numbers, or ranges. For example 1,4,10–14 means numbers 1, 4, and 10 through 14.

Example:

```
R 1-1,5-14,1-558
X 1-1,5-14,1-5
I 1-1,8-11,5-5
```

The ranges checked are traces 1–558 in channel sets 5–14 (e.g. the seismic channel sets for a 10 streamer vessel). X record states traces 1–5 on all streamers fails the test (e.g. due to noise), but I record overrides, and indicates channel 5 in channel set 8–11 passed.

See Appendix E.7 for a complete example of a Trace Edit v1.0 block.

5.6.2 Navigation data backup

Trailer block used as a backup of navigation related information. The contents is defined by the following table

Contents	Block type	ASCII/ binary	Description
P1 processed navigation data	**20_{16}**	**02_{16} (ASCII)**	**`“P1 FILE FORMAT  ”`**
P2 Raw marine positioning data	**21_{16}**	**02_{16} (ASCII)**	**`“P2 FILE FORMAT  ”`**
P3 Land survey field data	**22_{16}**	**02_{16} (ASCII)**	**`“P3 File Format  ”`**
P5 Pipeline positioning data	**23_{16}**	**02_{16} (ASCII)**	**`“P5 FILE FORMAT  ”`**
P6 3D seismic binning grids	**24_{16}**	**02_{16} (ASCII)**	**`“P6 FILE FORMAT  ”`**
P7 Well deviation data	**25_{16}**	**02_{16} (ASCII)**	**`“P7 FILE FORMAT  ”`**
SPS Shell Processing Support data	**26_{16}**	**02_{16} (ASCII)**	**`“SPS FILE FORMAT ”`**
SEG-Y Extended Textual Stanza	**27_{16}**	**02_{16} (ASCII)**	**`“SEGY TEXT STANZA”`**

A whole file is added in a trailer block as described above. If multiple files need backing up, several trailer blocks may be added.

5.6.3 Text Comment

Block Type (byte 1) is set to 10_{16}, ASCII/binary (byte 2) set to 02_{16} and Description (bytes 9–24) set to `"TEXT COMMENT    "`.

This trailer block contains any user defined, textual comment belonging to the SEG-D record. May be multiple lines, each line terminated with Line Feed ('\n', $0A_{16}$), or optionally Carriage Return/LineFeed ("\r\n", $0D0A_{16}$).

Multiple text comments may exist.

Text comments are padded with spaces (20_{16}) until a multiple of 32 bytes is reached.

5.6.4 Observer log

Block Type (byte 1) is set to 11_{16}, ASCII/binary (byte 2) set to 02_{16} and Description (bytes 9–24) set to `"OBSERVER LOG    "`.

This is the observer log entry for this record. The format is textual.

Observer logs are padded with spaces (20_{16}) until a multiple of 32 bytes is reached.

5.6.5 User defined

The contents of the user defined block is completely left to the recording system, however the size must be a multiple of 32 bytes. The format is textual or binary as defined by the ASCII/binary flag in the description block.

6.0 Data Body

Data is recorded as a byte stream in demultiplexed format. Preceding each trace of data is a trace header and optional trace header extensions. Each trace is a sequential set of samples from one channel in one channel set.

6.1 Data Recording Method

To accommodate diverse recording needs, the data recording allows sample representations of 8, 16, 20, 24, 32, and 64 bits.

The data word is a numeric representation of the sign and magnitude of the instantaneous voltage presented to the system. It is not an indication of how the hardware gain system functions. The output of stepped gain systems may be represented as a binary mantissa and a binary exponent of base 2, 4, or 16 (binary, quaternary, or hexadecimal system) or a simple integer value. Depending upon the particular recording method, the mantissa or integer may be represented as a sign-magnitude, one's complement, or two's complement binary value. In a sign-magnitude representation, the initial bit is set to 1 if the value is the negative of the number represented by the remaining bits and 0 otherwise. A one's complement binary value is the same as sign-magnitude for nonnegative numbers but all the bits are flipped (XOR'ed with 1's) to represent negative values. A nonnegative value in two's complement representation is also the same as sign-magnitude, but negative numbers are represented by first flipping all the bits and then adding 1. If a two's complement number has n bits, then its value can be determined by first adding 2^{n-1}, taking the result modulo 2^n, and finally subtracting 2^{n-1}.

Following are descriptions of each of the data recording methods permitted. The same number system is to be used on all samples in a record, including auxiliary and all other types of channels. All recording methods are valid for demultiplexed records; as of Rev 1 SEG-D no longer supports multiplexed data.

1 byte quaternary exponent data recording method

The following illustrates the 8 bit word and the corresponding bit weights:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_2	C_1	C_0	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}

S=sign bit. —(One = negative number).

C=quaternary exponent.—This is a three bit positive binary exponent of 4 written as 4^{CCC} where CCC can assume values from 0–7.

Q_{-1-4} fraction.—This is a 4 bit one's complement binary fraction. The radix point is to the left of the most significant bit (Q_{-1}) with the MSB being defined as 2^{-1}. The fraction can have values from $-1+2^{-4}$ to $1-2^{-4}$.

Input signal = S.QQQQ × 4^{CCC} × DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header.

2 byte quaternary exponent data recording method

The following illustrates the 16-bit word and the corresponding bit weights:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_2	C_1	C_0	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}
Byte 2	Q_{-5}	Q_{-6}	Q_{-7}	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}

S=sign bit.—(One = negative number).

C=quaternary exponent.—This is a three bit positive binary exponent of 4 written as 4^{CCC} where CCC can assume values from 0–7.

$Q_{-1\text{-}12}$ fraction.—This is a 12 bit one's complement binary fraction. The radix point is to the left of the most significant bit (Q_{-1}) with the MSB being defined as 2^{-1}. The fraction can have values from $-1 + 2^{-12}$ to $1 - 2^{-12}$.

Input signal = S.QQQQ,QQQQ,QQQQ × 4^{CCC} × DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header.

2½ byte binary exponent data recording method—demultiplexed

The following illustrates the 20 bit word and the corresponding bit weights:

Bit	0	1	2	3	4	5	6	7	
Byte 1	C_3	C_2	C_1	C_0	C_3	C_2	C_1	C_0	Exponents for
Byte 2	C_3	C_2	C_1	C_0	C_3	C_2	C_1	C_0	samples 1 thru 4†
Byte 3	S	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}	Q_{-6}	Q_{-7}	Sample 1
Byte 4	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}	Q_{-14}	Q_{-15}	
Byte 5	S	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}	Q_{-6}	Q_{-7}	Sample 2
Byte 6	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}	Q_{-14}	Q_{-15}	
Byte 7	S	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}	Q_{-6}	Q_{-7}	Sample 3
Byte 8	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}	Q_{-14}	Q_{-15}	
Byte 9	S	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}	Q_{-6}	Q_{-7}	Sample 4
Byte 10	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}	Q_{-14}	Q_{-15}	

†Bytes 1 and 2 contain the exponents for the following four samples of the channel. The sample numbers are relative and are only to denote position in the four sample group.

S=sign bit—(One = negative number).

C=binary exponent—This is a 4 bit positive binary exponent of 2 written as 2^{CCCC} where CCCC can assume values of 0–15. The four exponents are in sample order for the four samples starting with the first sample in bits 0–3 of Byte 1.

$Q_{-1\text{-}15}$ fraction—This is a 15 bit one's complement binary fraction. The radix point is to the left of the most significant bit (Q_{-1}) with the MSB being defined as 2^{-1}. The sign and fraction can assume values from $1 - 2^{-15}$ to $-1 + 2^{-15}$.

Input signal=S.QQQ,QQQQ,QQQQ,QQQQ × 2^{CCCC} × DSM millivolts where DSM is the value required to descale the data word to the recording system input level. DSM is defined in Bytes 17–20 of each of the corresponding channel set descriptors in the scan type header.

Note that in utilizing this data recording method, the number of samples per channel must be exactly divisible by 4 in order to preserve the data grouping of this method.

1 byte hexadecimal exponent data - recording method

The following illustrates the 8-bit word and the corresponding bit weights:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_1	C_0	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}

S=sign bit.—(One = negative number).

C=hexadecimal exponent.—This is a two positive binary exponent of 16 written as 16^{CC} where CC can assume values from 0–3.

$Q_{-1\text{-}5}$ fraction—This is a 5 bit positive binary fraction. The radix point is to the left of the most significant bit (Q_{-1}) with the MSB being defined as 2^{-1}. The sign and fraction can have any value from $-1 + 2^{-5}$ to $1 - 2^{-5}$.

Input signal= $(-1)^S \times 0.Q,QQQQ \times 16^{CC} \times$ DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header.

2 byte hexadecimal exponent data - recording method

The following illustrates the 16-bit word and the corresponding bit weights:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_1	C_0	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}
Byte 2	Q_{-6}	Q_{-7}	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}

S=sign bit.—(One = negative number).

C=hexadecimal exponent.—This is a two bit positive binary exponent of 16^{CC} where CC can assume values from 0–3.

$Q_{-1\text{-}13}$ fraction—This is a 13 bit positive binary fraction. The radix point is to the left of the most significant bit (Q_{-1}) with the MSB being defined as 2^{-1}. The sign and fraction can have any value from $-1 + 2^{-13}$ to $1 - 2^{-13}$.

Input signal = $(-1)^S \times 0.Q,QQQQ,QQQQ,QQQQ \times 16^{CC} \times$ DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header.

4 byte hexadecimal exponent data - recording method

The following illustrates the 32-bit word and the corresponding bit weights:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_6	C_5	C_4	C_3	C_2	C_1	C_0
Byte 2	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}	Q_{-6}	Q_{-7}	Q_{-8}
Byte 3	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}	Q_{-14}	Q_{-15}	Q_{-16}
Byte 4	Q_{-17}	Q_{-18}	Q_{-19}	Q_{-20}	Q_{-21}	Q_{-22}	Q_{-23}	Q_{-24}

S=sign bit.—(One = negative number).

C=excess 64 hexadecimal exponent.—This is a binary exponent of 16. It has been biased by 64 such that it represents $16^{(CCCCCCC-64)}$ where CCCCCCC can assume values from 0 to 127.

$Q_{-1\text{-}24}$ magnitude fraction.—This is a 24 bit positive binary fraction (i.e., the number system is sign and magnitude). The radix point is to the left of the most significant bit (Q_{-1}) with the MSB being defined as 2^{-1}. The sign and fraction can assume values from ($1 - 2^{-24}$ to $-1 + 2^{-24}$). It must always be written as a hexadecimal left justified number. If this fraction is zero, the sign and exponent must also be zero (i.e., the entire word is zero.)

Input signal = $(-1)^S \times 0.QQQQ,QQQQ,QQQQ,QQQQ,QQQQ,QQQQ \times 16^{(CCCCCCC-64)}$ × DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header. This data recording method has more than sufficient range to handle the dynamic range of a typical seismic system. Thus, DSM may not be needed to account for any scaling and may be recorded as 1.0.

4 byte IEEE floating point data - recording method

The IEEE (Institute of Electrical and Electronics Engineers) format is fully documented in the IEEE standard, "ANSI/IEEE Std 754 - l985", available from the IEEE (http://ieeexplore.ieee.org/servlet/opac?punumber=2355).

The IEEE format is summarized as follows:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_7	C_6	C_5	C_4	C_3	C_2	C_1
Byte 2	C_0	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}	Q_{-5}	Q_{-6}	Q_{-7}
Byte 3	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}	Q_{-13}	Q_{-14}	Q_{-15}
Byte 4	Q_{-16}	Q_{-17}	Q_{-18}	Q_{-19}	Q_{-20}	Q_{-21}	Q_{-22}	Q_{-23}

The value (v) of a floating-point number represented in this format is determined as follows:

if $e = 255$ & $f \neq 0$. .$v = \text{NaN}$	Not-a-Number (see Note 1)
if $e = 255$ & $f = 0$. .$v = (-1)^s \times \infty$	Overflow
if $0 < e < 255$. . . . .$v = (-1)^s \times 2^{e-127} \times (1.f)$	Normalized
if $e = 0$ & $f \neq 0$. . . .$v = (-1)^s \times 2^{e-126} \times (0.f)$	Denormalized
if $e = 0$ & $f = 0$. . . .$v = (-1)^s \times 0$	± zero

where e = binary value of all C's (exponent)
f = binary value of all Q's (fraction)

NOTE 1. A Not-a-Number (NaN) is interpreted as an invalid number. All other numbers are valid and interpreted as described above.

Input signal = v × DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header. This data recording method has more than sufficient range to handle the dynamic range of a typical seismic system. Thus, DSM may not be needed to account for any scaling and may be recorded as 1.0.

8 byte IEEE floating point data – recording method

The IEEE (Institute of Electrical and Electronics Engineers) format is fully documented in the IEEE standard, "ANSI/IEEE Std 754 - l985", available from the IEEE (http://ieeexplore.ieee.org/servlet/opac?punumber=2355).

The IEEE format is summarized as follows:

Bit	0	1	2	3	4	5	6	7
Byte 1	S	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5
Byte 2	C_4	C_3	C_2	C_1	Q_{-1}	Q_{-2}	Q_{-3}	Q_{-4}
Byte 3	Q_{-5}	Q_{-6}	Q_{-7}	Q_{-8}	Q_{-9}	Q_{-10}	Q_{-11}	Q_{-12}
Byte 4	Q_{-13}	Q_{-14}	Q_{-15}	Q_{-16}	Q_{-17}	Q_{-18}	Q_{-19}	Q_{-20}
Byte 5	Q_{-21}	Q_{-22}	Q_{-23}	Q_{-24}	Q_{-25}	Q_{-26}	Q_{27}	Q_{-28}
Byte 6	Q_{-29}	Q_{-30}	Q_{-31}	Q_{-32}	Q_{-33}	Q_{-34}	Q_{-35}	Q_{-36}
Byte 7	Q_{-37}	Q_{-38}	Q_{-39}	Q_{-40}	Q_{-41}	Q_{-42}	Q_{-43}	Q_{-44}
Byte 8	Q_{-45}	Q_{-46}	Q_{-47}	Q_{-48}	Q_{-49}	Q_{-50}	Q_{-51}	Q_{-52}

The value (v) of a floating-point number represented in this format is determined as follows:

if $e = 2047$ & $f \neq 0$. .v = NaN	Not-a-Number (see Note 1)
if $e = 2047$ & $f = 0$. .$v = (-1)^s \times \infty$	Overflow (infinity)
if $0 < e < 2047$. . . . .$v = (-1)^s \times 2^{e-1023} \times (1.f)$	Normalized
if $e = 0$ & $f \neq 0$. . . . .$v = (-1)^s \times 2^{e-1022} \times (0.f)$	Denormalized
if $e = 0$ & $f = 0$. . . . .$v = (-1)^s \times 0$	± zero

where e = binary value of all C's (exponent)
f = binary value of all Q's (fraction)

NOTE 1. A Not-a-Number (NaN) is interpreted as an invalid number. All other numbers are valid and interpreted as described above.

Input signal = v × DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header. This data recording method has more than sufficient range to handle the dynamic range of a typical seismic system. Thus, DSM may not be needed to account for any scaling and may be recorded as 1.0.

Integer formats:

24 bit two’s complement format:

Bit	0	1	2	3	4	5	6	7
Byte 1	I_{23}	I_{22}	I_{21}	I_{20}	I_{19}	I_{18}	I_{17}	I_{16}

Byte 2	I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8
Byte 3	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Input signal = { (IIII,IIII,IIII,IIII,IIII,IIII + 2^{23}) mod $2^{24} - 2^{23}$ } × DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header.

32 bit two's complement format:

Bit	0	1	2	3	4	5	6	7
Byte 1	I_{31}	I_{30}	I_{29}	I_{28}	I_{27}	I_{26}	I_{25}	I_{24}
Byte 2	I_{23}	I_{22}	I_{21}	I_{20}	I_{19}	I_{18}	I_{17}	I_{16}
Byte 3	I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8
Byte 4	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Input signal = { (IIII,IIII,IIII,IIII,IIII,IIII,IIII,IIII + 2^{31}) mod $2^{32} - 2^{31}$ } × DSM millivolts where DSM is the value required to descale the data sample to the recording system input level. DSM is defined in Bytes 17–20 of each channel set descriptor in the scan type header.

6.2 DSM Factor calculation and physical unit

The Descale multiplier (DSM) parameter is provided to allow the dimensionless numbers recorded on tape to be "descaled" back to the instantaneous sample values in millivolts at the system inputs. As of Rev 3, DSM is stored in bytes 17–20 of the Channel Set Descriptor in the Scan Type Header as a 4 byte IEEE floating point number. Prior to that it was computed from an MP exponent stored in byte 8 of the Channel Set Descriptor.

In general, recording systems scale the input signal level in order to match the useful range of input levels to the gain-ranging amplifier. DSM must account for all scaling in the acquisition system. However, in some high resolution formats like the 4 byte hexadecimal, or 4 or 8 byte IEEE floating point cases, the data recording method may have sufficient range, and the DSM may be set to 1.0.

<u>Physical unit conversion</u>

To be able to relate the seismic traces to external events it may be needed to convert the millivolt values in the traces to the physical unit. The physical unit of the traces is listed in byte 62 of the Channel Set Descriptor. This is repeated in byte 31 of Trace Header Extension block. The sensor sensitivity (Sensor Info Trace Header Extension bytes 9–12, 4 byte IEEE float) is multiplied with the millivolt sample value to calculate the physical unit number.

The following equation converts a trace sample value to a physical unit sample value

$$V_p = V_t * DSM * SS$$

V_p - Physical sample value

V_t - Trace sample value

DSM - Multiplication factor to convert to millivolt value

SS - Sensor sensitivity

The type of the physical unit are listed in the table found Chapter 8, Trace Header Extension, byte 31 definition. Typical types are millibar (e.g. hydrophone data), millimeters/second (e.g. geophone data) or meters (e.g. depth measurement).

DSM may be written

$$DSM = 2^{MP}$$

The calculation of MP for a data recording method is given by one of the following equations:

For floating point data:

1. $MP = FS - PA - C_{max}$	For binary exponents.
2. $MP = FS - PA - 2 \times C_{max}$	For quaternary exponents.
3. $MP = FS - PA - 4 \times C_{max}$	For hexadecimal exponents (except for the four byte excess 64 method).
4. $MP = FS - PA - 4 \times (C_{max} - 64)$	For excess 64 hexadecimal exponents.
5. $MP = FS - PA - (C_{max} - 127)$	For 32 bit IEEE exponents.
6. $MP = FS - PA - (C_{max} - 1023)$	For 64 bit IEEE exponents.

Where:

2^{FS} = Converter full scale (millivolts),
2^{PA} = Minimum system gain,
C_{max} = maximum value of the data exponent;

C_{max} =

15 for binary exponents
7 for quaternary exponents,
3 for hexadecimal exponents, except excess 64,
127 for excess 64 exponents and the output of the analog-to-digital converter is written as the fractional portion of the data value.
255 for 32 bit IEEE exponents and the output of the analog-to-digital converter is written as the fractional portion of the data value.
2047 for 64 bit IEEE exponents and the output of the analog-to-digital converter is written as the fractional portion of the data value.

For integer data:

1. $MP = FS - PA - IS$;

Where:

2^{FS} = Converter full scale (millivolts),
2^{PA} = Minimum system gain,
$2^{IS}-1$ = Integer number system positive full scale, and

the output of the analog-to-digital converter is written as an integer.

The term “minimum system gain” includes preamplifier gain and the minimum floating point amplifier gain. For example, one system may use a preamplifier gain of 256 and a minimum floating point amplifier gain of one. The minimum system gain is 256 x 1 = 2^8, so PA = 8. Another system may use a preamplifier gain of 320 and a minimum floating point amplifier gain of 0.8. In this case, the minimum system gain is 320 x 0.8 = 256 or 2^8. Again PA = 8.

PA may also account for any amplification needed to accommodate an analog to digital converter with a full scale value that is not a power of 2 in millivolts. For example, a 10 V (10,000 mV) converter may be preceded by an amplifier with a gain of 1.221 (10,000/8,192). This gain may be accounted for in PA. Alternatively, it could be considered part of the converter, making it appear to have a binary full scale.

Justifications for the Equations

The output of the analog-to-digital converter is written as the fractional portion of the data value. This is equivalent to dividing the value by the full scale of the converter. In order to compensate for this, the data value recorded on tape must be multiplied by the full scale value of the converter (2^{FS}). Thus FS appears in equations (1)–(4) with a positive sign.

The input signal was multiplied by the minimum system gain (2^{PA}) which, as mentioned, includes any preamplifier gain, minimum floating point amplifier gain, or analog-to-digital converter adjustment gain. The data recorded on tape must be divided by this minimum system gain; thus, PA appears in the equations with a negative sign.

Large input signals converted at minimum floating point amplifier gain are written on tape with the maximum exponent for the data recording method used. Likewise, small signals converted at full gain are written with the maximum exponent. The data as written have been multiplied by the exponent base raised to C_{max} (minus 64, 127, or 1023 as appropriate to the number format.) Thus C_{max} appears in the equations with a negative sign. MP is a power of 2 so the quaternary and hexadecimal C_{max} values are multiplied by 2 and 4, respectively ($4^c = 2^{2c}$ and $16^c = 2^{4c}$).

Data processing done prior to data storage.

In modern recording systems significant data processing is commonly done prior to storing the data on tape. The DSM factor may have to be adapted to make sure the number resolution is maintained in the trace data, i.e. the processing scaling parameters must be incorporated in the DSM factor.

6.3 Sensor calibration

Trace Header frequency domain calibration

SEGD supports storing individual sensor calibration values in Trace Header Extension blocks (see Sensor Calibration Header)". Each 32 byte Sensor Calibration Header block contains two frequency/amplitude/phase groups, and multiple blocks may be used to define the complete frequency domain calibration function for the sensor. Sensor Calibration Header block allows a maximum of about 500 frequencies per trace, assuming no other meta-information exist in the trace header

The calibration corrected sensor value is achieved by converting the time series trace value to frequency domain, and multiplying with the calibration function

$$V_{calibrated} = F_c \cdot V_{trace}$$

Where

$V_{calibrated}$ = Calibration corrected trace signal (in frequency domain)
F_c = Sensor calibration transfer function as described by the values in the sensor calibration header.
V_{trace} = The uncorrected trace value (in frequency domain)

The sensor calibration values in addition to the individual Sensor Sensitivity values can be used to correct for most sensor variations needed to achieve a high fidelity signal needed in modern seismic acquisition systems.

Calibration Channel – time domain

A Calibration Channel Set can be used if time series calibration is preferred. The Calibration Channel Set also allows a more high fidelity calibration compared to the Trace Header Extension blocks. The frequency and length of the calibration channel may be adjusted to achieve the accuracy required. Calibration Channel Sets has Channel type set to $F0_{16}$ in

Channel Set Descriptor byte 4. The Calibration Channel is a time series signal that can be correlated with the trace value to achieve a calibrated trace signal. By converting both the trace and calibration trace time series to frequency domain, the formula shown above can be used. The indices in the Trace Header Extension block indicate which trace the calibration trace applies to.

7.0 Header Tables

Note:
Several blocks contain undefined fields, indicated by X in the tables. These values are undefined by the format, and may be used by the manufacturer to store manufacturer defined information. However, <u>be aware that the fields may be used in future versions of the format</u>, and the manufacturer use these fields at own risk. It is recommended to record '0' in fields indicated by X, and store all manufacturer defined information in user defined header blocks (External or Extended header, General Trailer blocks, or General Header/Trace Header blocks with ID $B0_{16}$ through FF_{16}). These will more likely to be compliant with future versions of the format.

7.1 GENERAL HEADER BLOCK #1

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
File Number	F_1	F_1	F_1	F_1	F_2	F_2	F_2	F_2	1
	F_3	F_3	F_3	F_3	F_4	F_4	F_4	F_4	2
Format Code	Y_1	Y_1	Y_1	Y_1	Y_2	Y_2	Y_2	Y_2	3
	Y_3	Y_3	Y_3	Y_3	Y_4	Y_4	Y_4	Y_4	4
General Constants	K_1	K_1	K_1	K_1	K_2	K_2	K_2	K_2	5
	K_3	K_3	K_3	K_3	K_4	K_4	K	K_4	6
	K_5	K_5	K_5	K_5	K_6	K_6	K_6	K_6	7
	K_7	K_7	K_7	K_7	K_8	K_8	K_8	K_8	8
	K_9	K_9	K_9	K_9	K_{10}	K_{10}	K_{10}	K_{10}	9
	K_{11}	K_{11}	K_{11}	K_{11}	K_{12}	K_{12}	K_{12}	K_{12}	10
Year	YR_1	YR_1	YR_1	YR_1	YR_2	YR_2	YR_2	YR_2	11
# Additional Blks in Gen Hdr	GH_3	GH_2	GH_1	GH_0	DY_1	DY_1	DY_1	DY_1	12
Day (DY)	DY_2	DY_2	DY_2	DY_2	DY_3	DY_3	DY_3	DY_3	13
Hour	H_1	H_1	H_1	H_1	H_2	H_2	H_2	H_2	14
Minute	MI_1	MI_1	MI_1	MI_1	MI_2	MI_2	MI_2	MI_2	15
Second	SE_1	SE_1	SE_1	SE_1	SE_2	SE_2	SE_2	SE_2	16
Manufacture's Code	M_1	M_1	M_1	M_1	M_2	M_2	M_2	M_2	17
	M_3	M_3	M_3	M_3	M_4	M_4	M_4	M_4	18
	M_5	M_5	M_5	M_5	M_6	M_6	M_6	M_6	19
	0	0	0	0	0	0	0	0	20
	0	0	0	0	0	0	0	0	21
	0	0	0	0	0	0	0	0	22
Base Scan Interval	I_3	I_2	I_1	I_0	I_{-1}	I_{-2}	I_{-3}	I_{-4}	23
Polarity (P)	P	P	P	P	0	0	0	0	24
	0	0	0	0	0	0	0	0	25
Record Type (Z)	Z	Z	Z	Z	R_1	R_1	R_1	R_1	26
Record Length (R)	R_2	R_2	R_2	R_2	R_3	R_3	R_3	R_3	27
Scan Types/Record	ST/R_1	ST/R_1	ST/R_1	ST/R_1	ST/R_2	ST/R_2	ST/R_2	ST/R_2	28
Chan Sets/Scan Type	CS_1	CS_1	CS_1	CS_1	CS_2	CS_2	CS_2	CS_2	29
Skew Blocks	SK_1	SK_1	SK_1	SK_1	SK_2	SK_2	SK_2	SK_2	30
Extended Header Blk	EC_1	EC_1	EC_1	EC_1	EC_2	EC_2	EC_2	EC_2	31
External Header Blk	EX_1	EX_1	EX_1	EX_1	EX_2	EX_2	EX_2	EX_2	32

7.2 GENERAL HEADER BLOCK #2

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Expanded File Number	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}	1
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_9	EF_8	2
	EF_7	EF_6	EF_5	EF_4	EF_3	EF_2	EF_1	EF_0	3
Extended Channel Sets/ Scan Type	EN_{15}	EN_{14}	EN_{13}	EN_{12}	EN_{11}	EN_{10}	EN_9	EN_8	4
	EN_7	EN_6	EN_5	EN_4	EN_3	EN_2	EN_1	EN_0	5
Extended Header Blks	ECX_{23}	ECX_{22}	ECX_{21}	ECX_{20}	ECX_{19}	ECX_{18}	ECX_{17}	ECX_{16}	6
	ECX_{15}	ECX_{14}	ECX_{13}	ECX_{12}	ECX_{11}	ECX_{10}	ECX_9	ECX_8	7
	ECX_7	ECX_6	ECX_5	ECX_4	ECX_3	ECX_2	ECX_1	ECX_0	8
Extended Skew blks	ESK_{15}	ESK_{14}	ESK_{13}	ESK_{12}	ESK_{11}	ESK_{10}	ESK_9	ESK_8	9
	ESK_7	ESK_6	ESK_5	ESK_4	ESK_3	ESK_2	ESK_1	ESK_0	10
SEG-D Revision No.	RMJ_7	RMJ_6	RMJ_5	RMJ_4	RMJ_3	RMJ_2	RMJ_1	RMJ_0	11
“major” and “minor”	RMN_7	RMN_6	RMN_5	RMN_4	RMN_3	RMN_2	RMN_1	RMN_0	12
General Trailer, Number of Blks	GT_{31}	GT_{30}	GT_{29}	GT_{28}	GT_{27}	GT_{26}	GT_{25}	GT_{24}	13
	GT_{23}	GT_{22}	GT_{21}	GT_{20}	GT_{19}	GT_{18}	GT_{17}	GT_{16}	14
	GT_{15}	GT_{14}	GT_{13}	GT_{12}	GT_{11}	GT_{10}	GT_9	GT_8	15
	GT_7	GT_6	GT_5	GT_4	GT_3	GT_2	GT_1	GT_0	16
Extended Record Length	ERL_{31}	ERL_{30}	ERL_{29}	ERL_{28}	ERL_{27}	ERL_{26}	ERL_{25}	ERL_{24}	17
	ERL_{23}	ERL_{22}	ERL_{21}	ERL_{20}	ERL_{19}	ERL_{18}	ERL_{17}	ERL_{16}	18
	ERL_{15}	ERL_{14}	ERL_{13}	ERL_{12}	ERL_{11}	ERL_{10}	ERL_9	ERL_8	19
	ERL_7	ERL_6	ERL_5	ERL_4	ERL_3	ERL_2	ERL_1	ERL_0	20
Record Set Number	SN_{15}	SN_{14}	SN_{13}	SN_{12}	SN_{11}	SN_{10}	SN_9	SN_8	21
	SN_7	SN_6	SN_5	SN_4	SN_3	SN_2	SN_1	SN_0	22
Extended # Additional Blks in Gen Hdr	EGH_{15}	EGH_{14}	EGH_{13}	EGH_{12}	EGH_{11}	EGH_{10}	EGH_9	EGH_8	23
	EGH_7	EGH_6	EGH_5	EGH_4	EGH_3	EGH_2	EGH_1	EGH_0	24
Dominant Sampling Interval	BSI_{23}	BSI_{22}	BSI_{21}	BSI_{20}	BSI_{19}	BSI_{18}	BSI_{17}	BSI_{16}	25
	BSI_{15}	BSI_{14}	BSI_{13}	BSI_{12}	BSI_{11}	BSI_{10}	BSI_9	BSI_8	26
	BSI_7	BSI_6	BSI_5	BSI_4	BSI_3	BSI_2	BSI_1	BSI_0	27
External Header Blks	EH_{23}	EH_{22}	EH_{21}	EH_{20}	EH_{19}	EH_{18}	EH_{17}	EH_{16}	28
	EH_{15}	EH_{14}	EH_{13}	EH_{12}	EH_{11}	EH_{10}	EH_9	EH_8	29
	EH_7	EH_6	EH_5	EH_4	EH_3	EH_2	EH_1	EH_0	30
	X	X	X	X	X	X	X	X	31
Header Block Type	0	0	0	0	0	0	1	0	32

7.3 GENERAL HEADER BLOCK #3 (Timestamp and size header)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Time Zero	TZ_{63}	TZ_{62}	TZ_{61}	TZ_{60}	TZ_{59}	TZ_{58}	TZ_{57}	TZ_{56}	1
	TZ_{55}	TZ_{54}	TZ_{53}	TZ_{52}	TZ_{51}	TZ_{50}	TZ_{49}	TZ_{48}	2
	TZ_{47}	TZ_{46}	TZ_{45}	TZ_{44}	TZ_{43}	TZ_{42}	TZ_{41}	TZ_{40}	3
	TZ_{39}	TZ_{38}	TZ_{37}	TZ_{36}	TZ_{35}	TZ_{34}	TZ_{33}	TZ_{32}	4
	TZ_{31}	TZ_{30}	TZ_{29}	TZ_{28}	TZ_{27}	TZ_{26}	TZ_{25}	TZ_{24}	5
	TZ_{23}	TZ_{22}	TZ_{21}	TZ_{20}	TZ_{19}	TZ_{18}	TZ_{17}	TZ_{16}	6
	TZ_{15}	TZ_{14}	TZ_{13}	TZ_{12}	TZ_{11}	TZ_{10}	TZ_{9}	TZ_{8}	7
	TZ_{7}	TZ_{6}	TZ_{5}	TZ_{4}	TZ_{3}	TZ_{2}	TZ_{1}	TZ_{0}	8
Record Size	RS_{63}	RS_{62}	RS_{61}	RS_{60}	RS_{59}	RS_{58}	RS_{57}	RS_{56}	9
	RS_{55}	RS_{54}	RS_{53}	RS_{52}	RS_{51}	RS_{50}	RS_{49}	RS_{48}	10
	RS_{47}	RS_{46}	RS_{45}	RS_{44}	RS_{43}	RS_{42}	RS_{41}	RS_{40}	11
	RS_{39}	RS_{38}	RS_{37}	RS_{36}	RS_{35}	RS_{34}	RS_{33}	RS_{32}	12
	RS_{31}	RS_{30}	RS_{29}	RS_{28}	RS_{27}	RS_{26}	RS_{25}	RS_{24}	13
	RS_{23}	RS_{22}	RS_{21}	RS_{20}	RS_{19}	RS_{18}	RS_{17}	RS_{16}	14
	RS_{15}	RS_{14}	RS_{13}	RS_{12}	RS_{11}	RS_{10}	RS_{9}	RS_{8}	15
	RS_{7}	RS_{6}	RS_{5}	RS_{4}	RS_{3}	RS_{2}	RS_{1}	RS_{0}	16
Data Size	DS_{63}	DS_{62}	DS_{61}	DS_{60}	DS_{59}	DS_{58}	DS_{57}	DS_{56}	17
	DS_{55}	DS_{54}	DS_{53}	DS_{52}	DS_{51}	DS_{50}	DS_{49}	DS_{48}	18
	DS_{47}	DS_{46}	DS_{45}	DS_{44}	DS_{43}	DS_{42}	DS_{41}	DS_{40}	19
	DS_{39}	DS_{38}	DS_{37}	DS_{36}	DS_{35}	DS_{34}	DS_{33}	DS_{32}	20
	DS_{31}	DS_{30}	DS_{29}	DS_{28}	DS_{27}	DS_{26}	DS_{25}	DS_{24}	21
	DS_{23}	DS_{22}	DS_{21}	DS_{20}	DS_{19}	DS_{18}	DS_{17}	DS_{16}	22
	DS_{15}	DS_{14}	DS_{13}	DS_{12}	DS_{11}	DS_{10}	DS_{9}	DS_{8}	23
	DS_{7}	DS_{6}	DS_{5}	DS_{4}	DS_{3}	DS_{2}	DS_{1}	DS_{0}	24
Header Size	HS_{31}	HS_{30}	HS_{29}	HS_{28}	HS_{27}	HS_{26}	HS_{25}	HS_{24}	25
	HS_{23}	HS_{22}	HS_{21}	HS_{20}	HS_{19}	HS_{18}	HS_{17}	HS_{16}	26
	HS_{15}	HS_{14}	HS_{13}	HS_{12}	HS_{11}	HS_{10}	HS_{9}	HS_{8}	27
	HS_{7}	HS_{6}	HS_{5}	HS_{4}	HS_{3}	HS_{2}	HS_{1}	HS_{0}	28
Extd Rec Mode	ERM_{7}	ERM_{6}	ERM_{5}	ERM_{4}	ERM_{3}	ERM_{2}	ERM_{1}	ERM_{0}	29
Rel Time Mode	RTM_{7}	RTM_{6}	RTM_{5}	RTM_{4}	RTM_{3}	RTM_{2}	RTM_{1}	RTM_{0}	30
	X	X	X	X	X	X	X	X	31
Header Block Type	0	0	0	0	0	0	1	1	32

7.4 GENERAL HEADER BLOCK #4 (Vessel/Crew Identification) (optional)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Abbr Vessel or Crew Name	VCA	VCA	VCA	VCA	VCA	VCA	VCA	VCA	1
	VCA	VCA	VCA	VCA	VCA	VCA	VCA	VCA	2
	VCA	VCA	VCA	VCA	VCA	VCA	VCA	VCA	3
Vessel or Crew Name	VC	VC	VC	VC	VC	VC	VC	VC	4
	VC	VC	VC	VC	VC	VC	VC	VC	5

	VC	VC	VC	VC	VC	VC	VC	VC	6
	VC	VC	VC	VC	VC	VC	VC	VC	7
	VC	VC	VC	VC	VC	VC	VC	VC	8
	VC	VC	VC	VC	VC	VC	VC	VC	9
	VC	VC	VC	VC	VC	VC	VC	VC	10
	VC	VC	VC	VC	VC	VC	VC	VC	11
	VC	VC	VC	VC	VC	VC	VC	VC	12
	VC	VC	VC	VC	VC	VC	VC	VC	13
	VC	VC	VC	VC	VC	VC	VC	VC	14
	VC	VC	VC	VC	VC	VC	VC	VC	15
	VC	VC	VC	VC	VC	VC	VC	VC	16
	VC	VC	VC	VC	VC	VC	VC	VC	17
	VC	VC	VC	VC	VC	VC	VC	VC	18
	VC	VC	VC	VC	VC	VC	VC	VC	19
	VC	VC	VC	VC	VC	VC	VC	VC	20
	VC	VC	VC	VC	VC	VC	VC	VC	21
	VC	VC	VC	VC	VC	VC	VC	VC	22
	VC	VC	VC	VC	VC	VC	VC	VC	23
	VC	VC	VC	VC	VC	VC	VC	VC	24
	VC	VC	VC	VC	VC	VC	VC	VC	25
	VC	VC	VC	VC	VC	VC	VC	VC	26
	VC	VC	VC	VC	VC	VC	VC	VC	27
	VC	VC	VC	VC	VC	VC	VC	VC	28
	VC	VC	VC	VC	VC	VC	VC	VC	29
	VC	VC	VC	VC	VC	VC	VC	VC	30
	VC	VC	VC	VC	VC	VC	VC	VC	31
Header Block Type	0	0	0	1	0	0	0	0	32

7.5 GENERAL HEADER BLOCK #5 (Survey Area Name) (optional)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Survey Area Name	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	1
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	2
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	3
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	4
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	5
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	6
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	7
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	8
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	9
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	10
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	11
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	12
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	13
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	14
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	15

	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	16
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	17
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	18
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	19
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	20
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	21
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	22
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	23
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	24
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	25
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	26
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	27
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	28
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	29
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	30
	SAN	SAN	SAN	SAN	SAN	SAN	SAN	SAN	31
Header Block Type	0	0	0	1	0	0	0	1	32

7.6 GENERAL HEADER BLOCK #6 (Client Identification) (optional)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Survey Area Name	CI	CI	CI	CI	CI	CI	CI	CI	1
	CI	CI	CI	CI	CI	CI	CI	CI	2
	CI	CI	CI	CI	CI	CI	CI	CI	3
	CI	CI	CI	CI	CI	CI	CI	CI	4
	CI	CI	CI	CI	CI	CI	CI	CI	5
	CI	CI	CI	CI	CI	CI	CI	CI	6
	CI	CI	CI	CI	CI	CI	CI	CI	7
	CI	CI	CI	CI	CI	CI	CI	CI	8
	CI	CI	CI	CI	CI	CI	CI	CI	9
	CI	CI	CI	CI	CI	CI	CI	CI	10
	CI	CI	CI	CI	CI	CI	CI	CI	11
	CI	CI	CI	CI	CI	CI	CI	CI	12
	CI	CI	CI	CI	CI	CI	CI	CI	13
	CI	CI	CI	CI	CI	CI	CI	CI	14
	CI	CI	CI	CI	CI	CI	CI	CI	15
	CI	CI	CI	CI	CI	CI	CI	CI	16
	CI	CI	CI	CI	CI	CI	CI	CI	17
	CI	CI	CI	CI	CI	CI	CI	CI	18
	CI	CI	CI	CI	CI	CI	CI	CI	19
	CI	CI	CI	CI	CI	CI	CI	CI	20
	CI	CI	CI	CI	CI	CI	CI	CI	21
	CI	CI	CI	CI	CI	CI	CI	CI	22
	CI	CI	CI	CI	CI	CI	CI	CI	23
	CI	CI	CI	CI	CI	CI	CI	CI	24
	CI	CI	CI	CI	CI	CI	CI	CI	25

	CI	CI	CI	CI	CI	CI	CI	CI	26
	CI	CI	CI	CI	CI	CI	CI	CI	27
	CI	CI	CI	CI	CI	CI	CI	CI	28
	CI	CI	CI	CI	CI	CI	CI	CI	29
	CI	CI	CI	CI	CI	CI	CI	CI	30
	CI	CI	CI	CI	CI	CI	CI	CI	31
Header Block Type	0	0	0	1	0	0	1	0	32

7.7 GENERAL HEADER BLOCK #7 (Job identification) (optional)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Abbr Identification	JIA	JIA	JIA	JIA	JIA	JIA	JIA	JIA	1
	JIA	JIA	JIA	JIA	JIA	JIA	JIA	JIA	2
	JIA	JIA	JIA	JIA	JIA	JIA	JIA	JIA	3
	JIA	JIA	JIA	JIA	JIA	JIA	JIA	JIA	4
	JIA	JIA	JIA	JIA	JIA	JIA	JIA	JIA	5
Job Identification	JI	JI	JI	JI	JI	JI	JI	JI	6
	JI	JI	JI	JI	JI	JI	JI	JI	7
	JI	JI	JI	JI	JI	JI	JI	JI	8
	JI	JI	JI	JI	JI	JI	JI	JI	9
	JI	JI	JI	JI	JI	JI	JI	JI	10
	JI	JI	JI	JI	JI	JI	JI	JI	11
	JI	JI	JI	JI	JI	JI	JI	JI	12
	JI	JI	JI	JI	JI	JI	JI	JI	13
	JI	JI	JI	JI	JI	JI	JI	JI	14
	JI	JI	JI	JI	JI	JI	JI	JI	15
	JI	JI	JI	JI	JI	JI	JI	JI	16
	JI	JI	JI	JI	JI	JI	JI	JI	17
	JI	JI	JI	JI	JI	JI	JI	JI	18
	JI	JI	JI	JI	JI	JI	JI	JI	19
	JI	JI	JI	JI	JI	JI	JI	JI	20
	JI	JI	JI	JI	JI	JI	JI	JI	21
	JI	JI	JI	JI	JI	JI	JI	JI	22
	JI	JI	JI	JI	JI	JI	JI	JI	23
	JI	JI	JI	JI	JI	JI	JI	JI	24
	JI	JI	JI	JI	JI	JI	JI	JI	25
	JI	JI	JI	JI	JI	JI	JI	JI	26
	JI	JI	JI	JI	JI	JI	JI	JI	27
	JI	JI	JI	JI	JI	JI	JI	JI	28
	JI	JI	JI	JI	JI	JI	JI	JI	29
	JI	JI	JI	JI	JI	JI	JI	JI	30
	JI	JI	JI	JI	JI	JI	JI	JI	31
Header Block Type	0	0	0	1	0	0	1	1	32

7.8 GENERAL HEADER BLOCK #8 (Line identification) (optional)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Line Abbreviation	LA	LA	LA	LA	LA	LA	LA	LA	1
	LA	LA	LA	LA	LA	LA	LA	LA	2
	LA	LA	LA	LA	LA	LA	LA	LA	3
	LA	LA	LA	LA	LA	LA	LA	LA	4
	LA	LA	LA	LA	LA	LA	LA	LA	5
	LA	LA	LA	LA	LA	LA	LA	LA	6
	LA	LA	LA	LA	LA	LA	LA	LA	7
Line Identification	LI	LI	LI	LI	LI	LI	LI	LI	8
	LI	LI	LI	LI	LI	LI	LI	LI	9
	LI	LI	LI	LI	LI	LI	LI	LI	10
	LI	LI	LI	LI	LI	LI	LI	LI	11
	LI	LI	LI	LI	LI	LI	LI	LI	12
	LI	LI	LI	LI	LI	LI	LI	LI	13
	LI	LI	LI	LI	LI	LI	LI	LI	14
	LI	LI	LI	LI	LI	LI	LI	LI	15
	LI	LI	LI	LI	LI	LI	LI	LI	16
	LI	LI	LI	LI	LI	LI	LI	LI	17
	LI	LI	LI	LI	LI	LI	LI	LI	18
	LI	LI	LI	LI	LI	LI	LI	LI	19
	LI	LI	LI	LI	LI	LI	LI	LI	20
	LI	LI	LI	LI	LI	LI	LI	LI	21
	LI	LI	LI	LI	LI	LI	LI	LI	22
	LI	LI	LI	LI	LI	LI	LI	LI	23
	LI	LI	LI	LI	LI	LI	LI	LI	24
	LI	LI	LI	LI	LI	LI	LI	LI	25
	LI	LI	LI	LI	LI	LI	LI	LI	26
	LI	LI	LI	LI	LI	LI	LI	LI	27
	LI	LI	LI	LI	LI	LI	LI	LI	28
	LI	LI	LI	LI	LI	LI	LI	LI	29
	LI	LI	LI	LI	LI	LI	LI	LI	30
	LI	LI	LI	LI	LI	LI	LI	LI	31
Header Block Type	0	0	0	1	0	1	0	0	32

7.9 SOURCE DESCRIPTION BLOCK (optional)

7.9.1 VIBRATOR

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Expanded File No.	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}	1
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_{9}	EF_{8}	2
	EF_{7}	EF_{6}	EF_{5}	EF_{4}	EF_{3}	EF_{2}	EF_{1}	EF_{0}	3
Source Line No. (INTEGER)	SLN_{23}	SLN_{22}	SLN_{21}	SLN_{20}	SLN_{19}	SLN_{18}	SLN_{17}	SLN_{16}	4
	SLN_{15}	SLN_{14}	SLN_{13}	SLN_{12}	SLN_{11}	SLN_{10}	SLN_{9}	SLN_{8}	5
	SLN_{7}	SLN_{6}	SLN_{5}	SLN_{4}	SLN_{3}	SLN_{2}	SLN_{1}	SLN_{0}	6
Source Line No. (FRACTION)	SLN_{-1}	SLN_{-2}	SLN_{-3}	SLN_{-4}	SLN_{-5}	SLN_{-6}	SLN_{-7}	SLN_{-8}	7
	SLN_{-9}	SLN_{-10}	SLN_{-11}	SLN_{-12}	SLN_{-13}	SLN_{-14}	SLN_{-15}	SLN_{-16}	8
Source Point No. (INTEGER)	SPN_{23}	SPN_{22}	SPN_{21}	SPN_{20}	SPN_{19}	SPN_{18}	SPN_{17}	SPN_{16}	9
	SPN_{15}	SPN_{14}	SPN_{13}	SPN_{12}	SPN_{11}	SPN_{10}	SPN_{9}	SPN_{8}	10
	SPN_{7}	SPN_{6}	SPN_{5}	SPN_{4}	SPN_{3}	SPN_{2}	SPN_{1}	SPN_{0}	11
Source Point No. (FRACTION)	SPN_{-1}	SPN_{-2}	SPN_{-3}	SPN_{-4}	SPN_{-5}	SPN_{-6}	SPN_{-7}	SPN_{-8}	12
	SPN_{-9}	SPN_{-10}	SPN_{-11}	$SPN-_{12}$	$SPN-_{13}$	SPN_{-14}	SPN_{-15}	SPN_{-16}	13
Source Point Index	SPI_{7}	SPI_{6}	SPI_{5}	SPI_{4}	SPI_{3}	SPI_{2}	SPI_{1}	SPI_{0}	14
Phase Control	PC_{7}	PC_{6}	PC_{5}	PC_{4}	PC_{3}	PC_{2}	PC_{1}	PC_{0}	15
Type Vibrator	V_{7}	V_{6}	V_{5}	V_{4}	V_{3}	V_{2}	V_{1}	V_{0}	16
Phase Angle	PA_{15}	PA_{14}	PA_{13}	PA_{12}	PA_{11}	PA_{10}	PA_{9}	PA_{8}	17
	PA_{7}	PA_{6}	PA_{5}	PA_{4}	PA_{3}	PA_{2}	PA_{1}	PA_{0}	18
Source Id	SI_{7}	SI_{6}	SI_{5}	SI_{4}	SI_{3}	SI_{2}	SI_{1}	SI_{0}	19
Source Set No.	SS_{7}	SS_{6}	SS_{5}	SS_{4}	SS_{3}	SS_{2}	SS_{1}	SS_{0}	20
Re-shoot Index	RI_{7}	RI_{6}	RI_{5}	RI_{4}	RI_{3}	RI_{2}	RI_{1}	RI_{0}	21
Group Index	GI_{7}	GI_{6}	GI_{5}	GI_{4}	GI_{3}	GI_{2}	GI_{1}	GI_{0}	22
Depth Index	DI_{7}	DI_{6}	DI_{5}	DI_{4}	DI_{3}	DI_{2}	DI_{1}	DI_{0}	23
Offset Cross-line	OX_{15}	OX_{14}	OX_{13}	OX_{12}	OX_{11}	OX_{10}	OX_{9}	OX_{8}	24
	OX_{7}	OX_{6}	OX_{5}	OX_{4}	OX_{3}	OX_{2}	OX_{1}	OX_{0}	25
Offset In-line	OI_{15}	OI_{14}	OI_{13}	OI_{12}	OI_{11}	OI_{10}	OI_{9}	OI_{8}	26
	OI_{7}	OI_{6}	OI_{5}	OI_{4}	OI_{3}	OI_{2}	OI_{1}	OI_{0}	27
Size	SZ_{15}	SZ_{14}	SZ_{13}	SZ_{12}	SZ_{11}	SZ_{10}	SZ_{9}	SZ_{8}	28
	SZ_{7}	SZ_{6}	SZ_{5}	SZ_{4}	SZ_{3}	SZ_{2}	SZ_{1}	SZ_{0}	29
Offset Depth	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_{9}	OD_{8}	30
	OD_{7}	OD_{6}	OD_{5}	OD_{4}	OD_{3}	OD_{2}	OD_{1}	OD_{0}	31
Header Block Type	0	0	0	1	0	1	0	1	32

7.9.2 EXPLOSIVE

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Expanded File No.	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}	1
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_{9}	EF_{8}	2
	EF_{7}	EF_{6}	EF_{5}	EF_{4}	EF_{3}	EF_{2}	EF_{1}	EF_{0}	3
Source Line No.	SLN_{23}	SLN_{22}	SLN_{21}	SLN_{20}	SLN_{19}	SLN_{18}	SLN_{17}	SLN_{16}	4

(INTEGER)	SLN_{15}	SLN_{14}	SLN_{13}	SLN_{12}	SLN_{11}	SLN_{10}	SLN_{9}	SLN_{8}	5
	SLN_{7}	SLN_{6}	SLN_{5}	SLN_{4}	SLN_{3}	SLN_{2}	SLN_{1}	SLN_{0}	6
Source Line No.	SLN_{-1}	SLN_{-2}	SLN_{-3}	SLN_{-4}	SLN_{-5}	SLN_{-6}	SLN_{-7}	SLN_{-8}	7
(FRACTION)	SLN_{-9}	SLN_{-10}	SLN_{-11}	SLN_{-12}	SLN_{-13}	SLN_{-14}	SLN_{-15}	SLN_{-16}	8
Source Point No.	SPN_{23}	SPN_{22}	SPN_{21}	SPN_{20}	SPN_{19}	SPN_{18}	SPN_{17}	SPN_{16}	9
(INTEGER)	SPN_{15}	SPN_{14}	SPN_{13}	SPN_{12}	SPN_{11}	SPN_{10}	SPN_{9}	SPN_{8}	10
	SPN_{7}	SPN_{6}	SPN_{5}	SPN_{4}	SPN_{3}	SPN_{2}	SPN_{1}	SPN_{0}	11
Source Point No.	SPN_{-1}	SPN_{-2}	SPN_{-3}	SPN_{-4}	SPN_{-5}	SPN_{-6}	SPN_{-7}	SPN_{-8}	12
(FRACTION)	SPN_{-9}	SPN_{-10}	SPN_{-11}	SPN_{-12}	SPN_{-13}	SPN_{-14}	SPN_{-15}	SPN_{-16}	13
Source Point Index	SPI_{7}	SPI_{6}	SPI_{5}	SPI_{4}	SPI_{3}	SPI_{2}	SPI_{1}	SPI_{0}	14
Depth	DE_{15}	DE_{14}	DE_{13}	DE_{12}	DE_{11}	DE_{10}	DE_{9}	DE_{8}	15
	DE_{7}	DE_{6}	DE_{5}	DE_{4}	DE_{3}	DE_{2}	DE_{1}	DE_{0}	16
Charge Length	CL_{7}	CL_{6}	CL_{5}	CL_{4}	CL_{3}	CL_{2}	CL_{1}	CL_{0}	17
Soil Type	ST_{7}	ST_{6}	ST_{5}	ST_{4}	ST_{3}	ST_{2}	ST_{1}	ST_{0}	18
Source Id	SI_{7}	SI_{6}	SI_{5}	SI_{4}	SI_{3}	SI_{2}	SI_{1}	SI_{0}	19
Source Set No.	SS_{7}	SS_{6}	SS_{5}	SS_{4}	SS_{3}	SS_{2}	SS_{1}	SS_{0}	20
Re-shoot Index	RI_{7}	RI_{6}	RI_{5}	RI_{4}	RI_{3}	RI_{2}	RI_{1}	RI_{0}	21
Group Index	GI_{7}	GI_{6}	GI_{5}	GI_{4}	GI_{3}	GI_{2}	GI_{1}	GI_{0}	22
Depth Index	DI_{7}	DI_{6}	DI_{5}	DI_{4}	DI_{3}	DI_{2}	DI_{1}	DI_{0}	23
Offset Cross-line	OX_{15}	OX_{14}	OX_{13}	OX_{12}	OX_{11}	OX_{10}	OX_{9}	OX_{8}	24
	OX_{7}	OX_{6}	OX_{5}	OX_{4}	OX_{3}	OX_{2}	OX_{1}	OX_{0}	25
Offset In-line	OI_{15}	OI_{14}	OI_{13}	OI_{12}	OI_{11}	OI_{10}	OI_{9}	OI_{8}	26
	OI_{7}	OI_{6}	OI_{5}	OI_{4}	OI_{3}	OI_{2}	OI_{1}	OI_{0}	27
Size	SZ_{15}	SZ_{14}	SZ_{13}	SZ_{12}	SZ_{11}	SZ_{10}	SZ_{9}	SZ_{8}	28
	SZ_{7}	SZ_{6}	SZ_{5}	SZ_{4}	SZ_{3}	SZ_{2}	SZ_{1}	SZ_{0}	29
Offset Depth	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_{9}	OD_{8}	30
	OD_{7}	OD_{6}	OD_{5}	OD_{4}	OD_{3}	OD_{2}	OD_{1}	OD_{0}	31
Header Block Type	0	0	0	1	0	1	1	0	32

7.9.3 AIRGUN

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Expanded File No.	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}	1
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_{9}	EF_{8}	2
	EF_{7}	EF_{6}	EF_{5}	EF_{4}	EF_{3}	EF_{2}	EF_{1}	EF_{0}	3
Source Line No.	SLN_{23}	SLN_{22}	SLN_{21}	SLN_{20}	SLN_{19}	SLN_{18}	SLN_{17}	SLN_{16}	4
(INTEGER)	SLN_{15}	SLN_{14}	SLN_{13}	SLN_{12}	SLN_{11}	SLN_{10}	SLN_{9}	SLN_{8}	5
	SLN_{7}	SLN_{6}	SLN_{5}	SLN_{4}	SLN_{3}	SLN_{2}	SLN_{1}	SLN_{0}	6
Source Line No.	SLN_{-1}	SLN_{-2}	SLN_{-3}	SLN_{-4}	SLN_{-5}	SLN_{-6}	SLN_{-7}	SLN_{-8}	7
(FRACTION)	SLN_{-9}	SLN_{-10}	SLN_{-11}	SLN_{-12}	SLN_{-13}	SLN_{-14}	SLN_{-15}	SLN_{-16}	8
Source Point No.	SPN_{23}	SPN_{22}	SPN_{21}	SPN_{20}	SPN_{19}	SPN_{18}	SPN_{17}	SPN_{16}	9
(INTEGER)	SPN_{15}	SPN_{14}	SPN_{13}	SPN_{12}	SPN_{11}	SPN_{10}	SPN_{9}	SPN_{8}	10
	SPN_{7}	SPN_{6}	SPN_{5}	SPN_{4}	SPN_{3}	SPN_{2}	SPN_{1}	SPN_{0}	11
Source Point No.	SPN_{-1}	SPN_{-2}	SPN_{-3}	SPN_{-4}	SPN_{-5}	SPN_{-6}	SPN_{-7}	SPN_{-8}	12
(FRACTION)	SPN_{-9}	SPN_{-10}	SPN_{-11}	SPN_{-12}	SPN_{-13}	SPN_{-14}	SPN_{-15}	SPN_{-16}	13
Source Point Index	SPI_{7}	SPI_{6}	SPI_{5}	SPI_{4}	SPI_{3}	SPI_{2}	SPI_{1}	SPI_{0}	14

Depth	DE_{15}	DE_{14}	DE_{13}	DE_{12}	DE_{11}	DE_{10}	DE_9	DE_8	15
	DE_7	DE_6	DE_5	DE_4	DE_3	DE_2	DE_1	DE_0	16
Air Pressure	AP_{15}	AP_{14}	AP_{13}	AP_{12}	AP_{11}	AP_{10}	AP_9	AP_8	17
	AP_7	AP_6	AP_5	AP_4	AP_3	AP_2	AP_1	AP_0	18
Source Id	SI_7	SI_6	SI_5	SI_4	SI_3	SI_2	SI_1	SI_0	19
Source Set No.	SS_7	SS_6	SS_5	SS_4	SS_3	SS_2	SS_1	SS_0	20
Re-shoot Index	RI_7	RI_6	RI_5	RI_4	RI_3	RI_2	RI_1	RI_0	21
Group Index	GI_7	GI_6	GI_5	GI_4	GI_3	GI_2	GI_1	GI_0	22
Depth Index	DI_7	DI_6	DI_5	DI_4	DI_3	DI_2	DI_1	DI_0	23
Offset Cross-line	OX_{15}	OX_{14}	OX_{13}	OX_{12}	OX_{11}	OX_{10}	OX_9	OX_8	24
	OX_7	OX_6	OX_5	OX_4	OX_3	OX_2	OX_1	OX_0	25
Offset In-line	OI_{15}	OI_{14}	OI_{13}	OI_{12}	OI_{11}	OI_{10}	OI_9	OI_8	26
	OI_7	OI_6	OI_5	OI_4	OI_3	OI_2	OI_1	OI_0	27
Size	SZ_{15}	SZ_{14}	SZ_{13}	SZ_{12}	SZ_{11}	SZ_{10}	SZ_9	SZ_8	28
	SZ_7	SZ_6	SZ_5	SZ_4	SZ_3	SZ_2	SZ_1	SZ_0	29
Offset Depth	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_9	OD_8	30
	OD_7	OD_6	OD_5	OD_4	OD_3	OD_2	OD_1	OD_0	31
Header Block Type	0	0	0	1	0	1	1	1	32

7.9.4 WATERGUN

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Expanded File No.	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}	1
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_9	EF_8	2
	EF_7	EF_6	EF_5	EF_4	EF_3	EF_2	EF_1	EF_0	3
Source Line No. (INTEGER)	SLN_{23}	SLN_{22}	SLN_{21}	SLN_{20}	SLN_{19}	SLN_{18}	SLN_{17}	SLN_{16}	4
	SLN_{15}	SLN_{14}	SLN_{13}	SLN_{12}	SLN_{11}	SLN_{10}	SLN_9	SLN_8	5
	SLN_7	SLN_6	SLN_5	SLN_4	SLN_3	SLN_2	SLN_1	SLN_0	6
Source Line No. (FRACTION)	SLN_{-1}	SLN_{-2}	SLN_{-3}	SLN_{-4}	SLN_{-5}	SLN_{-6}	SLN_{-7}	SLN_{-8}	7
	SLN_{-9}	SLN_{-10}	SLN_{-11}	SLN_{-12}	SLN_{-13}	SLN_{-14}	SLN_{-15}	SLN_{-16}	8
Source Point No. (INTEGER)	SPN_{23}	SPN_{22}	SPN_{21}	SPN_{20}	SPN_{19}	SPN_{18}	SPN_{17}	SPN_{16}	9
	SPN_{15}	SPN_{14}	SPN_{13}	SPN_{12}	SPN_{11}	SPN_{10}	SPN_9	SPN_8	10
	SPN_7	SPN_6	SPN_5	SPN_4	SPN_3	SPN_2	SPN_1	SPN_0	11
Source Point No. (FRACTION)	SPN_{-1}	SPN_{-2}	SPN_{-3}	SPN_{-4}	SPN_{-5}	SPN_{-6}	SPN_{-7}	SPN_{-8}	12
	SPN_{-9}	SPN_{-10}	SPN_{-11}	SPN_{-12}	SPN_{-13}	SPN_{-14}	SPN_{-15}	SPN_{-16}	13
Source Point Index	SPI_7	SPI_6	SPI_5	SPI_4	SPI_3	SPI_2	SPI_1	SPI_0	14
Depth	DE_{15}	DE_{14}	DE_{13}	DE_{12}	DE_{11}	DE_{10}	DE_9	DE_8	15
	DE_7	DE_6	DE_5	DE_4	DE_3	DE_2	DE_1	DE_0	16
Air Pressure	AP_{15}	AP_{14}	AP_{13}	AP_{12}	AP_{11}	AP_{10}	AP_9	AP_8	17
	AP_7	AP_6	AP_5	AP_4	AP_3	AP_2	AP_1	AP_0	18
Source Id	SI_7	SI_6	SI_5	SI_4	SI_3	SI_2	SI_1	SI_0	19
Source Set No.	SS_7	SS_6	SS_5	SS_4	SS_3	SS_2	SS_1	SS_0	20
Re-shoot Index	RI_7	RI_6	RI_5	RI_4	RI_3	RI_2	RI_1	RI_0	21
Group Index	GI_7	GI_6	GI_5	GI_4	GI_3	GI_2	GI_1	GI_0	22
Depth Index	DI_7	DI_6	DI_5	DI_4	DI_3	DI_2	DI_1	DI_0	23
Offset Cross-line	OX_{15}	OX_{14}	OX_{13}	OX_{12}	OX_{11}	OX_{10}	OX_9	OX_8	24

	OX_7	OX_6	OX_5	OX_4	OX_3	OX_2	OX_1	OX_0	25
Offset In-line	OI_{15}	OI_{14}	OI_{13}	OI_{12}	OI_{11}	OI_{10}	OI_9	OI_8	26
	OI_7	OI_6	OI_5	OI_4	OI_3	OI_2	OI_1	OI_0	27
Size	SZ_{15}	SZ_{14}	SZ_{13}	SZ_{12}	SZ_{11}	SZ_{10}	SZ_9	SZ_8	28
	SZ_7	SZ_6	SZ_5	SZ_4	SZ_3	SZ_2	SZ_1	SZ_0	29
Offset Depth	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_9	OD_8	30
	OD_7	OD_6	OD_5	OD_4	OD_3	OD_2	OD_1	OD_0	31
Header Block Type	0	0	0	1	1	0	0	0	32

7.9.5 ELECTROMAGNETIC SOURCE

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Expanded File No.	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}	1
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_9	EF_8	2
	EF_7	EF_6	EF_5	EF_4	EF_3	EF_2	EF_1	EF_0	3
Source Line No. (INTEGER)	SLN_{23}	SLN_{22}	SLN_{21}	SLN_{20}	SLN_{19}	SLN_{18}	SLN_{17}	SLN_{16}	4
	SLN_{15}	SLN_{14}	SLN_{13}	SLN_{12}	SLN_{11}	SLN_{10}	SLN_9	SLN_8	5
	SLN_7	SLN_6	SLN_5	SLN_4	SLN_3	SLN_2	SLN_1	SLN_0	6
Source Line No. (FRACTION)	SLN_{-1}	SLN_{-2}	SLN_{-3}	SLN_{-4}	SLN_{-5}	SLN_{-6}	SLN_{-7}	SLN_{-8}	7
	SLN_{-9}	SLN_{-10}	SLN_{-11}	SLN_{-12}	SLN_{-13}	SLN_{-14}	SLN_{-15}	SLN_{-16}	8
Source Point No. (INTEGER)	SPN_{23}	SPN_{22}	SPN_{21}	SPN_{20}	SPN_{19}	SPN_{18}	SPN_{17}	SPN_{16}	9
	SPN_{15}	SPN_{14}	SPN_{13}	SPN_{12}	SPN_{11}	SPN_{10}	SPN_9	SPN_8	10
	SPN_7	SPN_6	SPN_5	SPN_4	SPN_3	SPN_2	SPN_1	SPN_0	11
Source Point No. (FRACTION)	SPN_{-1}	SPN_{-2}	SPN_{-3}	SPN_{-4}	SPN_{-5}	SPN_{-6}	SPN_{-7}	SPN_{-8}	12
	SPN_{-9}	SPN_{-10}	SPN_{-11}	SPN_{-12}	SPN_{-13}	SPN_{-14}	SPN_{-15}	SPN_{-16}	13
Source Point Index	SPI_7	SPI_6	SPI_5	SPI_4	SPI_3	SPI_2	SPI_1	SPI_0	14
Source Type	ST_7	ST_6	ST_5	ST_4	ST_3	ST_2	ST_1	ST_0	15
Moment	MO_{23}	MO_{22}	MO_{21}	MO_{20}	MO_{19}	MO_{18}	MO_{17}	MO_{16}	16
	MO_{15}	MO_{14}	MO_{13}	MO_{12}	MO_{11}	MO_{10}	MO_9	MO_8	17
	MO_7	MO_6	MO_5	MO_4	MO_3	MO_2	MO_1	MO_0	18
Source Id	SI_7	SI_6	SI_5	SI_4	SI_3	SI_2	SI_1	SI_0	19
Source Set No.	SS_7	SS_6	SS_5	SS_4	SS_3	SS_2	SS_1	SS_0	20
Re-shoot Index	RI_7	RI_6	RI_5	RI_4	RI_3	RI_2	RI_1	RI_0	21
Group Index	GI_7	GI_6	GI_5	GI_4	GI_3	GI_2	GI_1	GI_0	22
Depth Index	DI_7	DI_6	DI_5	DI_4	DI_3	DI_2	DI_1	DI_0	23
Offset Cross-line	OX_{15}	OX_{14}	OX_{13}	OX_{12}	OX_{11}	OX_{10}	OX_9	OX_8	24
	OX_7	OX_6	OX_5	OX_4	OX_3	OX_2	OX_1	OX_0	25
Offset In-line	OI_{15}	OI_{14}	OI_{13}	OI_{12}	OI_{11}	OI_{10}	OI_9	OI_8	26
	OI_7	OI_6	OI_5	OI_4	OI_3	OI_2	OI_1	OI_0	27
Current/Voltage	CV_{15}	CV_{14}	CV_{13}	CV_{12}	CV_{11}	CV_{10}	CV_9	CV_8	28
	CV_7	CV_6	CV_5	CV_4	CV_3	CV_2	CV_1	CV_0	29
Offset Depth	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_9	OD_8	30
	OD_7	OD_6	OD_5	OD_4	OD_3	OD_2	OD_1	OD_0	31
Header Block Type	0	0	0	1	1	0	0	1	32

7.9.6 OTHER SOURCE

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Expanded File No.	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}	1
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_{9}	EF_{8}	2
	EF_{7}	EF_{6}	EF_{5}	EF_{4}	EF_{3}	EF_{2}	EF_{1}	EF_{0}	3
Source Line No. (INTEGER)	SLN_{23}	SLN_{22}	SLN_{21}	SLN_{20}	SLN_{19}	SLN_{18}	SLN_{17}	SLN_{16}	4
	SLN_{15}	SLN_{14}	SLN_{13}	SLN_{12}	SLN_{11}	SLN_{10}	SLN_{9}	SLN_{8}	5
	SLN_{7}	SLN_{6}	SLN_{5}	SLN_{4}	SLN_{3}	SLN_{2}	SLN_{1}	SLN_{0}	6
Source Line No. (FRACTION)	SLN_{-1}	SLN_{-2}	SLN_{-3}	SLN_{-4}	SLN_{-5}	SLN_{-6}	SLN_{-7}	SLN_{-8}	7
	SLN_{-9}	SLN_{-10}	SLN_{-11}	SLN_{-12}	SLN_{-13}	SLN_{-14}	SLN_{-15}	SLN_{-16}	8
Source Point No. (INTEGER)	SPN_{23}	SPN_{22}	SPN_{21}	SPN_{20}	SPN_{19}	SPN_{18}	SPN_{17}	SPN_{16}	9
	SPN_{15}	SPN_{14}	SPN_{13}	SPN_{12}	SPN_{11}	SPN_{10}	SPN_{9}	SPN_{8}	10
	SPN_{7}	SPN_{6}	SPN_{5}	SPN_{4}	SPN_{3}	SPN_{2}	SPN_{1}	SPN_{0}	11
Source Point No. (FRACTION)	SPN_{-1}	SPN_{-2}	SPN_{-3}	SPN_{-4}	SPN_{-5}	SPN_{-6}	SPN_{-7}	SPN_{-8}	12
	SPN_{-9}	SPN_{-10}	SPN_{-11}	SPN_{-12}	SPN_{-13}	SPN_{-14}	SPN_{-15}	SPN_{-16}	13
Source Point Index	SPI_{7}	SPI_{6}	SPI_{5}	SPI_{4}	SPI_{3}	SPI_{2}	SPI_{1}	SPI_{0}	14
	X	X	X	X	X	X	X	X	15
	X	X	X	X	X	X	X	X	16
	X	X	X	X	X	X	X	X	17
	X	X	X	X	X	X	X	X	18
Source Id	SI_{7}	SI_{6}	SI_{5}	SI_{4}	SI_{3}	SI_{2}	SI_{1}	SI_{0}	19
Source Set No.	SS_{7}	SS_{6}	SS_{5}	SS_{4}	SS_{3}	SS_{2}	SS_{1}	SS_{0}	20
Re-shoot Index	RI_{7}	RI_{6}	RI_{5}	RI_{4}	RI_{3}	RI_{2}	RI_{1}	RI_{0}	21
Group Index	GI_{7}	GI_{6}	GI_{5}	GI_{4}	GI_{3}	GI_{2}	GI_{1}	GI_{0}	22
Depth Index	DI_{7}	DI_{6}	DI_{5}	DI_{4}	DI_{3}	DI_{2}	DI_{1}	DI_{0}	23
Offset Cross-line	OX_{15}	OX_{14}	OX_{13}	OX_{12}	OX_{11}	OX_{10}	OX_{9}	OX_{8}	24
	OX_{7}	OX_{6}	OX_{5}	OX_{4}	OX_{3}	OX_{2}	OX_{1}	OX_{0}	25
Offset In-line	OI_{15}	OI_{14}	OI_{13}	OI_{12}	OI_{11}	OI_{10}	OI_{9}	OI_{8}	26
	OI_{7}	OI_{6}	OI_{5}	OI_{4}	OI_{3}	OI_{2}	OI_{1}	OI_{0}	27
	X	X	X	X	X	X	X	X	28
	X	X	X	X	X	X	X	X	29
Offset Depth	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_{9}	OD_{8}	30
	OD_{7}	OD_{6}	OD_{5}	OD_{4}	OD_{3}	OD_{2}	OD_{1}	OD_{0}	31
Header Block Type	0	0	0	1	1	1	1	1	32

7.10 ADDITIONAL SOURCE INFO (optional)

Follows the Source Description Block

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD

Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Time	T_{63}	T_{62}	T_{61}	T_{60}	T_{59}	T_{58}	T_{57}	T_{56}	1
	T_{55}	T_{54}	T_{53}	T_{52}	T_{51}	T_{50}	T_{49}	T_{48}	2
	T_{47}	T_{46}	T_{45}	T_{44}	T_{43}	T_{42}	T_{41}	T_{40}	3
	T_{39}	T_{38}	T_{37}	T_{36}	T_{35}	T_{34}	T_{33}	T_{32}	4
	T_{31}	T_{30}	T_{29}	T_{28}	T_{27}	T_{26}	T_{25}	T_{24}	5
	T_{23}	T_{22}	T_{21}	T_{20}	T_{19}	T_{18}	T_{17}	T_{16}	6
	T_{15}	T_{14}	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8	7
	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0	8
Source Status	SST_7	SST_6	SST_5	SST_4	SST_3	SST_2	SST_1	SST_0	9
Source Id	SI_7	SI_6	SI_5	SI_4	SI_3	SI_2	SI_1	SI_0	10
Source Moving	MV_7	MV_6	MV_5	MV_4	MV_3	MV_2	MV_1	MV_0	11
Error description	ED	ED	ED	ED	ED	ED	ED	ED	12
	ED	ED	ED	ED	ED	ED	ED	ED	13
	ED	ED	ED	ED	ED	ED	ED	ED	14
	ED	ED	ED	ED	ED	ED	ED	ED	15
	ED	ED	ED	ED	ED	ED	ED	ED	16
	ED	ED	ED	ED	ED	ED	ED	ED	17
	ED	ED	ED	ED	ED	ED	ED	ED	18
	ED	ED	ED	ED	ED	ED	ED	ED	19
	ED	ED	ED	ED	ED	ED	ED	ED	20
	ED	ED	ED	ED	ED	ED	ED	ED	21
	ED	ED	ED	ED	ED	ED	ED	ED	22
	ED	ED	ED	ED	ED	ED	ED	ED	23
	ED	ED	ED	ED	ED	ED	ED	ED	24
	ED	ED	ED	ED	ED	ED	ED	ED	25
	ED	ED	ED	ED	ED	ED	ED	ED	26
	ED	ED	ED	ED	ED	ED	ED	ED	27
	ED	ED	ED	ED	ED	ED	ED	ED	28
	ED	ED	ED	ED	ED	ED	ED	ED	29
	ED	ED	ED	ED	ED	ED	ED	ED	30
	ED	ED	ED	ED	ED	ED	ED	ED	31
Header Block Type	0	0	0	1	1	0	1	0	32

7.11 SOURCE AUXILIARY CHANNEL REFERENCE (optional)

Follows the Source Description Block

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Source Id	SI_7	SI_6	SI_5	SI_4	SI_3	SI_2	SI_1	SI_0	1
Scan Type Number 1	$ST1_1$	$ST1_1$	$ST1_1$	$ST1_1$	$ST1_2$	$ST1_2$	$ST1_2$	$ST1_2$	2
Channel Set Number 1	$CS1_{15}$	$CS1_{14}$	$CS1_{13}$	$CS1_{12}$	$CS1_{11}$	$CS1_{10}$	$CS1_9$	$CS1_8$	3
	$CS1_7$	$CS1_6$	$CS1_5$	$CS1_4$	$CS1_3$	$CS1_2$	$CS1_1$	$CS1_0$	4
Trace Number 1	$TN1_{23}$	$TN1_{22}$	$TN1_{21}$	$TN1_{20}$	$TN1_{19}$	$TN1_{18}$	$TN1_{17}$	$TN1_{16}$	5
	$TN1_{15}$	$TN1_{14}$	$TN1_{13}$	$TN1_{12}$	$TN1_{11}$	$TN1_{10}$	$TN1_9$	$TN1_8$	6
	$TN1_7$	$TN1_6$	$TN1_5$	$TN1_4$	$TN1_3$	$TN1_2$	$TN1_1$	$TN1_0$	7

Scan Type Number 2	$ST2_1$	$ST2_1$	$ST2_1$	$ST2_1$	$ST2_2$	$ST2_2$	$ST2_2$	$ST2_2$	8
Channel Set Number 2	$CS2_{15}$	$CS2_{14}$	$CS2_{13}$	$CS2_{12}$	$CS2_{11}$	$CS2_{10}$	$CS2_9$	$CS2_8$	9
	$CS2_7$	$CS2_6$	$CS2_5$	$CS2_4$	$CS2_3$	$CS2_2$	$CS2_1$	$CS2_0$	10
Trace Number 2	$TN2_{23}$	$TN2_{22}$	$TN2_{21}$	$TN2_{20}$	$TN2_{19}$	$TN2_{18}$	$TN2_{17}$	$TN2_{16}$	11
	$TN2_{15}$	$TN2_{14}$	$TN2_{13}$	$TN2_{12}$	$TN2_{11}$	$TN2_{10}$	$TN2_9$	$TN1_8$	12
	$TN2_7$	$TN2_6$	$TN2_5$	$TN2_4$	$TN2_3$	$TN2_2$	$TN2_1$	$TN2_0$	13
Scan Type Number 3	$ST3_1$	$ST3_1$	$ST3_1$	$ST3_1$	$ST3_2$	$ST3_2$	$ST3_2$	$ST3_2$	14
Channel Set Number 3	$CS3_{15}$	$CS3_{14}$	$CS3_{13}$	$CS3_{12}$	$CS3_{11}$	$CS3_{10}$	$CS3_9$	$CS3_8$	15
	$CS3_7$	$CS3_6$	$CS3_5$	$CS3_4$	$CS3_3$	$CS3_2$	$CS3_1$	$CS3_0$	16
Trace Number 3	$TN3_{23}$	$TN3_{22}$	$TN3_{21}$	$TN3_{20}$	$TN3_{19}$	$TN3_{18}$	$TN3_{17}$	$TN3_{16}$	17
	$TN3_{15}$	$TN3_{14}$	$TN3_{13}$	$TN3_{12}$	$TN3_{11}$	$TN3_{10}$	$TN3_9$	$TN3_8$	18
	$TN3_7$	$TN3_6$	$TN3_5$	$TN3_4$	$TN3_3$	$TN3_2$	$TN3_1$	$TN3_0$	19
Scan Type Number 4	$ST4_1$	$ST4_1$	$ST4_1$	$ST4_1$	$ST4_2$	$ST4_2$	$ST4_2$	$ST4_2$	20
Channel Set Number 4	$CS4_{15}$	$CS4_{14}$	$CS4_{13}$	$CS4_{12}$	$CS4_{11}$	$CS4_{10}$	$CS4_9$	$CS4_8$	21
	$CS4_7$	$CS4_6$	$CS4_5$	$CS4_4$	$CS4_3$	$CS4_2$	$CS4_1$	$CS4_0$	22
Trace Number 4	$TN4_{23}$	$TN4_{22}$	$TN4_{21}$	$TN4_{20}$	$TN4_{19}$	$TN4_{18}$	$TN4_{17}$	$TN4_{16}$	23
	$TN4_{15}$	$TN4_{14}$	$TN4_{13}$	$TN4_{12}$	$TN4_{11}$	$TN4_{10}$	$TN4_9$	$TN4_8$	24
	$TN4_7$	$TN4_6$	$TN4_5$	$TN4_4$	$TN4_3$	$TN4_2$	$TN4_1$	$TN4_0$	25
Scan Type Number 5	$ST5_1$	$ST5_1$	$ST5_1$	$ST5_1$	$ST5_2$	$ST5_2$	$ST5_2$	$ST5_2$	26
Channel Set Number 5	$CS5_{15}$	$CS5_{14}$	$CS5_{13}$	$CS5_{12}$	$CS5_{11}$	$CS5_{10}$	$CS5_9$	$CS5_8$	27
	$CS5_7$	$CS5_6$	$CS5_5$	$CS5_4$	$CS5_3$	$CS5_2$	$CS5_1$	$CS5_0$	28
Trace Number 5	$TN5_{23}$	$TN5_{22}$	$TN5_{21}$	$TN5_{20}$	$TN5_{19}$	$TN5_{18}$	$TN5_{17}$	$TN5_{16}$	29
	$TN5_{15}$	$TN5_{14}$	$TN5_{13}$	$TN5_{12}$	$TN5_{11}$	$TN5_{10}$	$TN5_9$	$TN5_8$	30
	$TN5_7$	$TN5_6$	$TN5_5$	$TN5_4$	$TN5_3$	$TN5_2$	$TN5_1$	$TN5_0$	31
Header Block Type	0	0	1	0	0	0	0	1	32

7.12 COORDINATE REFERENCE SYSTEM IDENTIFICATION (conditional)

This header block is mandatory if a position block is included in the record.

Bit No.	0	1	2	3	4	5	6	7	
Coordinate Reference System (CRS) identification	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	1
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	2
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	3
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	4
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	5
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	6
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	7
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	8
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	9
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	10
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	11
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	12
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	13
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	14
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	15
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	16
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	17
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	18
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	19
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	20
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	21
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	22
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	23
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	24
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	25
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	26
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	27
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	28
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	29
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	30
	CRID	CRID	CRID	CRID	CRID	CRID	CRID	CRID	31
Header Blk Type	0	1	0	1	0	1	0	1	32

7.13 POSITION BLOCKS (optional)

Bit No.	0	1	2	3	4	5	6	7	
Time of position	TP_{63}	TP_{62}	TP_{61}	TP_{60}	TP_{59}	TP_{58}	TP_{57}	TP_{56}	1
	TP_{55}	TP_{54}	TP_{53}	TP_{52}	TP_{51}	TP_{50}	TP_{49}	TP_{48}	2
	TP_{47}	TP_{46}	TP_{45}	TP_{44}	TP_{43}	TP_{42}	TP_{41}	TP_{40}	3
	TP_{39}	TP_{38}	TP_{37}	TP_{36}	TP_{35}	TP_{34}	TP_{33}	TP_{32}	4
	TP_{31}	TP_{30}	TP_{29}	TP_{28}	TP_{27}	TP_{26}	TP_{25}	TP_{24}	5
	TP_{23}	TP_{22}	TP_{21}	TP_{20}	TP_{19}	TP_{18}	TP_{17}	TP_{16}	6
	TP_{15}	TP_{14}	TP_{13}	TP_{12}	TP_{11}	TP_{10}	TP_{9}	TP_{8}	7

	TP_7	TP_6	TP_5	TP_4	TP_3	TP_2	TP_1	TP_0	8
Time of measurement/ calculation	TC_{63}	TC_{62}	TC_{61}	TC_{60}	TC_{59}	TC_{58}	TC_{57}	TC_{56}	9
	TC_{55}	TC_{54}	TC_{53}	TC_{52}	TC_{51}	TC_{50}	TC_{49}	TC_{48}	10
	TC_{47}	TC_{46}	TC_{45}	TC_{44}	TC_{43}	TC_{42}	TC_{41}	TC_{40}	11
	TC_{39}	TC_{38}	TC_{37}	TC_{36}	TC_{35}	TC_{34}	TC_{33}	TC_{32}	12
	TC_{31}	TC_{30}	TC_{29}	TC_{28}	TC_{27}	TC_{26}	TC_{25}	TC_{24}	13
	TC_{23}	TC_{22}	TC_{21}	TC_{20}	TC_{19}	TC_{18}	TC_{17}	TC_{16}	14
	TC_{15}	TC_{14}	TC_{13}	TC_{12}	TC_{11}	TC_{10}	TC_9	TC_8	15
	TC_7	TC_6	TC_5	TC_4	TC_3	TC_2	TC_1	TC_0	16
Vertical error	VE_{31}	VE_{30}	VE_{29}	VE_{28}	VE_{27}	VE_{26}	VE_{25}	VE_{24}	17
	VE_{23}	VE_{22}	VE_{21}	VE_{20}	VE_{19}	VE_{18}	VE_{17}	VE_{16}	18
	VE_{15}	VE_{14}	VE_{13}	VE_{12}	VE_{11}	VE_{10}	VE_9	VE_8	19
	VE_7	VE_6	VE_5	VE_4	VE_3	VE_2	VE_1	VE_0	20
Horizontal error ellipse semi-major dimension	HEA_{31}	HEA_{30}	HEA_{29}	HEA_{28}	HEA_{27}	HEA_{26}	HEA_{25}	HEA_{24}	21
	HEA_{23}	HEA_{22}	HEA_{21}	HEA_{20}	HEA_{19}	HEA_{18}	HEA_{17}	HEA_{16}	22
	HEA_{15}	HEA_{14}	HEA_{13}	HEA_{12}	HEA_{11}	HEA_{10}	HEA_9	HEA_8	23
	HEA_7	HEA_6	HEA_5	HEA_4	HEA_3	HEA_2	HEA_1	HEA_0	24
Horizontal error ellipse semi-minor dimension	HEB_{31}	HEB_{30}	HEB_{29}	HEB_{28}	HEB_{27}	HEB_{26}	HEB_{25}	HEB_{24}	25
	HEB_{23}	HEB_{22}	HEB_{21}	HEB_{20}	HEB_{19}	HEB_{18}	HEB_{17}	HEB_{16}	26
	HEB_{15}	HEB_{14}	HEB_{13}	HEB_{12}	HEB_{11}	HEB_{10}	HEB_9	HEB_8	27
	HEB_7	HEB_6	HEB_5	HEB_4	HEB_3	HEB_2	HEB_1	HEB_0	28
Horizontal error ellipse orientation	HEO_{15}	HEO_{14}	HEO_{13}	HEO_{12}	HEO_{11}	HEO_{10}	HEO_9	HEO_8	29
	HEO_7	HEO_6	HEO_5	HEO_4	HEO_3	HEO_2	HEO_1	HEO_0	30
Position Type	PT_7	PT_6	PT_5	PT_4	PT_3	PT_2	PT_1	PT_0	31
Header Blk Type	0	1	0	1	0	0	0	0	32

Bit No.	0	1	2	3	4	5	6	7	
Coord tuple 1 Coord 1	$T1C1_{63}$	$T1C1_{62}$	$T1C1_{61}$	$T1C1_{60}$	$T1C1_{59}$	$T1C1_{58}$	$T1C1_{57}$	$T1C1_{56}$	33
	$T1C1_{55}$	$T1C1_{54}$	$T1C1_{53}$	$T1C1_{52}$	$T1C1_{51}$	$T1C1_{50}$	$T1C1_{49}$	$T1C1_{48}$	34
	$T1C1_{47}$	$T1C1_{46}$	$T1C1_{45}$	$T1C1_{44}$	$T1C1_{43}$	$T1C1_{42}$	$T1C1_{41}$	$T1C1_{40}$	35
	$T1C1_{39}$	$T1C1_{38}$	$T1C1_{37}$	$T1C1_{36}$	$T1C1_{35}$	$T1C1_{34}$	$T1C1_{33}$	$T1C1_{32}$	36
	$T1C1_{31}$	$T1C1_{30}$	$T1C1_{29}$	$T1C1_{28}$	$T1C1_{27}$	$T1C1_{26}$	$T1C1_{25}$	$T1C1_{24}$	37
	$T1C1_{23}$	$T1C1_{22}$	$T1C1_{21}$	$T1C1_{20}$	$T1C1_{19}$	$T1C1_{18}$	$T1C1_{17}$	$T1C1_{16}$	38
	$T1C1_{15}$	$T1C1_{14}$	$T1C1_{13}$	$T1C1_{12}$	$T1C1_{11}$	$T1C1_{10}$	$T1C1_9$	$T1C1_8$	39
	$T1C1_7$	$T1C1_6$	$T1C1_5$	$T1C1_4$	$T1C1_3$	$T1C1_2$	$T1C1_1$	$T1C1_0$	40
Coord tuple 1 Coord 2	$T1C2_{63}$	$T1C2_{62}$	$T1C2_{61}$	$T1C2_{60}$	$T1C2_{59}$	$T1C2_{58}$	$T1C2_{57}$	$T1C2_{56}$	41
	$T1C2_{55}$	$T1C2_{54}$	$T1C2_{53}$	$T1C2_{52}$	$T1C2_{51}$	$T1C2_{50}$	$T1C2_{49}$	$T1C2_{48}$	42
	$T1C2_{47}$	$T1C2_{46}$	$T1C2_{45}$	$T1C2_{44}$	$T1C2_{43}$	$T1C2_{42}$	$T1C2_{41}$	$T1C2_{40}$	43
	$T1C2_{39}$	$T1C2_{38}$	$T1C2_{37}$	$T1C2_{36}$	$T1C2_{35}$	$T1C2_{34}$	$T1C2_{33}$	$T1C2_{32}$	44
	$T1C2_{31}$	$T1C2_{30}$	$T1C2_{29}$	$T1C2_{28}$	$T1C2_{27}$	$T1C2_{26}$	$T1C2_{25}$	$T1C2_{24}$	45
	$T1C2_{23}$	$T1C2_{22}$	$T1C2_{21}$	$T1C2_{20}$	$T1C2_{19}$	$T1C2_{18}$	$T1C2_{17}$	$T1C2_{16}$	46
	$T1C2_{15}$	$T1C2_{14}$	$T1C2_{13}$	$T1C2_{12}$	$T1C2_{11}$	$T1C2_{10}$	$T1C2_9$	$T1C2_8$	47
	$T1C2_7$	$T1C2_6$	$T1C2_5$	$T1C2_4$	$T1C2_3$	$T1C2_2$	$T1C2_1$	$T1C2_0$	48
Coord tuple 1 Coord 3	$T1C3_{63}$	$T1C3_{62}$	$T1C3_{61}$	$T1C3_{60}$	$T1C3_{59}$	$T1C3_{58}$	$T1C3_{57}$	$T1C3_{56}$	49
	$T1C3_{55}$	$T1C3_{54}$	$T1C3_{53}$	$T1C3_{52}$	$T1C3_{51}$	$T1C3_{50}$	$T1C3_{49}$	$T1C3_{48}$	50
	$T1C3_{47}$	$T1C3_{46}$	$T1C3_{45}$	$T1C3_{44}$	$T1C3_{43}$	$T1C3_{42}$	$T1C3_{41}$	$T1C3_{40}$	51
	$T1C3_{39}$	$T1C3_{38}$	$T1C3_{37}$	$T1C3_{36}$	$T1C3_{35}$	$T1C3_{34}$	$T1C3_{33}$	$T1C3_{32}$	52
	$T1C3_{31}$	$T1C3_{30}$	$T1C3_{29}$	$T1C3_{28}$	$T1C3_{27}$	$T1C3_{26}$	$T1C3_{25}$	$T1C3_{24}$	53
	$T1C3_{23}$	$T1C3_{22}$	$T1C3_{21}$	$T1C3_{20}$	$T1C3_{19}$	$T1C3_{18}$	$T1C3_{17}$	$T1C3_{16}$	54

	$T1C3_{15}$	$T1C3_{14}$	$T1C3_{13}$	$T1C3_{12}$	$T1C3_{11}$	$T1C3_{10}$	$T1C3_{9}$	$T1C3_{8}$	55
	$T1C3_{7}$	$T1C3_{6}$	$T1C3_{5}$	$T1C3_{4}$	$T1C3_{3}$	$T1C3_{2}$	$T1C3_{1}$	$T1C3_{0}$	56
Stanza ID 1	$ID1_{15}$	$ID1_{14}$	$ID1_{13}$	$ID1_{12}$	$ID1_{11}$	$ID1_{10}$	$ID1_{9}$	$ID1_{8}$	57
	$ID1_{7}$	$ID1_{6}$	$ID1_{5}$	$ID1_{4}$	$ID1_{3}$	$ID1_{2}$	$ID1_{1}$	$ID1_{0}$	58
Position 1 valid	$PV1_{7}$	$PV1_{6}$	$PV1_{5}$	$PV1_{4}$	$PV1_{3}$	$PV1_{2}$	$PV1_{1}$	$PV1_{0}$	59
Position 1 quality	$PQ1_{7}$	$PQ1_{6}$	$PQ1_{5}$	$PQ1_{4}$	$PQ1_{3}$	$PQ1_{2}$	$PQ1_{1}$	$PQ1_{0}$	60
Undefined	X	X	X	X	X	X	X	X	61
	X	X	X	X	X	X	X	X	62
	X	X	X	X	X	X	X	X	63
Header Blk Type	0	1	0	1	0	0	0	1	64

Bit No.	0	1	2	3	4	5	6	7	
Coord tuple 2 Coord 1	$T2C1_{63}$	$T2C1_{62}$	$T2C1_{61}$	$T2C1_{60}$	$T2C1_{59}$	$T2C1_{58}$	$T2C1_{57}$	$T2C1_{56}$	65
	$T2C1_{55}$	$T2C1_{54}$	$T2C1_{53}$	$T2C1_{52}$	$T2C1_{51}$	$T2C1_{50}$	$T2C1_{49}$	$T2C1_{48}$	66
	$T2C1_{47}$	$T2C1_{46}$	$T2C1_{45}$	$T2C1_{44}$	$T2C1_{43}$	$T2C1_{42}$	$T2C1_{41}$	$T2C1_{40}$	67
	$T2C1_{39}$	$T2C1_{38}$	$T2C1_{37}$	$T2C1_{36}$	$T2C1_{35}$	$T2C1_{34}$	$T2C1_{33}$	$T2C1_{32}$	68
	$T2C1_{31}$	$T2C1_{30}$	$T2C1_{29}$	$T2C1_{28}$	$T2C1_{27}$	$T2C1_{26}$	$T2C1_{25}$	$T2C1_{24}$	69
	$T2C1_{23}$	$T2C1_{22}$	$T2C1_{21}$	$T2C1_{20}$	$T2C1_{19}$	$T2C1_{18}$	$T2C1_{17}$	$T2C1_{16}$	70
	$T2C1_{15}$	$T2C1_{14}$	$T2C1_{13}$	$T2C1_{12}$	$T2C1_{11}$	$T2C1_{10}$	$T2C1_{9}$	$T2C1_{8}$	71
	$T2C1_{7}$	$T2C1_{6}$	$T2C1_{5}$	$T2C1_{4}$	$T2C1_{3}$	$T2C1_{2}$	$T2C1_{1}$	$T2C1_{0}$	72
Coord tuple 2 Coord 2	$T2C2_{63}$	$T2C2_{62}$	$T2C2_{61}$	$T2C2_{60}$	$T2C2_{59}$	$T2C2_{58}$	$T2C2_{57}$	$T2C2_{56}$	73
	$T2C2_{55}$	$T2C2_{54}$	$T2C2_{53}$	$T2C2_{52}$	$T2C2_{51}$	$T2C2_{50}$	$T2C2_{49}$	$T2C2_{48}$	74
	$T2C2_{47}$	$T2C2_{46}$	$T2C2_{45}$	$T2C2_{44}$	$T2C2_{43}$	$T2C2_{42}$	$T2C2_{41}$	$T2C2_{40}$	75
	$T2C2_{39}$	$T2C2_{38}$	$T2C2_{37}$	$T2C2_{36}$	$T2C2_{35}$	$T2C2_{34}$	$T2C2_{33}$	$T2C2_{32}$	76
	$T2C2_{31}$	$T2C2_{30}$	$T2C2_{29}$	$T2C2_{28}$	$T2C2_{27}$	$T2C2_{26}$	$T2C2_{25}$	$T2C2_{24}$	77
	$T2C2_{23}$	$T2C2_{22}$	$T2C2_{21}$	$T2C2_{20}$	$T2C2_{19}$	$T2C2_{18}$	$T2C2_{17}$	$T2C2_{16}$	78
	$T2C2_{15}$	$T2C2_{14}$	$T2C2_{13}$	$T2C2_{12}$	$T2C2_{11}$	$T2C2_{10}$	$T2C2_{9}$	$T2C2_{8}$	79
	$T2C2_{7}$	$T2C2_{6}$	$T2C2_{5}$	$T2C2_{4}$	$T2C2_{3}$	$T2C2_{2}$	$T2C2_{1}$	$T2C2_{0}$	80
Coord tuple 2 Coord 3	$T2C3_{63}$	$T2C3_{62}$	$T2C3_{61}$	$T2C3_{60}$	$T2C3_{59}$	$T2C3_{58}$	$T2C3_{57}$	$T2C3_{56}$	81
	$T2C3_{55}$	$T2C3_{54}$	$T2C3_{53}$	$T2C3_{52}$	$T2C3_{51}$	$T2C3_{50}$	$T2C3_{49}$	$T2C3_{48}$	82
	$T2C3_{47}$	$T2C3_{46}$	$T2C3_{45}$	$T2C3_{44}$	$T2C3_{43}$	$T2C3_{42}$	$T2C3_{41}$	$T2C3_{40}$	83
	$T2C3_{39}$	$T2C3_{38}$	$T2C3_{37}$	$T2C3_{36}$	$T2C3_{35}$	$T2C3_{34}$	$T2C3_{33}$	$T2C3_{32}$	84
	$T2C3_{31}$	$T2C3_{30}$	$T2C3_{29}$	$T2C3_{28}$	$T2C3_{27}$	$T2C3_{26}$	$T2C3_{25}$	$T2C3_{24}$	85
	$T2C3_{23}$	$T2C3_{22}$	$T2C3_{21}$	$T2C3_{20}$	$T2C3_{19}$	$T2C3_{18}$	$T2C3_{17}$	$T2C3_{16}$	86
	$T2C3_{15}$	$T2C3_{14}$	$T2C3_{13}$	$T2C3_{12}$	$T2C3_{11}$	$T2C3_{10}$	$T2C3_{9}$	$T2C3_{8}$	87
	$T2C3_{7}$	$T2C3_{6}$	$T2C3_{5}$	$T2C3_{4}$	$T2C3_{3}$	$T2C3_{2}$	$T2C3_{1}$	$T2C3_{0}$	88
Stanza ID 2	$ID2_{15}$	$ID2_{14}$	$ID2_{13}$	$ID2_{12}$	$ID2_{11}$	$ID2_{10}$	$ID2_{9}$	$ID2_{8}$	89
	$ID2_{7}$	$ID2_{6}$	$ID2_{5}$	$ID2_{4}$	$ID2_{3}$	$ID2_{2}$	$ID2_{1}$	$ID2_{0}$	90
Position 2 valid	$PV2_{7}$	$PV2_{6}$	$PV2_{5}$	$PV2_{4}$	$PV2_{3}$	$PV2_{2}$	$PV2_{1}$	$PV2_{0}$	91
Position 2 quality	$PQ2_{7}$	$PQ2_{6}$	$PQ2_{5}$	$PQ2_{4}$	$PQ2_{3}$	$PQ2_{2}$	$PQ2_{1}$	$PQ2_{0}$	92
Undefined	X	X	X	X	X	X	X	X	93
	X	X	X	X	X	X	X	X	94
	X	X	X	X	X	X	X	X	95
Header Blk Type	0	1	0	1	0	0	1	0	96

7.14 RELATIVE POSITION BLOCK (optional)

Bit No.	0	1	2	3	4	5	6	7	
Offset easting	OE_{31}	OE_{30}	OE_{29}	OE_{28}	OE_{27}	OE_{26}	OE_{25}	OE_{24}	1
	OE_{23}	OE_{22}	OE_{21}	OE_{20}	OE_{19}	OE_{18}	OE_{17}	OE_{16}	2
	OE_{15}	OE_{14}	OE_{13}	OE_{12}	OE_{11}	OE_{10}	OE_{9}	OE_{8}	3
	OE_{7}	OE_{6}	OE_{5}	OE_{4}	OE_{3}	OE_{2}	OE_{1}	OE_{0}	4
Offset northing	ON_{31}	ON_{30}	ON_{29}	ON_{28}	ON_{27}	ON_{26}	ON_{25}	ON_{24}	5
	ON_{23}	ON_{22}	ON_{21}	ON_{20}	ON_{19}	ON_{18}	ON_{17}	ON_{16}	6
	ON_{15}	ON_{14}	ON_{13}	ON_{12}	ON_{11}	ON_{10}	ON_{9}	ON_{8}	7
	ON_{7}	ON_{6}	ON_{5}	ON_{4}	ON_{3}	ON_{2}	ON_{1}	ON_{0}	8
Offset vertical	OD_{31}	OD_{30}	OD_{29}	OD_{28}	OD_{27}	OD_{26}	OD_{25}	OD_{24}	9
	OD_{23}	OD_{22}	OD_{21}	OD_{20}	OD_{19}	OD_{18}	OD_{17}	OD_{16}	10
	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_{9}	OD_{8}	11
	OD_{7}	OD_{6}	OD_{5}	OD_{4}	OD_{3}	OD_{2}	OD_{1}	OD_{0}	12
Description	DE	DE	DE	DE	DE	DE	DE	DE	13
	DE	DE	DE	DE	DE	DE	DE	DE	14
	DE	DE	DE	DE	DE	DE	DE	DE	15
	DE	DE	DE	DE	DE	DE	DE	DE	16
	DE	DE	DE	DE	DE	DE	DE	DE	17
	DE	DE	DE	DE	DE	DE	DE	DE	18
	DE	DE	DE	DE	DE	DE	DE	DE	19
	DE	DE	DE	DE	DE	DE	DE	DE	20
	DE	DE	DE	DE	DE	DE	DE	DE	21
	DE	DE	DE	DE	DE	DE	DE	DE	22
	DE	DE	DE	DE	DE	DE	DE	DE	23
	DE	DE	DE	DE	DE	DE	DE	DE	24
	DE	DE	DE	DE	DE	DE	DE	DE	25
	DE	DE	DE	DE	DE	DE	DE	DE	26
	DE	DE	DE	DE	DE	DE	DE	DE	27
	DE	DE	DE	DE	DE	DE	DE	DE	28
	DE	DE	DE	DE	DE	DE	DE	DE	29
	DE	DE	DE	DE	DE	DE	DE	DE	30
	DE	DE	DE	DE	DE	DE	DE	DE	31
Header Blk Type	0	1	0	1	0	1	1	0	32

7.15 SCAN TYPE HEADER (Channel Set Descriptor)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	
Binary Value MSB	128	64	32	16	8	4	2	1	
Scan Type No.	ST_1	ST_1	ST_1	ST_1	ST_2	ST_2	ST_2	ST_2	1

Channel Set No.	CS_{15}	CS_{14}	CS_{13}	CS_{12}	CS_{11}	CS_{10}	CS_9	CS_8	2
	CS_7	CS_6	CS_5	CS_4	CS_3	CS_2	CS_1	CS_0	3
Channel Type	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	4
Channel Set Start Time	TF_{31}	TF_{30}	TF_{29}	TF_{28}	TF_{27}	TF_{26}	TF_{25}	TF_{24}	5
	TF_{23}	TF_{22}	TF_{21}	TF_{20}	TF_{19}	TF_{18}	TF_{17}	TF_{16}	6
	TF_{15}	TF_{14}	TF_{13}	TF_{12}	TF_{11}	TF_{10}	TF_9	TF_8	7
	TF_7	TF_6	TF_5	TF_4	TF_3	TF_2	TF_1	TF_0	8
Channel Set End Time	TE_{31}	TE_{30}	TE_{29}	TE_{28}	TE_{27}	TE_{26}	TE_{25}	TE_{24}	9
	TE_{23}	TE_{22}	TE_{21}	TE_{20}	TE_{19}	TE_{18}	TE_{17}	TE_{16}	10
	TE_{15}	TE_{14}	TE_{13}	TE_{12}	TE_{11}	TE_{10}	TE_9	TE_8	11
	TE_7	TE_6	TE_5	TE_4	TE_3	TE_2	TE_1	TE_0	12
Number of Samples	NS_{31}	NS_{30}	NS_{29}	NS_{28}	NS_{27}	NS_{26}	NS_{25}	NS_{24}	13
	NS_{23}	NS_{22}	NS_{21}	NS_{20}	NS_{19}	NS_{18}	NS_{17}	NS_{16}	14
	NS_{15}	NS_{14}	NS_{13}	NS_{12}	NS_{11}	NS_{10}	NS_9	NS_8	15
	NS_7	NS_6	NS_5	NS_4	NS_3	NS_2	NS_1	NS_0	16
Descale Multiplier	DSM_{31}	DSM_{30}	DSM_{29}	DSM_{28}	DSM_{27}	DSM_{26}	DSM_{25}	DSM_{24}	17
	DSM_{23}	DSM_{22}	DSM_{21}	DSM_{20}	DSM_{19}	DSM_{18}	DSM_{17}	DSM_{16}	18
	DSM_{15}	DSM_{14}	DSM_{13}	DSM_{12}	DSM_{11}	DSM_{10}	DSM_9	DSM_8	19
	DSM_7	DSM_6	DSM_5	DSM_4	DSM_3	DSM_2	DSM_1	DSM_0	20
No. of Channels	C/S_{23}	C/S_{22}	C/S_{21}	C/S_{20}	C/S_{19}	C/S_{18}	C/S_{17}	C/S_{16}	21
	C/S_{15}	C/S_{14}	C/S_{13}	C/S_{12}	C/S_{11}	C/S_{10}	C/S_9	C/S_8	22
	C/S_7	C/S_6	C/S_5	C/S_4	C/S_3	C/S_2	C/S_1	C/S_0	23
Sampling Interval	SR_{23}	SR_{22}	SR_{21}	SR_{20}	SR_{19}	SR_{18}	SR_{17}	SR_{16}	24
	SR_{15}	SR_{14}	SR_{13}	SR_{12}	SR_{11}	SR_{10}	SR_9	SR_8	25
	SR_7	SR_6	SR_5	SR_4	SR_3	SR_2	SR_1	SR_0	26
Array Forming	ARY_7	ARY_6	ARY_5	ARY_4	ARY_3	ARY_2	ARY_1	ARY_0	27
Trace Header Extensions	THE_7	THE_6	THE_5	THE_4	THE_3	THE_2	THE_1	THE_0	28
Extended Header Flag/Channel Gain	EFH_3	EFH_2	EFH_1	EFH_0	J	J	J	J	29
Vertical Stack	VS_7	VS_6	VS_5	VS_4	VS_3	VS_2	VS_1	VS_0	30
Streamer No.	CAB_7	CAB_6	CAB_5	CAB_4	CAB_3	CAB_2	CAB_1	CAB_0	31
Header Block Type	0	0	1	1	0	0	0	0	32

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	
Binary Value MSB	128	64	32	16	8	4	2	1	
Alias Filter Frequency	AF_{31}	AF_{30}	AF_{29}	AF_{28}	AF_{27}	AF_{26}	AF_{25}	AF_{24}	33
	AF_{23}	AF_{22}	AF_{21}	AF_{20}	AF_{19}	AF_{18}	AF_{17}	AF_{16}	34
	AF_{15}	AF_{14}	AF_{13}	AF_{12}	AF_{11}	AF_{10}	AF_9	AF_8	35
	AF_7	AF_6	AF_5	AF_4	AF_3	AF_2	AF_1	AF_0	36
Low Cut Filter	LC_{31}	LC_{30}	LC_{29}	LC_{28}	LC_{27}	LC_{26}	LC_{25}	LC_{24}	37
	LC_{23}	LC_{22}	LC_{21}	LC_{20}	LC_{19}	LC_{18}	LC_{17}	LC_{16}	38
	LC_{15}	LC_{14}	LC_{13}	LC_{12}	LC_{11}	LC_{10}	LC_9	LC_8	39
	LC_7	LC_6	LC_5	LC_4	LC_3	LC_2	LC_1	LC_0	40
Alias Filter Slope	AS_{31}	AS_{30}	AS_{29}	AS_{28}	AS_{27}	AS_{26}	AS_{25}	AS_{24}	41
	AS_{23}	AS_{22}	AS_{21}	AS_{20}	AS_{19}	AS_{18}	AS_{17}	AS_{16}	42
	AS_{15}	AS_{14}	AS_{13}	AS_{12}	AS_{11}	AS_{10}	AS_9	AS_8	43
	AS_7	AS_6	AS_5	AS_4	AS_3	AS_2	AS_1	AS_0	44

Low Cut Filter Slope	LS_{31}	LS_{30}	LS_{29}	LS_{28}	LS_{27}	LS_{26}	LS_{25}	LS_{24}	45
	LS_{23}	LS_{22}	LS_{21}	LS_{20}	LS_{19}	LS_{18}	LS_{17}	LS_{16}	46
	LS_{15}	LS_{14}	LS_{13}	LS_{12}	LS_{11}	LS_{10}	LS_{9}	LS_{8}	47
	LS_{7}	LS_{6}	LS_{5}	LS_{4}	LS_{3}	LS_{2}	LS_{1}	LS_{0}	48
First Notch Filter	$NT1_{31}$	$NT1_{30}$	$NT1_{29}$	$NT1_{28}$	$NT1_{27}$	$NT1_{26}$	$NT1_{25}$	$NT1_{24}$	49
	$NT1_{23}$	$NT1_{22}$	$NT1_{21}$	$NT1_{20}$	$NT1_{19}$	$NT1_{18}$	$NT1_{17}$	$NT1_{16}$	50
	$NT1_{15}$	$NT1_{14}$	$NT1_{13}$	$NT1_{12}$	$NT1_{11}$	$NT1_{10}$	$NT1_{9}$	$NT1_{8}$	51
	$NT1_{7}$	$NT1_{6}$	$NT1_{5}$	$NT1_{4}$	$NT1_{3}$	$NT1_{2}$	$NT1_{1}$	$NT1_{0}$	52
Second Notch Filter	$NT2_{31}$	$NT2_{30}$	$NT2_{29}$	$NT2_{28}$	$NT2_{27}$	$NT2_{26}$	$NT2_{25}$	$NT2_{24}$	53
	$NT2_{23}$	$NT2_{22}$	$NT2_{21}$	$NT2_{20}$	$NT2_{19}$	$NT2_{18}$	$NT2_{17}$	$NT2_{16}$	54
	$NT2_{15}$	$NT2_{14}$	$NT2_{13}$	$NT2_{12}$	$NT2_{11}$	$NT2_{10}$	$NT2_{9}$	$NT2_{8}$	55
	$NT2_{7}$	$NT2_{6}$	$NT2_{5}$	$NT2_{4}$	$NT2_{3}$	$NT2_{2}$	$NT2_{1}$	$NT2_{0}$	56
Third Notch Filter	$NT3_{31}$	$NT3_{30}$	$NT3_{29}$	$NT3_{28}$	$NT3_{27}$	$NT3_{26}$	$NT3_{25}$	$NT3_{24}$	57
	$NT3_{23}$	$NT3_{22}$	$NT3_{21}$	$NT3_{20}$	$NT3_{19}$	$NT3_{18}$	$NT3_{17}$	$NT3_{16}$	58
	$NT3_{15}$	$NT3_{14}$	$NT3_{13}$	$NT3_{12}$	$NT3_{11}$	$NT3_{10}$	$NT3_{9}$	$NT3_{8}$	59
	$NT3_{7}$	$NT3_{6}$	$NT3_{5}$	$NT3_{4}$	$NT3_{3}$	$NT3_{2}$	$NT3_{1}$	$NT3_{0}$	60
Filter Phase	FPH_{7}	FPH_{6}	FPH_{5}	FPH_{4}	FPH_{3}	FPH_{2}	FPH_{1}	FPH_{0}	61
Physical Unit	PHU_{7}	PHU_{6}	PHU_{5}	PHU_{4}	PHU_{3}	PHU_{2}	PHU_{1}	PHU_{0}	62
	X	X	X	X	X	X	X	X	63
Header Block Type	0	0	1	1	0	0	0	1	64

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	
Binary Value MSB	128	64	32	16	8	4	2	1	
Filter delay	FDL_{31}	FDL_{30}	FDL_{29}	FDL_{28}	FDL_{27}	FDL_{26}	FDL_{25}	FDL_{24}	65
	FDL_{23}	FDL_{22}	FDL_{21}	FDL_{20}	FDL_{19}	FDL_{18}	FDL_{17}	FDL_{16}	66
	FDL_{15}	FDL_{14}	FDL_{13}	FDL_{12}	FDL_{11}	FDL_{10}	FDL_{9}	FDL_{8}	67
	FDL_{7}	FDL_{6}	FDL_{5}	FDL_{4}	FDL_{3}	FDL_{2}	FDL_{1}	FDL_{0}	68
Description	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	69
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	70
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	71
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	72
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	73
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	74
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	75
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	76
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	77
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	78
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	79
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	80
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	81
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	82
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	83
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	84
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	85
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	86
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	87
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	88

	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	89
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	90
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	91
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	92
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	93
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	94
	DSC	DSC	DSC	DSC	DSC	DSC	DSC	DSC	95
Header Block Type	0	0	1	1	0	0	1	0	96

7.16 DEMUX TRACE HEADER

Bit No.	0	1	2	3	4	5	6	7	
File Number	F_1	F_1	F_1	F_1	F_2	F_2	F_2	F_2	1
	F_3	F_3	F_3	F_3	F_4	F_4	F_4	F_4	2
Scan Type Number	ST_1	ST_1	ST_1	ST_1	ST_2	ST_2	ST_2	ST_2	3
Channel Set Number	CN_1	CN_1	CN_1	CN_1	CN_2	CN_2	CN_2	CN_2	4
Trace Number	TN_1	TN_1	TN_1	TN_1	TN_2	TN_2	TN_2	TN_2	5
	TN_3	TN_3	TN_3	TN_3	TN_4	TN_4	TN_4	TN_4	6
First Timing Word	T_{15}	T_{14}	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8	7
	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0	8
	T_{-1}	T_{-2}	T_{-3}	T_{-4}	T_{-5}	T_{-6}	T_{-7}	T_{-8}	9
Trace Header Extension	THE_7	THE_6	THE_5	THE_4	THE_3	THE_2	THE_1	THE_0	10
Sample Skew	SSK_{-1}	SSK_{-2}	SSK_{-3}	SSK_{-4}	SSK_{-5}	SSK_{-6}	SSK_{-7}	SSK_{-8}	11
Trace Edit	TR_7	TR_6	TR_5	TR_4	TR_3	TR_2	TR_1	TR_0	12
Time Break Window	TW_{15}	TW_{14}	TW_{13}	TW_{12}	TW_{11}	TW_{10}	TW_9	TW_8	13
	TW_7	TW_6	TW_5	TW_4	TW_3	TW_2	TW_1	TW_0	14
	TW_{-1}	TW_{-2}	TW_{-3}	TW_{-4}	TW_{-5}	TW_{-6}	TW_{-7}	TW_{-8}	15
Extended Channel Set Number	EN_{15}	EN_{14}	EN_{13}	EN_{12}	EN_{11}	EN_{10}	EN_9	EN_8	16
	EN_7	EN_6	EN_5	EN_4	EN_3	EN_2	EN_1	EN_0	17
Extended File Number	EFN_{23}	EFN_{22}	EFN_{21}	EFN_{20}	EFN_{19}	EFN_{18}	EFN_{17}	EFN_{16}	18
	EFN_{15}	EFN_{14}	EFN_{13}	EFN_{12}	EFN_{11}	EFN_{10}	EFN_9	EFN_8	19
	EFN_7	EFN_6	EFN_5	EFN_4	EFN_3	EFN_2	EFN_1	EFN_0	20

7.17 TRACE HEADER EXTENSION

Bit No.	0	1	2	3	4	5	6	7	
Receiver Line Number	RLN_{23}	RLN_{22}	RLN_{21}	RLN_{20}	RLN_{19}	RLN_{18}	RLN_{17}	RLN_{16}	1
	RLN_{15}	RLN_{14}	RLN_{13}	RLN_{12}	RLN_{11}	RLN_{10}	RLN_{9}	RLN_{8}	2
	RLN_{7}	RLN_{6}	RLN_{5}	RLN_{4}	RLN_{3}	RLN_{2}	RLN_{1}	RLN_{0}	3
Receiver Point Number	RPN_{23}	RPN_{22}	RPN_{21}	RPN_{20}	RPN_{19}	RPN_{18}	RPN_{17}	RPN_{16}	4
	RPN_{15}	RPN_{14}	RPN_{13}	RPN_{12}	RPN_{11}	RPN_{10}	RPN_{9}	RPN_{8}	5
	RPN_{7}	RPN_{6}	RPN_{5}	RPN_{4}	RPN_{3}	RPN_{2}	RPN_{1}	RPN_{0}	6
Receiver Point Index	RPI_{7}	RPI_{6}	RPI_{5}	RPI_{4}	RPI_{3}	RPI_{2}	RPI_{1}	RPI_{0}	7
Reshoot Index	RI_{7}	RI_{6}	RI_{5}	RI_{4}	RI_{3}	RI_{2}	RI_{1}	RI_{0}	8
Group Index	GI_{7}	GI_{6}	GI_{5}	GI_{4}	GI_{3}	GI_{2}	GI_{1}	GI_{0}	9
Depth Index	DI_{7}	DI_{6}	DI_{5}	DI_{4}	DI_{3}	DI_{2}	DI_{1}	DI_{0}	10
Extended Receiver Line Number	$ERLN_{23}$	$ERLN_{22}$	$ERLN_{21}$	$ERLN_{20}$	$ERLN_{19}$	$ERLN_{18}$	$ERLN_{17}$	$ERLN_{16}$	11
	$ERLN_{15}$	$ERLN_{14}$	$ERLN_{13}$	$ERLN_{12}$	$ERLN_{11}$	$ERLN_{10}$	$ERLN_{9}$	$ERLN_{8}$	12
	$ERLN_{7}$	$ERLN_{6}$	$ERLN_{5}$	$ERLN_{4}$	$ERLN_{3}$	$ERLN_{2}$	$ERLN_{1}$	$ERLN_{0}$	13
	$ERLN_{-1}$	$ERLN_{-2}$	$ERLN_{-3}$	$ERLN_{-4}$	$ERLN_{-5}$	$ERLN_{-6}$	$ERLN_{-7}$	$ERLN_{-8}$	14
	$ERLN_{-9}$	$ERLN_{-10}$	$ERLN_{-11}$	$ERLN_{-12}$	$ERLN_{-13}$	$ERLN_{-14}$	$ERLN_{-15}$	$ERLN_{-16}$	15
Extended Receiver Point #	$ERPN_{23}$	$ERPN_{22}$	$ERPN_{21}$	$ERPN_{20}$	$ERPN_{19}$	$ERPN_{18}$	$ERPN_{17}$	$ERPN_{16}$	16
	$ERPN_{15}$	$ERPN_{14}$	$ERPN_{13}$	$ERPN_{12}$	$ERPN_{11}$	$ERPN_{10}$	$ERPN_{9}$	$ERPN_{8}$	17
	$ERPN_{7}$	$ERPN_{6}$	$ERPN_{5}$	$ERPN_{4}$	$ERPN_{3}$	$ERPN_{2}$	$ERPN_{1}$	$ERPN_{0}$	18
	$ERPN_{-1}$	$ERPN_{-2}$	$ERPN_{-3}$	$ERPN_{-4}$	$ERPN_{-5}$	$ERPN_{-6}$	$ERPN_{-7}$	$ERPN_{-8}$	19
	$ERPN_{-9}$	$ERPN_{-10}$	$ERPN_{-11}$	$ERPN_{-12}$	$ERPN_{-13}$	$ERPN_{-14}$	$ERPN_{-15}$	$ERPN_{-16}$	20
Sensor Type	SEN_{7}	SEN_{6}	SEN_{5}	SEN_{4}	SEN_{3}	SEN_{2}	SEN_{1}	SEN_{0}	21
Extended Trace Number	ETN_{23}	ETN_{22}	ETN_{21}	ETN_{20}	ETN_{19}	ETN_{18}	ETN_{17}	ETN_{16}	22
	ETN_{15}	ETN_{14}	ETN_{13}	ETN_{12}	ETN_{11}	ETN_{10}	ETN_{9}	ETN_{8}	23
	ETN_{7}	ETN_{6}	ETN_{5}	ETN_{4}	ETN_{3}	ETN_{2}	ETN_{1}	ETN_{0}	24
# of Samples per Trace	NS_{31}	NS_{30}	NS_{29}	NS_{28}	NS_{27}	NS_{26}	NS_{25}	NS_{24}	25
	NS_{23}	NS_{22}	NS_{21}	NS_{20}	NS_{19}	NS_{18}	NS_{17}	NS_{16}	26
	NS_{15}	NS_{14}	NS_{13}	NS_{12}	NS_{11}	NS_{10}	NS_{9}	NS_{8}	27
	NS_{7}	NS_{6}	NS_{5}	NS_{4}	NS_{3}	NS_{2}	NS_{1}	NS_{0}	28
Sensor Moving	MV_{7}	MV_{6}	MV_{5}	MV_{4}	MV_{3}	MV_{2}	MV_{1}	MV_{0}	29
Undefined	X	X	X	X	X	X	X	X	30
Physical Unit	PHU_{7}	PHU_{6}	PHU_{5}	PHU_{4}	PHU_{3}	PHU_{2}	PHU_{1}	PHU_{0}	31
Header Blk Type	0	1	0	0	0	0	0	0	32

7.18 SENSOR INFO TRACE HEADER EXTENSION (optional)

Bit No.	0	1	2	3	4	5	6	7	
Instrument Test Time	ITT_{63}	ITT_{62}	ITT_{61}	ITT_{60}	ITT_{59}	ITT_{58}	ITT_{57}	ITT_{56}	1
	ITT_{55}	ITT_{54}	ITT_{53}	ITT_{52}	ITT_{51}	ITT_{50}	ITT_{49}	ITT_{48}	2
	ITT_{47}	ITT_{46}	ITT_{45}	ITT_{44}	ITT_{43}	ITT_{42}	ITT_{41}	ITT_{40}	3
	ITT_{39}	ITT_{38}	ITT_{37}	ITT_{36}	ITT_{35}	ITT_{34}	ITT_{33}	ITT_{32}	4
	ITT_{31}	ITT_{30}	ITT_{29}	ITT_{28}	ITT_{27}	ITT_{26}	ITT_{25}	ITT_{24}	5
	ITT_{23}	ITT_{22}	ITT_{21}	ITT_{20}	ITT_{19}	ITT_{18}	ITT_{17}	ITT_{16}	6
	ITT_{15}	ITT_{14}	ITT_{13}	ITT_{12}	ITT_{11}	ITT_{10}	ITT_{9}	ITT_{8}	7

	ITT_7	ITT_6	ITT_5	ITT_4	ITT_3	ITT_2	ITT_1	ITT_0	8
Sensor Sensitivity	STY_{31}	STY_{30}	STY_{29}	STY_{28}	STY_{27}	STY_{26}	STY_{25}	STY_{24}	9
	STY_{23}	STY_{22}	STY_{21}	STY_{20}	STY_{19}	STY_{18}	STY_{17}	STY_{16}	10
	STY_{15}	STY_{14}	STY_{13}	STY_{12}	STY_{11}	STY_{10}	STY_9	STY_8	11
	STY_7	STY_6	STY_5	STY_4	STY_3	STY_2	STY_1	STY_0	12
Instr Test Result	ITR_7	ITR_6	ITR_5	ITR_4	ITR_3	ITR_2	ITR_1	ITR_0	13
Serial Number	SN	SN	SN	SN	SN	SN	SN	SN	14
	SN	SN	SN	SN	SN	SN	SN	SN	15
	SN	SN	SN	SN	SN	SN	SN	SN	16
	SN	SN	SN	SN	SN	SN	SN	SN	17
	SN	SN	SN	SN	SN	SN	SN	SN	18
	SN	SN	SN	SN	SN	SN	SN	SN	19
	SN	SN	SN	SN	SN	SN	SN	SN	20
	SN	SN	SN	SN	SN	SN	SN	SN	21
	SN	SN	SN	SN	SN	SN	SN	SN	22
	SN	SN	SN	SN	SN	SN	SN	SN	23
	SN	SN	SN	SN	SN	SN	SN	SN	24
	SN	SN	SN	SN	SN	SN	SN	SN	25
	SN	SN	SN	SN	SN	SN	SN	SN	26
	SN	SN	SN	SN	SN	SN	SN	SN	27
	SN	SN	SN	SN	SN	SN	SN	SN	28
	SN	SN	SN	SN	SN	SN	SN	SN	29
	SN	SN	SN	SN	SN	SN	SN	SN	30
	SN	SN	SN	SN	SN	SN	SN	SN	31
Header Blk Type	0	1	0	0	0	0	0	1	32

7.19 TIMESTAMP HEADER (optional)

Bit No.	0	1	2	3	4	5	6	7	
Time Zero of Data	TZD_{63}	TZD_{62}	TZD_{61}	TZD_{60}	TZD_{59}	TZD_{58}	TZD_{57}	TZD_{56}	1
	TZD_{55}	TZD_{54}	TZD_{53}	TZD_{52}	TZD_{51}	TZD_{50}	TZD_{49}	TZD_{48}	2
	TZD_{47}	TZD_{46}	TZD_{45}	TZD_{44}	TZD_{43}	TZD_{42}	TZD_{41}	TZD_{40}	3
	TZD_{39}	TZD_{38}	TZD_{37}	TZD_{36}	TZD_{35}	TZD_{34}	TZD_{33}	TZD_{32}	4
	TZD_{31}	TZD_{30}	TZD_{29}	TZD_{28}	TZD_{27}	TZD_{26}	TZD_{25}	TZD_{24}	5
	TZD_{23}	TZD_{22}	TZD_{21}	TZD_{20}	TZD_{19}	TZD_{18}	TZD_{17}	TZD_{16}	6
	TZD_{15}	TZD_{14}	TZD_{13}	TZD_{12}	TZD_{11}	TZD_{10}	TZD_9	TZD_8	7
	TZD_7	TZD_6	TZD_5	TZD_4	TZD_3	TZD_2	TZD_1	TZD_0	8
Undefined	X	X	X	X	X	X	X	X	9
	X	X	X	X	X	X	X	X	10
	X	X	X	X	X	X	X	X	11
	X	X	X	X	X	X	X	X	12
	X	X	X	X	X	X	X	X	13
	X	X	X	X	X	X	X	X	14
	X	X	X	X	X	X	X	X	15
	X	X	X	X	X	X	X	X	16
	X	X	X	X	X	X	X	X	17
	X	X	X	X	X	X	X	X	18
	X	X	X	X	X	X	X	X	19

	X	X	X	X	X	X	X	X	20
	X	X	X	X	X	X	X	X	21
	X	X	X	X	X	X	X	X	22
	X	X	X	X	X	X	X	X	23
	X	X	X	X	X	X	X	X	24
	X	X	X	X	X	X	X	X	25
	X	X	X	X	X	X	X	X	26
	X	X	X	X	X	X	X	X	27
	X	X	X	X	X	X	X	X	28
	X	X	X	X	X	X	X	X	29
	X	X	X	X	X	X	X	X	30
	X	X	X	X	X	X	X	X	31
Header Blk Type	0	1	0	0	0	0	1	0	32

7.20 SENSOR CALIBRATION HEADER (optional)

Bit No.	0	1	2	3	4	5	6	7	
Frequency 1	$FR1_{31}$	$FR1_{30}$	$FR1_{29}$	$FR1_{28}$	$FR1_{27}$	$FR1_{26}$	$FR1_{25}$	$FR1_{24}$	1
	$FR1_{23}$	$FR1_{22}$	$FR1_{21}$	$FR1_{20}$	$FR1_{19}$	$FR1_{18}$	$FR1_{17}$	$FR1_{16}$	2
	$FR1_{15}$	$FR1_{14}$	$FR1_{13}$	$FR1_{12}$	$FR1_{11}$	$FR1_{10}$	$FR1_{9}$	$FR1_{8}$	3
	$FR1_{7}$	$FR1_{6}$	$FR1_{5}$	$FR1_{4}$	$FR1_{3}$	$FR1_{2}$	$FR1_{1}$	$FR1_{0}$	4
Amplitude 1	$AM1_{31}$	$AM1_{30}$	$AM1_{29}$	$AM1_{28}$	$AM1_{27}$	$AM1_{26}$	$AM1_{25}$	$AM1_{24}$	5
	$AM1_{23}$	$AM1_{22}$	$AM1_{21}$	$AM1_{20}$	$AM1_{19}$	$AM1_{18}$	$AM1_{17}$	$AM1_{16}$	6
	$AM1_{15}$	$AM1_{14}$	$AM1_{13}$	$AM1_{12}$	$AM1_{11}$	$AM1_{10}$	$AM1_{9}$	$AM1_{8}$	7
	$AM1_{7}$	$AM1_{6}$	$AM1_{5}$	$AM1_{4}$	$AM1_{3}$	$AM1_{2}$	$AM1_{1}$	$AM1_{0}$	8
Phase 1	$PH1_{31}$	$PH1_{30}$	$PH1_{29}$	$PH1_{28}$	$PH1_{27}$	$PH1_{26}$	$PH1_{25}$	$PH1_{24}$	9
	$PH1_{23}$	$PH1_{22}$	$PH1_{21}$	$PH1_{20}$	$PH1_{19}$	$PH1_{18}$	$PH1_{17}$	$PH1_{16}$	10
	$PH1_{15}$	$PH1_{14}$	$PH1_{13}$	$PH1_{12}$	$PH1_{11}$	$PH1_{10}$	$PH1_{9}$	$PH1_{8}$	11
	$PH1_{7}$	$PH1_{6}$	$PH1_{5}$	$PH1_{4}$	$PH1_{3}$	$PH1_{2}$	$PH1_{1}$	$PH1_{0}$	12
Frequency 2	$FR2_{31}$	$FR2_{30}$	$FR2_{29}$	$FR2_{28}$	$FR2_{27}$	$FR2_{26}$	$FR2_{25}$	$FR2_{24}$	13
	$FR2_{23}$	$FR2_{22}$	$FR2_{21}$	$FR2_{20}$	$FR2_{19}$	$FR2_{18}$	$FR2_{17}$	$FR2_{16}$	14
	$FR2_{15}$	$FR2_{14}$	$FR2_{13}$	$FR2_{12}$	$FR2_{11}$	$FR2_{10}$	$FR2_{9}$	$FR2_{8}$	15
	$FR2_{7}$	$FR2_{6}$	$FR2_{5}$	$FR2_{4}$	$FR2_{3}$	$FR2_{2}$	$FR2_{1}$	$FR2_{0}$	16
Amplitude 2	$AM2_{31}$	$AM2_{30}$	$AM2_{29}$	$AM2_{28}$	$AM2_{27}$	$AM2_{26}$	$AM2_{25}$	$AM2_{24}$	17
	$AM2_{23}$	$AM2_{22}$	$AM2_{21}$	$AM2_{20}$	$AM2_{19}$	$AM2_{18}$	$AM2_{17}$	$AM2_{16}$	18
	$AM2_{15}$	$AM2_{14}$	$AM2_{13}$	$AM2_{12}$	$AM2_{11}$	$AM2_{10}$	$AM2_{9}$	$AM2_{8}$	19
	$AM2_{7}$	$AM2_{6}$	$AM2_{5}$	$AM2_{4}$	$AM2_{3}$	$AM2_{2}$	$AM2_{1}$	$AM2_{0}$	20
Phase 2	$PH2_{31}$	$PH2_{30}$	$PH2_{29}$	$PH2_{28}$	$PH2_{27}$	$PH2_{26}$	$PH2_{25}$	$PH2_{24}$	21
	$PH2_{23}$	$PH2_{22}$	$PH2_{21}$	$PH2_{20}$	$PH2_{19}$	$PH2_{18}$	$PH2_{17}$	$PH2_{16}$	22
	$PH2_{15}$	$PH2_{14}$	$PH2_{13}$	$PH2_{12}$	$PH2_{11}$	$PH2_{10}$	$PH2_{9}$	$PH2_{8}$	23
	$PH2_{7}$	$PH2_{6}$	$PH2_{5}$	$PH2_{4}$	$PH2_{3}$	$PH2_{2}$	$PH2_{1}$	$PH2_{0}$	24
Calibration applied	CA_{7}	CA_{6}	CA_{5}	CA_{4}	CA_{3}	CA_{2}	CA_{1}	CA_{0}	25
Undefined	X	X	X	X	X	X	X	X	26
	X	X	X	X	X	X	X	X	27
	X	X	X	X	X	X	X	X	28
	X	X	X	X	X	X	X	X	29

	X	X	X	X	X	X	X	X	30
	X	X	X	X	X	X	X	X	31
Header Blk Type	0	1	0	0	0	0	1	1	32

7.21 TIME DRIFT HEADER (optional)

Bit No.	0	1	2	3	4	5	6	7	
Time of deployment	TD_{63}	TD_{62}	TD_{61}	TD_{60}	TD_{59}	TD_{58}	TD_{57}	TD_{56}	1
	TD_{55}	TD_{54}	TD_{53}	TD_{52}	TD_{51}	TD_{50}	TD_{49}	TD_{48}	2
	TD_{47}	TD_{46}	TD_{45}	TD_{44}	TD_{43}	TD_{42}	TD_{41}	TD_{40}	3
	TD_{39}	TD_{38}	TD_{37}	TD_{36}	TD_{35}	TD_{34}	TD_{33}	TD_{32}	4
	TD_{31}	TD_{30}	TD_{29}	TD_{28}	TD_{27}	TD_{26}	TD_{25}	TD_{24}	5
	TD_{23}	TD_{22}	TD_{21}	TD_{20}	TD_{19}	TD_{18}	TD_{17}	TD_{16}	6
	TD_{15}	TD_{14}	TD_{13}	TD_{12}	TD_{11}	TD_{10}	TD_9	TD_8	7
	TD_7	TD_6	TD_5	TD_4	TD_3	TD_2	TD_1	TD_0	8
Time of retrieval	TR_{63}	TR_{62}	TR_{61}	TR_{60}	TR_{59}	TR_{58}	TR_{57}	TR_{56}	9
	TR_{55}	TR_{54}	TR_{53}	TR_{52}	TR_{51}	TR_{50}	TR_{49}	TR_{48}	10
	TR_{47}	TR_{46}	TR_{45}	TR_{44}	TR_{43}	TR_{42}	TR_{41}	TR_{40}	11
	TR_{39}	TR_{38}	TR_{37}	TR_{36}	TR_{35}	TR_{34}	TR_{33}	TR_{32}	12
	TR_{31}	TR_{30}	TR_{29}	TR_{28}	TR_{27}	TR_{26}	TR_{25}	TR_{24}	13
	TR_{23}	TR_{22}	TR_{21}	TR_{20}	TR_{19}	TR_{18}	TR_{17}	TR_{16}	14
	TR_{15}	TR_{14}	TR_{13}	TR_{12}	TR_{11}	TR_{10}	TR_9	TR_8	15
	TR_7	TR_6	TR_5	TR_4	TR_3	TR_2	TR_1	TR_0	16
Timer Offset Deployment	OD_{31}	OD_{30}	OD_{29}	OD_{28}	OD_{27}	OD_{26}	OD_{25}	OD_{24}	17
	OD_{23}	OD_{22}	OD_{21}	OD_{20}	OD_{19}	OD_{18}	OD_{17}	OD_{16}	18
	OD_{15}	OD_{14}	OD_{13}	OD_{12}	OD_{11}	OD_{10}	OD_9	OD_8	19
	OD_7	OD_6	OD_5	OD_4	OD_3	OD_2	OD_1	OD_0	20
Time Offset Retrieval	OR_{31}	OR_{30}	OR_{29}	OR_{28}	OR_{27}	OR_{26}	OR_{25}	OR_{24}	21
	OR_{23}	OR_{22}	OR_{21}	OR_{20}	OR_{19}	OR_{18}	OR_{17}	OR_{16}	22
	OR_{15}	OR_{14}	OR_{13}	OR_{12}	OR_{11}	OR_{10}	OR_9	OR_8	23
	OR_7	OR_6	OR_5	OR_4	OR_3	OR_2	OR_1	OR_0	24
Timedrift corrected	TDC_7	TDC_6	TDC_5	TDC_4	TDC_3	TDC_2	TDC_1	TDC_0	25
Correction method	CM_7	CM_6	CM_5	CM_4	CM_3	CM_2	CM_1	CM_0	26
Undefined	X	X	X	X	X	X	X	X	27
	X	X	X	X	X	X	X	X	28
	X	X	X	X	X	X	X	X	29
	X	X	X	X	X	X	X	X	30
	X	X	X	X	X	X	X	X	31
Header Blk Type	0	1	0	0	0	1	0	0	32

7.22 ORIENTATION HEADER (optional)

Bit No.	0	1	2	3	4	5	6	7	
Rotation X axis	RX_{31}	RX_{30}	RX_{29}	RX_{28}	RX_{27}	RX_{26}	RX_{25}	RX_{24}	1
	RX_{23}	RX_{22}	RX_{21}	RX_{20}	RX_{19}	RX_{18}	RX_{17}	RX_{16}	2

	RX_{15}	RX_{14}	RX_{13}	RX_{12}	RX_{11}	RX_{10}	RX_9	RX_8	3
	RX_7	RX_6	RX_5	RX_4	RX_3	RX_2	RX_1	RX_0	4
Rotation Y axis	RY_{31}	RY_{30}	RY_{29}	RY_{28}	RY_{27}	RY_{26}	RY_{25}	RY_{24}	5
	RY_{23}	RY_{22}	RY_{21}	RY_{20}	RY_{19}	RY_{18}	RY_{17}	RY_{16}	6
	RY_{15}	RY_{14}	RY_{13}	RY_{12}	RY_{11}	RY_{10}	RY_9	RY_8	7
	RY_7	RY_6	RY_5	RY_4	RY_3	RY_2	RY_1	RY_0	8
Rotation Z axis	RZ_{31}	RZ_{30}	RZ_{29}	RZ_{28}	RZ_{27}	RZ_{26}	RZ_{25}	RZ_{24}	9
	RZ_{23}	RZ_{22}	RZ_{21}	RZ_{20}	RZ_{19}	RZ_{18}	RZ_{17}	RZ_{16}	10
	RZ_{15}	RZ_{14}	RZ_{13}	RZ_{12}	RZ_{11}	RZ_{10}	RZ_9	RZ_8	11
	RZ_7	RZ_6	RZ_5	RZ_4	RZ_3	RZ_2	RZ_1	RZ_0	12
Reference Orientation	RO_{31}	RO_{30}	RO_{29}	RO_{28}	RO_{27}	RO_{26}	RO_{25}	RO_{24}	13
	RO_{23}	RO_{22}	RO_{21}	RO_{20}	RO_{19}	RO_{18}	RO_{17}	RO_{16}	14
	RO_{15}	RO_{14}	RO_{13}	RO_{12}	RO_{11}	RO_{10}	RO_9	RO_8	15
	RO_7	RO_6	RO_5	RO_4	RO_3	RO_2	RO_1	RO_0	16
Time Stamp	TS_{63}	TS_{62}	TS_{61}	TS_{60}	TS_{59}	TS_{58}	TS_{57}	TS_{56}	17
	TS_{55}	TS_{54}	TS_{53}	TS_{52}	TS_{51}	TS_{50}	TS_{49}	TS_{48}	18
	TS_{47}	TS_{46}	TS_{45}	TS_{44}	TS_{43}	TS_{42}	TS_{41}	TS_{40}	19
	TS_{39}	TS_{38}	TS_{37}	TS_{36}	TS_{35}	TS_{34}	TS_{33}	TS_{32}	20
	TS_{31}	TS_{30}	TS_{29}	TS_{28}	TS_{27}	TS_{26}	TS_{25}	TS_{24}	21
	TS_{23}	TS_{22}	TS_{21}	TS_{20}	TS_{19}	TS_{18}	TS_{17}	TS_{16}	22
	TS_{15}	TS_{14}	TS_{13}	TS_{12}	TS_{11}	TS_{10}	TS_9	TS_8	23
	TS_7	TS_6	TS_5	TS_4	TS_3	TS_2	TS_1	TS_0	24
Orientation Type	OT_7	OT_6	OT_5	OT_4	OT_3	OT_2	OT_1	OT_0	25
Ref Orient Valid	ROV_7	ROV_6	ROV_5	ROV_4	ROV_3	ROV_2	ROV_1	ROV_0	26
Rotation Applied	RA_7	RA_6	RA_5	RA_4	RA_3	RA_2	RA_1	RA_0	27
Rot North Applied	RNA_7	RNA_6	RNA_5	RNA_4	RNA_3	RNA_2	RNA_1	RNA_0	28
	X	X	X	X	X	X	X	X	29
	X	X	X	X	X	X	X	X	30
	X	X	X	X	X	X	X	X	31
Header Blk Type	0	1	1	0	0	0	0	0	32

7.23 MEASUREMENT BLOCK (optional)

NOTE: This block should not be used until final agreement with Energistics is reached regarding the unit of measurement system, and the proper resources are available.

Bit No.	0	1	2	3	4	5	6	7	
Timestamp	TS_{63}	TS_{62}	TS_{61}	TS_{60}	TS_{59}	TS_{58}	TS_{57}	TS_{56}	1
	TS_{55}	TS_{54}	TS_{53}	TS_{52}	TS_{51}	TS_{50}	TS_{49}	TS_{48}	2
	TS_{47}	TS_{46}	TS_{45}	TS_{44}	TS_{43}	TS_{42}	TS_{41}	TS_{40}	3
	TS_{39}	TS_{38}	TS_{37}	TS_{36}	TS_{35}	TS_{34}	TS_{33}	TS_{32}	4
	TS_{31}	TS_{30}	TS_{29}	TS_{28}	TS_{27}	TS_{26}	TS_{25}	TS_{24}	5
	TS_{23}	TS_{22}	TS_{21}	TS_{20}	TS_{19}	TS_{18}	TS_{17}	TS_{16}	6
	TS_{15}	TS_{14}	TS_{13}	TS_{12}	TS_{11}	TS_{10}	TS_9	TS_8	7
	TS_7	TS_6	TS_5	TS_4	TS_3	TS_2	TS_1	TS_0	8
Measurement Value 1	$VA1_{31}$	$VA1_{30}$	$VA1_{29}$	$VA1_{28}$	$VA1_{27}$	$VA1_{26}$	$VA1_{25}$	$VA1_{24}$	9
	$VA1_{23}$	$VA1_{22}$	$VA1_{21}$	$VA1_{20}$	$VA1_{19}$	$VA1_{18}$	$VA1_{17}$	$VA1_{16}$	10

	$VA1_{15}$	$VA1_{14}$	$VA1_{13}$	$VA1_{12}$	$VA1_{11}$	$VA1_{10}$	$VA1_{9}$	$VA1_{8}$	11
	$VA1_{7}$	$VA1_{6}$	$VA1_{5}$	$VA1_{4}$	$VA1_{3}$	$VA1_{2}$	$VA1_{1}$	$VA1_{0}$	12
Unit of Measure 1	$UOM1_{15}$	$UOM1_{14}$	$UOM1_{13}$	$UOM1_{12}$	$UOM1_{11}$	$UOM1_{10}$	$UOM1_{9}$	$UOM1_{8}$	13
	$UOM1_{7}$	$UOM1_{6}$	$UOM1_{5}$	$UOM1_{4}$	$UOM1_{3}$	$UOM1_{2}$	$UOM1_{1}$	$UOM1_{0}$	14
Measure Type 1	$MT1_{7}$	$MT1_{6}$	$MT1_{5}$	$MT1_{4}$	$MT1_{3}$	$MT1_{2}$	$MT1_{1}$	$MT1_{0}$	15
Measurement Value 2	$VA2_{31}$	$VA2_{30}$	$VA2_{29}$	$VA2_{28}$	$VA2_{27}$	$VA2_{26}$	$VA2_{25}$	$VA2_{24}$	16
	$VA2_{23}$	$VA2_{22}$	$VA2_{21}$	$VA2_{20}$	$VA2_{19}$	$VA2_{18}$	$VA2_{17}$	$VA2_{16}$	17
	$VA2_{15}$	$VA2_{14}$	$VA2_{13}$	$VA2_{12}$	$VA2_{11}$	$VA2_{10}$	$VA2_{9}$	$VA2_{8}$	18
	$VA2_{7}$	$VA2_{6}$	$VA2_{5}$	$VA2_{4}$	$VA2_{3}$	$VA2_{2}$	$VA2_{1}$	$VA2_{0}$	19
Unit of Measure 2	$UOM2_{15}$	$UOM2_{14}$	$UOM2_{13}$	$UOM2_{12}$	$UOM2_{11}$	$UOM2_{10}$	$UOM2_{9}$	$UOM2_{8}$	20
	$UOM2_{7}$	$UOM2_{6}$	$UOM2_{5}$	$UOM2_{4}$	$UOM2_{3}$	$UOM2_{2}$	$UOM2_{1}$	$UOM2_{0}$	21
Measure Type 2	$MT2_{7}$	$MT2_{6}$	$MT2_{5}$	$MT2_{4}$	$MT2_{3}$	$MT2_{2}$	$MT2_{1}$	$MT2_{0}$	22
Measurement Value 3	$VA3_{31}$	$VA3_{30}$	$VA3_{29}$	$VA3_{28}$	$VA3_{27}$	$VA3_{26}$	$VA3_{25}$	$VA3_{24}$	23
	$VA3_{23}$	$VA3_{22}$	$VA3_{21}$	$VA3_{20}$	$VA3_{19}$	$VA3_{18}$	$VA3_{17}$	$VA3_{16}$	24
	$VA3_{15}$	$VA3_{14}$	$VA3_{13}$	$VA3_{12}$	$VA3_{11}$	$VA3_{10}$	$VA3_{9}$	$VA3_{8}$	25
	$VA3_{7}$	$VA3_{6}$	$VA3_{5}$	$VA3_{4}$	$VA3_{3}$	$VA3_{2}$	$VA3_{1}$	$VA3_{0}$	26
Unit of Measure 3	$UOM3_{15}$	$UOM3_{14}$	$UOM3_{13}$	$UOM3_{12}$	$UOM3_{11}$	$UOM3_{10}$	$UOM3_{9}$	$UOM3_{8}$	27
	$UOM3_{7}$	$UOM3_{6}$	$UOM3_{5}$	$UOM3_{4}$	$UOM3_{3}$	$UOM3_{2}$	$UOM3_{1}$	$UOM3_{0}$	28
Measure Type 3	$MT3_{7}$	$MT3_{6}$	$MT3_{5}$	$MT3_{4}$	$MT3_{3}$	$MT3_{2}$	$MT3_{1}$	$MT3_{0}$	29
Undefined	X	X	X	X	X	X	X	X	30
	X	X	X	X	X	X	X	X	31
Header Blk Type	0	1	1	0	0	0	0	1	32

7.24 ELECTROMAGNETIC SRC/RECV DESC BLOCK (optional)

Bit No.	0	1	2	3	4	5	6	7	
Equipment Dimension X	LX_{23}	LX_{22}	LX_{21}	LX_{20}	LX_{19}	LX_{18}	LX_{17}	LX_{16}	1
	LX_{15}	LX_{14}	LX_{13}	LX_{12}	LX_{11}	LX_{10}	LX_{9}	LX_{8}	2
	LX_{7}	LX_{6}	LX_{5}	LX_{4}	LX_{3}	LX_{2}	LX_{1}	LX_{0}	3
Equipment Dimension Y	LY_{23}	LY_{22}	LY_{21}	LY_{20}	LY_{19}	LY_{18}	LY_{17}	LY_{16}	4
	LY_{15}	LY_{14}	LY_{13}	LY_{12}	LY_{11}	LY_{10}	LY_{9}	LY_{8}	5
	LY_{7}	LY_{6}	LY_{5}	LY_{4}	LY_{3}	LY_{2}	LY_{1}	LY_{0}	6
Equipment Dimension Z	LZ_{23}	LZ_{22}	LZ_{21}	LZ_{20}	LZ_{19}	LZ_{18}	LZ_{17}	LZ_{16}	7
	LZ_{15}	LZ_{14}	LZ_{13}	LZ_{12}	LZ_{11}	LZ_{10}	LZ_{9}	LZ_{8}	8
	LZ_{7}	LZ_{6}	LZ_{5}	LZ_{4}	LZ_{3}	LZ_{2}	LZ_{1}	LZ_{0}	9
Positive terminal	PT_{7}	PT_{6}	PT_{5}	PT_{4}	PT_{3}	PT_{2}	PT_{1}	PT_{0}	10
Equipment Offset X	OX_{23}	OX_{22}	OX_{21}	OX_{20}	OX_{19}	OX_{18}	OX_{17}	OX_{16}	11
	OX_{15}	OX_{14}	OX_{13}	OX_{12}	OX_{11}	OX_{10}	OX_{9}	OX_{8}	12
	OX_{7}	OX_{6}	OX_{5}	OX_{4}	OX_{3}	OX_{2}	OX_{1}	OX_{0}	13
Equipment Offset Y	OY_{23}	OY_{22}	OY_{21}	OY_{20}	OY_{19}	OY_{18}	OY_{17}	OY_{16}	14
	OY_{15}	OY_{14}	OY_{13}	OY_{12}	OY_{11}	OY_{10}	OY_{9}	OY_{8}	15
	OY_{7}	OY_{6}	OY_{5}	OY_{4}	OY_{3}	OY_{2}	OY_{1}	OY_{0}	16
Equipment Offset Z	OZ_{23}	OZ_{22}	OZ_{21}	OZ_{20}	OZ_{19}	OZ_{18}	OZ_{17}	OZ_{16}	17
	OZ_{15}	OZ_{14}	OZ_{13}	OZ_{12}	OZ_{11}	OZ_{10}	OZ_{9}	OZ_{8}	18
	OZ_{7}	OZ_{6}	OZ_{5}	OZ_{4}	OZ_{3}	OZ_{2}	OZ_{1}	OZ_{0}	19
Undefined	X	X	X	X	X	X	X	X	20
	X	X	X	X	X	X	X	X	21
	X	X	X	X	X	X	X	X	22
	X	X	X	X	X	X	X	X	23

	X	X	X	X	X	X	X	X	24
	X	X	X	X	X	X	X	X	25
	X	X	X	X	X	X	X	X	26
	X	X	X	X	X	X	X	X	27
	X	X	X	X	X	X	X	X	28
	X	X	X	X	X	X	X	X	29
	X	X	X	X	X	X	X	X	30
	X	X	X	X	X	X	X	X	31
Header Blk Type	0	1	0	0	0	1	0	1	32

7.25 GENERAL TRAILER DESCRIPTION BLOCK (optional)

Bit No.	0	1	2	3	4	5	6	7	
BCD Value MSD	8	4	2	1	8	4	2	1	LSD
Binary Value MSB	128	64	32	16	8	4	2	1	LSB
Block Type	BT_7	BT_6	BT_5	BT_4	BT_3	BT_2	BT_1	BT_0	1
ASCII or binary	AB_7	AB_6	AB_5	AB_4	AB_3	AB_2	AB_1	AB_0	2
	0	0	0	0	0	0	0	0	3
	0	0	0	0	0	0	0	0	4
Block Size	SI_{31}	SI_{30}	SI_{29}	SI_{28}	SI_{27}	SI_{26}	SI_{25}	SI_{24}	5
	SI_{23}	SI_{22}	SI_{21}	SI_{20}	SI_{19}	SI_{18}	SI_{17}	SI_{16}	6
	SI_{15}	SI_{14}	SI_{13}	SI_{12}	SI_{11}	SI_{10}	SI_9	SI_8	7
	SI_7	SI_6	SI_5	SI_4	SI_3	SI_2	SI_1	SI_0	8
Description	DE	DE	DE	DE	DE	DE	DE	DE	9
	DE	DE	DE	DE	DE	DE	DE	DE	10
	DE	DE	DE	DE	DE	DE	DE	DE	11
	DE	DE	DE	DE	DE	DE	DE	DE	12
	DE	DE	DE	DE	DE	DE	DE	DE	13
	DE	DE	DE	DE	DE	DE	DE	DE	14
	DE	DE	DE	DE	DE	DE	DE	DE	15
	DE	DE	DE	DE	DE	DE	DE	DE	16
	DE	DE	DE	DE	DE	DE	DE	DE	17
	DE	DE	DE	DE	DE	DE	DE	DE	18
	DE	DE	DE	DE	DE	DE	DE	DE	19
	DE	DE	DE	DE	DE	DE	DE	DE	20
	DE	DE	DE	DE	DE	DE	DE	DE	21
	DE	DE	DE	DE	DE	DE	DE	DE	22
	DE	DE	DE	DE	DE	DE	DE	DE	23
	DE	DE	DE	DE	DE	DE	DE	DE	24
	0	0	0	0	0	0	0	0	25
	0	0	0	0	0	0	0	0	26
	0	0	0	0	0	0	0	0	27
	0	0	0	0	0	0	0	0	28
	0	0	0	0	0	0	0	0	29
	0	0	0	0	0	0	0	0	30
	0	0	0	0	0	0	0	0	31
Header Blk Type	0	1	1	1	0	0	0	0	32

8.0 HEADER BLOCK PARAMETERS

8.1 GENERAL HEADER, BLOCK #1

All values are in packed BCD unless otherwise specified.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1	F_1, F_2	File number of four digits (0–9999), set to $FFFF_{16}$ when the
2	F_3, F_4	file number is greater than 9999. The expanded file number is contained in Bytes 1, 2, and 3 of General Header, Block #2.
3	Y_1, Y_2	Format code:
4	Y_3, Y_4	8015 20 bit binary 8022 8 bit quaternary 8024 16 bit quaternary 8036 24 bit 2's complement integer 8038 32 bit 2's complement integer 8042 8 bit hexadecimal 8044 16 bit hexadecimal 8048 32 bit hexadecimal 8058 32 bit IEEE 8080 64 bit IEEE 0200 Illegal, do not use 0000 Illegal, do not use
5	K_1, K_2	General constants, 12 digits
6	K_3, K_4	
7	K_5, K_6	
8	K_7, K_8	
9	K_9, K_{10}	
10	K_{11}, K_{12}	
11	YR_1, YR_2	Last two digits of year (0–99)
12	GH	Number of <u>additional</u> Blocks in General Header (unsigned binary). This number will be 2 or greater for Rev 3 (e.g., if only General Header Blocks #1, #2 and #3 are present then GH = 2). For each additional block, the value is increased by one.). If more than 14 are required, set to GH to F_{16}, and use bytes 23–24 of the General Header Block #2.
12	DY_1	
13	DY_2, DY_3	Julian day, 3 digits (1–366).
14	H_1, H_2	Hour of day, 2 digits (0–23) (UTC Time).
15	MI_1, MI_2	Minute of hour, 2 digits (0–59).
16	SE_1, SE_2	Second of minute, 2 digits (0–60). The value 60 is a legal value of this field only during UTC leap seconds.

17	M_1, M_2	Manufacturer's code 2 digits Note: See Appendix A for the current assignments
18	M_3, M_4	Manufacturer's serial number, 4 digits
19	M_5, M_6	
20	0	Not used. Record as zero
21	0	Not used. Record as zero
22	0	Not used. Record as zero
23	$I_3 – I_{-4}$	**Base scan interval.** This is coded as a binary number with the LSB equal to 1/16 msec. This will allow sampling intervals from 1/16 through 8 msec in binary steps. Thus, the allowable base scan intervals are 1/16, 1/8, 1/4, 1/2, 1, 2, 4, and 8 msec. The base scan interval is always the difference between successive timing words. Note: Each channel set is no longer required to have a scan interval that is a multiple of the base scan interval. Instead, this is defined to be the main/dominant scan rate. If a different base scan interval is required, set to FF_{16}, and use byte 25–27 in General Header Block #2.
24	P,	**Polarity.** These 4 binary bits are measured on the sensors, cables, instrument, and source combination and are set into the system manually. The codes are: 0_{16} Untested 1_{16} Zero 2_{16} 45 degrees 3_{16} 90 degrees 4_{16} 135 degrees 5_{16} 180 degrees 6_{16} 225 degrees 7_{16} 270 degrees 8_{16} 315 degrees C_{16} unassigned
24	X	These 4 bits are not used.
25	X	These 8 bits are not used.
26	Z,	Record type 2_{16} Test record 4_{16} Parallel channel test 6_{16} Direct channel test 8_{16} Normal record 1_{16} Other
26	R_1,	
27	R_2, R_3	Record length from time zero (in increments of 0.5 times 1.024 sec). This value can be set from 00.5 to 99.5 representing times from 0.512 sec. to 101.888 sec. A setting of 00.0 indicates the record length is indeterminate. These three nibbles must be set to FFF_{16} when using the Extended Record Length (record length in milliseconds), bytes 15–17, in General Header Block #2.
28	ST/R_1, ST/R_2	Scan types per record. This 2 digit code is the number of scan types per record (1–99). (Zero is invalid.)
29	CS_1, CS_2	Number of channel sets per scan type (1–99). (Zero is invalid; set to FF_{16} when using Extended Channel Sets/Scan Type.) This 2 digit code is the number of

		channel sets per scan. If multiple scan types are used (such as in a switching sampling interval environment), this number is equal to the number of channel sets contained in the scan type with the largest number of channel sets. If scan types also exist with less than this maximum number of channel sets per scan type, dummy channel set descriptors will have to be recorded in the Scan Type Header. This can be done by setting the number of channels in the dummy channel set descriptor to zero (reference Bytes 9 and 10 of the Scan Type Header description). Example 6 illustrates this requirement.
30	SK_1, SK_2	Number of 32 byte fields added to the end of each scan type header in order to record the sample skew of all channels (0–99). (See Appendix E.2 of the SEG-D Standard). Zero indicates that skew is not recorded. If more skew records are required, set to FF_{16}, and use bytes 9 and 10 of General Header Block #2.
31	EC_1, EC_2	Extended Header length. The Extended Header is used to record additional equipment parameters. The two digits (0–99) in this field specify the number of 32 byte extensions. If more than 99 extensions, then these bytes are set to FF_{16}. Bytes 6, 7 and 8 of General Header Block #2 contain the number of 32 byte extensions.
32	EX_1, EX_2	External Header length. The External Header is used to record additional user supplied information in the header. The two digits (0–99) in this field specify the number of 32 byte extensions. If more than 99 extensions, then these bytes are set to FF_{16}. Bytes 28, 29 and 30 of General Header Block #2 contain the number of 32 byte extensions.

8.2 GENERAL HEADER BLOCK #2

INDEX BYTE	ABBREVIATION	DESCRIPTION
1,2,3	$EF_{23} – EF_0$	Extended File Number (three bytes, unsigned binary). For file numbers greater than the 9999, bytes 1 and 2 of the General Header Block #1 must be set to $FFFF_{16}$.
4,5	$EN_{15} – EN_0$	Extended Channel Sets/Scan Type (two bytes, unsigned binary). Allows the number of Channel Sets/Scan Type to be greater than the 99 supported in the standard General Header Block #1 (byte 29). When using the Extended Channel Sets/Scan Type, byte 29 of General Header Block #1 must be set to FF_{16}.
6,7,8	$ECX_{23} – ECX_0$	Extended Header Blocks (three bytes, unsigned binary). Allows the number of Extended Header Blocks (of 32 bytes each) to be greater than the 99 allowed by the standard General Header Block #1 (byte 31). To use more than 99 Extended Header Blocks, set byte 31 of General Header Block #1 to FF_{16}, and use these two bytes. Max size 536,870,912 bytes or 512 MB.
9,10	$ESK_{15} – ESK_0$	Extended Skew Blocks (two bytes, unsigned binary). Number of 32 byte fields added to the end of each scan type header in order to record the sample skew of all channels. (See Appendix E2 of the SEG-D Standard). Zero indicates that

		skew is not recorded. When using the Extended Skew blocks field, byte 30 of General Header Block #1 must be set to FF_{16}.
11	$RMJ_7 – RMJ_0$	Major SEG-D Revision Number (one byte, unsigned binary). Set to 03_{16}.
12	$RMN_7 – RMN_0$	Minor SEG-D Revision Number (one byte, unsigned binary). Set to 00_{16}. Revisions 0 to 0.N are not valid. This version is Rev 3.0.
13–16	$GT_{31} – GT_0$	Number of Blocks of General Trailer (four bytes, unsigned binary). The number of 32 byte blocks to be used for General Trailers. Max size 137,438,953,472 bytes or 128 GB. If not known at the time (i.e. to allow easy appending of trailer blocks while data is stored on disk), it may be set to $FFFFFFFF_{16}$. The reader will then have to read the trailer block by block to determine the shot size. It is strongly recommended to set this field to its correct value when recording to tape.
17–20	$ERL_{31} – ERL_0$	Extended Record Length (four bytes, unsigned binary) indicates the record length in microseconds. When using extended record length, the record length in the General Header Block #1, the Record Length in bytes 26 and 27 must be set to FFF_{16}. Note: This is the maximum record length for any trace in the record. Please refer to the channel set descriptor for actual information of start and end times, number of samples per trace etc. Also note that negative start times are allowed.
21,22	$SN_{15} – SN_0$	Record set number. (two bytes, unsigned binary) This is the sequence number (e.g. the line’s unique sequence number), swath number, or any other applicable number that may be used to identify this group of shot records. Set to 0 if not valid. Allows values of 1–65535.
23,24	$EGH_{15} – EGH_0$	Extended Number of additional Blocks in the General Header (two bytes, unsigned binary). This number will be 2 or greater for Rev 3 (e.g. If only General Header Blocks #1, #2 and #3 are present then EGH = 2). For each additional block, the value is increased by one.) When using this field, the upper nibble of byte 12 of General Header Block #1 must be set to F_{16}.
25,26,27	$BSI_{23} – BSI_0$	Dominant Sampling Interval (3 bytes, unsigned binary). Only used if General Header Block #1 (byte 12) is set to F_{16}. Dominant sampling interval in microseconds (1/1000000 second). As sample rates allowed are very flexible, this is no longer related to the *base scan interval,* with each channel set taking a multiple of it. Instead, it is the main or dominant sampling interval for the record. The sampling interval within each channel set (byte 17–19 of the channel set header) should be used to find the correct sampling interval. This field should not be used to calculate size of records or traces in the record. The field is left for backwards compatibility. It can assume a value from 0 to 16,777,215.
28,29,30	$EH_{23} – EH_0$	External Header Blocks (3 bytes, unsigned binary). Allows the number of 32 byte External Header Blocks to be greater than the 99 allowed by General Header Block #1, byte 32. To use more than 99 External Header Blocks, set byte 32 of General Header Block #1 to FF_{16}, and use these three bytes. Maximum size is 536,870,912 bytes or 512 MB.
31	X	These 8 bits are undefined by the format and may have any value. Note: In future versions these may be defined, so use with caution.

32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For General Header Block #2 this byte is set to 02_{16}.

NOTES:

1. Where the range of allowable numbers is not indicated, the follow ranges apply:
   One byte, unsigned binary, range is 0 – 255
   Two bytes, unsigned binary, range is 0 – 65535
   Three bytes, unsigned binary, range is 0 – 16777215
   Four bytes, unsigned binary, range is 0 – 4294967295

8.3 GENERAL HEADER BLOCK #3 (Timestamp and size header)

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–8	$TZ_{63} – TZ_0$	Time Zero for this record (eight bytes, SEG-D timestamp). The time of sample 0 in all channel sets. Matches timestamp in General Header Block #1 bytes 11–16, but is accurate to 1 microsecond. If the recorder does not know the absolute time, this must be recorded as 0, and all timestamps in the record will be relative to this time.
9–16	$RS_{63} – RS_0$	Record Size (eight bytes, unsigned binary). The total size of the SEG-D record in number of bytes. May be set to 0 if unknown (e.g., the size of the General Trailer is set to $FFFFFF_{16}$ in bytes 13–16 in General Header Block #2). It is strongly recommended to set this field to a proper value unless absolutely necessary. If no General Trailer blocks exist, this field is equal to the Data Size (bytes 17–24). The field is very useful for any byte oriented devices like disks or networks.
17–24	$DS_{63} – DS_0$	Data Size (eight bytes, unsigned binary). The total size of the headers and data in this record in number of bytes. This value enables skipping of all traces to the beginning of the trailer. If no General Trailer exists, this is equal to the Record Size (bytes 9–16). The field is very useful for any byte oriented devices like disks and networks.
25–28	$HS_{31} – HS_0$	Header Size (four bytes, unsigned binary). The total size of the headers (i.e. General Headers, Channel Set Headers, Skew Headers, External Header, Extended Header, etc.) in this record in number of bytes. This value enables skipping of all headers to the beginning of the first trace. The field is very useful for any byte oriented devices like disks or networks.
29	$ERM_7 – ERM_0$	Extended Recording Mode (one byte, unsigned binary). Set to 1 if this record is using the Extended Recording Mode, 0 for normal record.
30	$RTM_7 – RTM_0$	Relative Time Mode (one byte, unsigned binary). Set to 1 if this record does not contain absolute timestamps, 0 for normal record. If set to 1, bytes 1 to 8 must be set to 0, and all timestamps in the record must be relative to this time (i.e. start of record).
31	X	These 8 bits are undefined by the format and may have any value. Note: In future versions these may be defined, so use with caution.

32	$HT_7 - HT_0$	Header block type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For General Header Block #3 this byte is set to 03_{16}.

8.4 GENERAL HEADER BLOCK #4 (Vessel/Crew Identification) (optional)

This block contains the name/description of the vessel or crew.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1,2,3	VCA	Abbreviated vessel or crew name (3 characters, ASCII). May be used on tape labels, etc. Left justified, padded with space (20_{16}) characters. May be blank if no abbreviation exists.
4–31	VC	Vessel or crew name (ASCII text). 28 character vessel or crew name which uniquely identifies the vessel/crew within the enterprise. Left justified, padded with space (20_{16}) characters.
32	$HT_7 - HT_0$	Header block type (1 byte, unsigned binary). Set to 10_{16} for Vessel/Crew Identification.

8.5 GENERAL HEADER BLOCK #5 (Survey Area Name) (optional)

This block contains the name/description of the survey area.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–31	SAN	Survey Area Name (ASCII text). 31 character survey area name which properly identifies the area the record was recorded in. Left justified, padded with space (20_{16}) characters.
32	$HT_7 - HT_0$	Header block type (1 byte, unsigned binary). Set to 11_{16} for Survey Area Name.

8.6 GENERAL HEADER BLOCK #6 (Client Identification) (optional)

This block contains the name/description of the client.

INDEX BYTE	ABBREVIATION	DESCRIPTION

1–31	CI	Client Identification (ASCII text). 31 character client name/identification which properly identifies the client for this survey. Left justified, padded with space (20_{16}) characters.
32	$HT_7 – HT_0$	Header block type (1 byte, unsigned binary). Set to 12_{16} for Client Identification.

8.7 GENERAL HEADER BLOCK #7 (Job identification) (optional)

This block contains the name/description of the job/survey.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–5	JIA	Abbreviated Job Identification (5 characters ASCII). May be used on tape labels etc. Left justified, padded with space (20_{16}) characters. May be blank if no abbreviation exists.
6–31	JI	Job Identification (ASCII text). 26 character job name/number/identification which properly identifies the survey within the enterprise. Left justified, padded with space (20_{16}) characters.
32	$HT_7 – HT_0$	Header block type (1 byte, unsigned binary). Set to 13_{16} for Job Identification.

8.8 GENERAL HEADER BLOCK #8 (Line identification) (optional)

This block contains the description of current line. Mainly targeted to Marine operations, however any recorder might use this field for line/swath related information. Make sure the information here matches the sequence number in General Header Block #2, bytes 21–22.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–7	LA	Line Abbreviation (ASCII text). 7 character line abbreviation (e.g. line number, though letters are allowed) which uniquely identifies the line within the survey. This is typically the pre-plot line number without information about sequence-number, line type (primary/in-fill/reshoot, etc.) Left justified, padded with space (20_{16}) characters.
8–31	LI	Line Identification (ASCII text). 24 character line name/identification which uniquely identifies the line within the survey/job. Left justified, padded with space (20_{16}) characters.
32	$HT_7 – HT_0$	Header block type (1 byte, unsigned binary). Set to 14_{16} for Line Identification.

8.9 SOURCE DESCRIPTION BLOCK (optional)

Source description blocks may be inserted into the General Header or relevant Trace Headers (e.g. source auxiliary traces), depending on what is most useful. It is also allowed to replicate Source Description Blocks in the General Header and Trace Headers if that may improve decoding and use of the data.

If multiple source events exist inside the same record, it is required to insert multiple Source Description Blocks describing each of the source events. If the source event does not occur at time zero for the SEG-D record, an Additional Source Info block must be inserted after each source block and the time of the event must be properly stored in bytes 1–8.

If source auxiliary traces are stored in the record, the source description block may be followed by Source Auxiliary Channel Reference blocks describing which traces contain more information regarding the sources.

See Appendix E.8 for examples of how to encode source information.

8.9.1 VIBRATOR

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$EF_{23} – EF_0$	Expanded File Number (three bytes, unsigned binary). For file numbers greater than 9999, bytes 1 and 2 of the General Header Block #1 must be set to $FFFF_{16}$.
4–6	$SLN_{23} – SLN_0$	Source Line Number, Integer (three bytes, two's complement, signed binary). General Header Block #2 contains the source location for one Source Set. Additional General Header blocks may be used to provide position information for additional source sets.
7,8	$SLN_{-1} – SLN_{-16}$	Source Line Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 4–8 is always equal to 65536 times the full Source Line Number.
9–11	$SPN_{23} – SPN_0$	Source Point Number, Integer (three bytes, two's complement, signed binary).
12,13	$SPN_{-1} – SPN_{-16}$	Source Point Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 9–13 is always equal to 65536 times the full Source Point Number.
14	$SPI_7 – SPI_0$	Source Point Index (one byte, unsigned binary). This index allows several locations for the source in the grid, the original value is one and that value is incremented by one every time the source is moved, even when it is moved back to a previous location. A zero value means that the Source Point Index is not recorded.
15	$PC_7 – PC_0$	Phase Control (unsigned binary). Identifies the signal used to control the phase of the vibrator output. Assumes following the 1991 Vibrator Polarity Standards: 00_{16} Phase Control not recorded 01_{16} Baseplate accelerometer

		02_{16} Reaction Mass 03_{16} Weighted sum (baseplate acceleration times mass plus reaction mass acceleration times its mass) 04_{16} Direct force measurement It is anticipated that additional codes will be added later. If Phase Control is set to Zero then the Phase Angle (Bytes 17, 18) is undefined.
16	$V_7 - V_0$	Type Vibrator (unsigned binary). 00_{16} Type not recorded 01_{16} P wave vibrator 02_{16} Shear wave vibrator 03_{16} Marine vibrator Other types may be added later.
17,18	$PA_{15} - PA_0$	Phase Angle (two bytes, two's complement, signed binary). The Phase angle of the intercept of the pilot signal with respect to the phase feedback signal, measured in degrees. Phase Angle is set to zero when Phase Control (Byte 15) is zero (Phase Control not recorded).
19	$SI_7 - SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey. 0 means unknown (same as source set).
20	$SS_7 - SS_0$	Source Set Number (one byte, unsigned binary). Used to allow multiple sets of sources. Zero means this source id is not part of a source set (may be a source set by itself).
21	$RI_7 - RI_0$	Re-shoot Index (one byte, unsigned binary). The re-shoot index is used to indicate that this shot has been recorded earlier, and this record should replace it. Re-shoot index starts at 0 (indicates recorded for first time).
22	$GI_7 - GI_0$	Group Index (one byte, unsigned binary). Group index is used to indicate this record is part of a set of records needed to be processed as a unit. This may be a stack fold number/vibrator sweep number in case of recording records that must be stacked prior to processing, or phase index in case of HFVS type of acquisition, where a group of records must be processed to do source separation prior to data processing. 0 is not allowed.
23	$DI_7 - DI_0$	Depth Index (one byte, unsigned binary). The depth index is used to indicate sources fired at different depths on the same location. 0 is not allowed.
24,25	$OX_{15} - OX_0$	Offset cross-line (two bytes, two's complement, signed binary). Cross-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
26,27	$OI_{15} - OI_0$	Offset in-line (two bytes, two's complement, signed binary). In-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.

28,29	$SZ_{15} – SZ_0$	Size (two bytes, unsigned binary). The size of the source. For vibrator this is the total peak force generated by the vibrators associated with this source. Unit 100 pound force, i.e. if 4 vibrators of a peak force of 70000lbs is combined in this source, the value of this field is set to 2800.
30,31	$OD_{15} – OD_0$	Offset depth (two bytes, two's complement, signed binary). Offset depth offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Positive direction upwards. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Vibrator source this byte is set to 15_{16}.

8.9.2 EXPLOSIVE

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$EF_{23} – EF_0$	Expanded File Number (three bytes, unsigned binary). For file numbers greater than 9999, bytes 1 and 2 of the General Header Block #1 must be set to $FFFF_{16}$.
4–6	$SLN_{23} – SLN_0$	Source Line Number, Integer (three bytes, two's complement, signed binary). General Header Block #2 contains the source location for one Source Set. Additional General Header blocks may be used to provide position information for additional source sets.
7,8	$SLN_{-1} – SLN_{-16}$	Source Line Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 4–8 is always equal to 65536 times the full Source Line Number.
9–11	$SPN_{23} – SPN_0$	Source Point Number, Integer (three bytes, two's complement, signed binary).
12,13	$SPN_{-1} – SPN_{-16}$	Source Point Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 9–13 is always equal to 65536 times the full Source Point Number.
14	$SPI_7 – SPI_0$	Source Point Index (one byte, unsigned binary). This index allows several locations for the source in the grid, the original value is one and that value is incremented by one every time the source is moved, even when it is moved back to a previous location. Zero value means that the Source Point Index is not recorded.
15,16	$DE_{15} – DE_0$	Depth (two bytes, unsigned binary). Nominal shot depth in centimeters (1/100 meter).
17	$CL_7 – CL_0$	Charge length (one byte, unsigned binary). Length of charge in decimeters (1/10 meter).
18	$ST_7 – ST_0$	Soil Type (one byte, unsigned binary). This is the nominal type of soil or near surface medium. The following types are defined 00_{16} Water

01_{16}	Alluvium
02_{16}	Dry sand
03_{16}	Weathering
04_{16}	Mud
05_{16}	Glacial
06_{16}	Shale
07_{16}	Sand
08_{16}	Sandstone
09_{16}	Limestone
$0A_{16}$	Granite
$0B_{16}$	Chalk
$0C_{16}$	Gypsum
$0D_{16}$	Salt
$0E_{16}$	Gabbro
FF_{16}	Other

Other types may be added later.

19	$SI_7 – SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey. 0 means unknown (same as source set).
20	$SS_7 – SS_0$	Source Set Number (one byte, unsigned binary). Used to allow multiple sets of sources. Zero means this source id is not part of a source set (may be a source set by itself).
21	$RI_7 – RI_0$	Re-shoot Index (one byte, unsigned binary). The re-shoot index is used to indicate that this shot has been recorded earlier, and this record should replace it. Re-shoot index starts at 0 (indicates recorded for first time).
22	$GI_7 – GI_0$	Group Index (one byte, unsigned binary). Group index is used to indicate this record is part of a set of records needed to be processed as a unit. This may be a stack fold number/vibrator sweep number in case of recording records that must be stacked prior to processing, or phase index in case of HFVS type of acquisition, where a group of records must be processed to do source separation prior to data processing. 0 is not allowed.
23	$DI_7 – DI_0$	Depth Index (one byte, unsigned binary). The depth index is used to indicate sources fired at different depths on the same location. 0 is not allowed.
24,25	$OX_{15} – OX_0$	Offset cross-line (two bytes, two's complement, signed binary). Cross-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
26,27	$OI_{15} – OI_0$	Offset in-line (two bytes, two's complement, signed binary). In-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
28,29	$SZ_{15} – SZ_0$	Size (two bytes, two's complement, signed binary). The size of source. For explosive this is the size of charge in grams.
30,31	$OD_{15} – OD_0$	Offset depth (two bytes, two's complement, signed binary). Offset depth offset is number of centimeters (1/100 meter) the source is located from the position

		indicated by the line/point index. Positive direction upwards. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Explosive source this byte is set to 16_{16}.

8.9.3 AIRGUN

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$EF_{23} – EF_0$	Expanded File Number (three bytes, unsigned binary). For file numbers greater than 9999, bytes 1 and 2 of the General Header Block #1 must be set to $FFFF_{16}$.
4–6	$SLN_{23} – SLN_0$	Source Line Number, Integer (three bytes, two's complement, signed binary). General Header Block #2 contains the source location for one Source Set. Additional General Header blocks may be used to provide position information for additional source sets.
7,8	$SLN_{-1} – SLN_{-16}$	Source Line Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 4–8 is always equal to 65536 times the full Source Line Number.
9–11	$SPN_{23} – SPN_0$	Source Point Number, Integer (three bytes, two's complement, signed binary).
12,13	$SPN_{-1} – SPN_{-16}$	Source Point Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 9–13 is always equal to 65536 times the full Source Point Number.
14	$SPI_7 – SPI_0$	Source Point Index (one byte, unsigned binary). This index allows several locations for the source in the grid, the original value is one and that value is incremented by one every time the source is moved, even when it is moved back to a previous location. Zero value means that the Source Point Index is not recorded.
15,16	$DE_{15} – DE_0$	Depth (two bytes, unsigned binary). This is the nominal depth from the water surface in centimeters (1/100 meter) for this source. Allows values of 0–655.35 meters
17,18	$AP_{15} – AP_0$	Air Pressure (two bytes, unsigned binary). The nominal operating air pressure Unit Pascal/Bar/PSI? SI unit Pascal preferred.
19	$SI_7 – SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey. 0 means unknown (same as source set).
20	$SS_7 – SS_0$	Source Set Number (one byte, unsigned binary). Used to allow multiple sets of sources. Zero means this source id is not part of a source set (may be a source set by itself).

21	$RI_7 – RI_0$	Re-shoot Index (one byte, unsigned binary). The re-shoot index is used to indicate that this shot has been recorded earlier, and this record should replace it. Re-shoot index starts at 0 (indicates recorded for first time).
22	$GI_7 – GI_0$	Group Index (one byte, unsigned binary). Group index is used to indicate this record is part of a set of records needed to be processed as a unit. This may be a stack fold number/vibrator sweep number in case of recording records that must be stacked prior to processing, or phase index in case of HFVS type of acquisition, where a group of records must be processed to do source separation prior to data processing. 0 is not allowed.
23	$DI_7 – DI_0$	Depth Index (one byte, unsigned binary). The depth index is used to indicate sources fired at different depths on the same location. 0 is not allowed.
24,25	$OX_{15} – OX_0$	Offset cross-line (two bytes, two's complement, signed binary). Cross-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
26,27	$OI_{15} – OI_0$	Offset in-line (two bytes, two's complement, signed binary). In-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
28,29	$SZ_{15} – SZ_0$	Size (two bytes, two's complement, signed binary). The size of source. For Airgun this is the combined volume of the guns associated with this source ID. Unit cubic inches.
30,31	$OD_{15} – OD_0$	Offset depth (two bytes, two's complement, signed binary). Offset depth offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Positive direction upwards. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Airgun source this byte is set to 17_{16}.

8.9.4 WATERGUN

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$EF_{23} – EF_0$	Expanded File Number (three bytes, unsigned binary). For file numbers greater than 9999, bytes 1 and 2 of the General Header Block #1 must be set to $FFFF_{16}$.
4–6	$SLN_{23} – SLN_0$	Source Line Number, Integer (three bytes, two's complement, signed binary). General Header Block #2 contains the source location for one Source Set. Additional General Header blocks may be used to provide position information for additional source sets.

7,8	$SLN_{-1} – SLN_{-16}$	Source Line Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 4–8 is always equal to 65536 times the full Source Line Number.
9–11	$SPN_{23} – SPN_0$	Source Point Number, Integer (three bytes, two's complement, signed binary).
12,13	$SPN_{-1} – SPN_{-16}$	Source Point Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 9–13 is always equal to 65536 times the full Source Point Number.
14	$SPI_7 – SPI_0$	Source Point Index (one byte, unsigned binary). This index allows several locations for the source in the grid, the original value is one and that value is incremented by one every time the source is moved, even when it is moved back to a previous location. Zero value means that the Source Point Index is not recorded.
15,16	$DE_{15} – DE_0$	Depth (two bytes, unsigned binary). This is the nominal depth from the water surface in centimeters (1/100 meter) for this source.
17,18	$AP_{15} – AP_0$	Air Pressure (two bytes, unsigned binary). The nominal operating air pressure Unit Pascal/Bar/PSI? SI unit preferred.
19	$SI_7 – SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey. 0 means unknown (same as source set).
20	$SS_7 – SS_0$	Source Set Number (one byte, unsigned binary). Used to allow multiple sets of sources. Zero means this source id is not part of a source set (may be a source set by itself).
21	$RI_7 – RI_0$	Re-shoot Index (one byte, unsigned binary). The re-shoot index is used to indicate that this shot has been recorded earlier, and this record should replace it. Re-shoot index starts at 0 (indicates recorded for first time).
22	$GI_7 – GI_0$	Group Index (one byte, unsigned binary). Group index is used to indicate this record is part of a set of records needed to be processed as a unit. This may be a stack fold number/vibrator sweep number in case of recording records that must be stacked prior to processing, or phase index in case of HFVS type of acquisition, where a group of records must be processed to do source separation prior to data processing. 0 is not allowed.
23	$DI_7 – DI_0$	Depth Index (one byte, unsigned binary). The depth index is used to indicate sources fired at different depths on the same location. 0 is not allowed.
24,25	$OX_{15} – OX_0$	Offset cross-line (two bytes, two's complement, signed binary). Cross-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
26,27	$OI_{15} – OI_0$	Offset in-line (two bytes, two's complement, signed binary). In-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.

28,29	$SZ_{15} – SZ_0$	Size (two bytes, two's complement, signed binary). The size of source. For Watergun this is the total volume of guns associated with this source ID. Unit cubic inches.
30,31	$OD_{15} – OD_0$	Offset depth (two bytes, two's complement, signed binary). Offset depth offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Positive direction upwards. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Watergun source this byte is set to 18_{16}.

8.9.5 ELECTROMAGNETIC

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$EF_{23} – EF_0$	Expanded File Number (three bytes, unsigned binary). For file numbers greater than 9999, bytes 1 and 2 of the General Header Block #1 must be set to $FFFF_{16}$.
4–6	$SLN_{23} – SLN_0$	Source Line Number, Integer (three bytes, two's complement, signed binary). General Header Block #2 contains the source location for one Source Set. Additional General Header blocks may be used to provide position information for additional source sets.
7,8	$SLN_{-1} – SLN_{-16}$	Source Line Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 4–8 is always equal to 65536 times the full Source Line Number.
9–11	$SPN_{23} – SPN_0$	Source Point Number, Integer (three bytes, two's complement, signed binary).
12,13	$SPN_{-1} – SPN_{-16}$	Source Point Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 9–13 is always equal to 65536 times the full Source Point Number.
14	$SPI_7 – SPI_0$	Source Point Index (one byte, unsigned binary). This index allows several locations for the source in the grid, the original value is one and that value is incremented by one every time the source is moved, even when it is moved back to a previous location. Zero value means that the Source Point Index is not recorded.
15	$ST_7 – ST_0$	EM Source Type (one byte, unsigned binary). One of the following source types: 00_{16} Unknown 01_{16} Dipole 02_{16} Loop Coil 03_{16} Airborne 04_{16} Open Loop

		05_{16} Closed loop $0F_{16}$ Other
16–18	$MO_{23} – MO_0$	Source Moment (three bytes, unsigned binary). The source moment in AmpereMeter (for Dipole source) or AmperMeter^2 (for Coil source).
19	$SI_7 – SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey. 0 means unknown (same as source set).
20	$SS_7 – SS_0$	Source Set Number (one byte, unsigned binary). Used to allow multiple sets of sources. Zero means this source id is not part of a source set (may be a source set by itself).
21	$RI_7 – RI_0$	Re-shoot Index (one byte, unsigned binary). The re-shoot index is used to indicate that this shot has been recorded earlier, and this record should replace it. Re-shoot index starts at 0 (indicates recorded for first time).
22	$GI_7 – GI_0$	Group Index (one byte, unsigned binary). Group index is used to indicate this record is part of a set of records needed to be processed as a unit. This may be a stack fold number/vibrator sweep number in case of recording records that must be stacked prior to processing, or phase index in case of HFVS type of acquisition, where a group of records must be processed to do source separation prior to data processing. 0 is not allowed.
23	$DI_7 – DI_0$	Depth Index (one byte, unsigned binary). The depth index is used to indicate sources fired at different depths on the same location. 0 is not allowed.
24,25	$OX_{15} – OX_0$	Offset cross-line (two bytes, two’s complement, signed binary). Cross-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
26,27	$OI_{15} – OI_0$	Offset in-line (two bytes, two’s complement, signed binary). In-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
28,29	$CV_{15} – CV_0$	Source Current or Voltage (two bytes, unsigned binary). The source current (in Ampere) or voltage (in Volts), depending on the source type.
30,31	$OD_{15} – OD_0$	Offset depth (two bytes, two’s complement, signed binary). Offset depth offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Positive direction upwards. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Electromagnetic source this byte is set to 19_{16}.

8.9.6 OTHER SOURCE

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$EF_{23} – EF_0$	Expanded File Number (three bytes, unsigned binary). For file numbers greater than 9999, bytes 1 and 2 of the General Header Block #1 must be set to $FFFF_{16}$.
4,5,6	$SLN_{23} – SLN_0$	Source Line Number, Integer (three bytes, two's complement, signed binary). General Header Block #2 contains the source location for one Source Set. Additional General Header Blocks may be used to provide position information for additional source sets.
7,8	$SLN_{-1} – SLN_{-16}$	Source Line Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 4–8 is always equal to 65536 times the full Source Line Number.
9–11	$SPN_{23} – SPN_0$	Source Point Number, Integer (three bytes, two's complement, signed binary).
12,13	$SPN_{-1} – SPN_{-16}$	Source Point Number, Fraction (two bytes, fixed binary point). The two's complement integer formed by Bytes 9–13 is always equal to 65536 times the full Source Point Number.
14	$SPI_7 – SPI_0$	Source Point Index (one byte, unsigned binary). This index allows several locations for the source in the grid, the original value is one and that value is incremented by one every time the source is moved, even when it is moved back to a previous location. Zero value means that the Source Point Index is not recorded.
15,16,17,18	X	Unused, may have any value.
19	$SI_7 – SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey. 0 means unknown (same as source set).
20	$SS_7 – SS_0$	Source Set Number (one byte, unsigned binary). Used to allow multiple sets of sources. Zero means this source id is not part of a source set (may be a source set by itself).
21	$RI_7 – RI_0$	Re-shoot Index (one byte, unsigned binary). The re-shoot index is used to indicate that this shot has been recorded earlier, and this record should replace it. Re-shoot index starts at 0 (indicates recorded for first time).
22	$GI_7 – GI_0$	Group Index (one byte, unsigned binary). Group index is used to indicate this record is part of a set of records needed to be processed as a unit. This may be a stack fold number/vibrator sweep number in case of recording records that must be stacked prior to processing, or phase index in case of HFVS type of acquisition, where a group of records must be processed to do source separation prior to data processing. 0 is not allowed.
23	$DI_7 – DI_0$	Depth Index (one byte, unsigned binary). The depth index is used to indicate sources fired at different depths on the same location. 0 is not allowed.

24,25	$OX_{15} – OX_0$	Offset cross-line (two bytes, two's complement, signed binary). Cross-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offset to it. The offset may be nominal or actual.
26,27	$OI_{15} – OI_0$	Offset in-line (two bytes, two's complement, signed binary). In-line offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Used to organize groups of sources firing at the same logical source point, but with different offset to it. The offset may be nominal or actual.
28,29	X	Unused space, may have any value.
30,31	$OD_{15} – OD_0$	Offset depth (two bytes, two's complement, signed binary). Offset depth offset is number of centimeters (1/100 meter) the source is located from the position indicated by the line/point index. Positive direction upwards. Used to organize groups of sources firing at the same logical source point, but with different offsets. The offset may be nominal or actual.
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Other Source this byte is set to $1F_{16}$.

8.10 ADDITIONAL SOURCE INFO (optional)

Follows the Source Description Block

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–8	$T_{63} – T_0$	Time (eight bytes, SEG-D timestamp). A time of 0 is allowed, and will indicate source fired at start of record (time as indicated in General Header Block #3, bytes 1 to 8).
9 .	$SST_7 – SST_0$	Source Status (1 byte, unsigned binary) 00_{16} Untested/unknown status 01_{16} Source fired OK 12_{16} Timing error 13_{16} Mechanical failure 14_{16} Software error 15_{16} Hydraulics failure 16_{16} Air leakage 17_{16} Power failure 18_{16} Communication failure FF_{16} Unknown error
10	$SI_7 – SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey. 0 means unknown (same as source set). Used to determine which source block this block belongs to.
11	$MV_7 – MV_0$	Source Moving (one byte, unsigned binary). Source moving or stationary flag. Set to 1 if source is moving during the record, 0 if stationary.

12–31	ED	Error Description (20 bytes, ASCII text). Error description in free text as defined by the recorder. The text should be a human readable string. "OK" should be used to indicate the source fired OK, however if another string would better represent the status, the recorder is free to do so. Field is left justified, and unused characters should be filled with spaces (20_{16}).
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Additional Source Info this byte is set to $1A_{16}$.

8.11 SOURCE AUXILIARY CHANNEL REFERENCE (optional)

Follows the Source Description Block

This block is inserted into the General Header (or Trace Header) after the Source Description Block to indicate which auxiliary channels contain information data regarding the source. Multiple Source Auxiliary Channel Reference blocks may be used for each source if necessary. May be inserted into (auxiliary source) trace headers to indicate which (other) traces are related to the source.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1	$SI_7 – SI_0$	Source Id (one byte, unsigned binary). The source id is a unique number assigned to each source in the survey.
2	$ST1_1$, $ST1_2$	Channel 1, Scan Type Number (one byte, two digit, BCD). Currently only values 1–99 allowed.
3,4	$CS1_{15} – CS1_0$	Channel 1, Channel Set Number (two bytes, unsigned binary).
5–7	$TN1_{23} – TN1_0$	Channel 1, Trace Number (3 bytes, unsigned binary).
8	$ST2_1$, $ST2_2$	Channel 2, Scan Type Number (one byte, two digit, BCD). Currently only values 1–99 allowed.
9,10	$CS2_{15} – CS2_0$	Channel 2, Channel Set Number (two bytes, unsigned binary).
11–13	$TN2_{23} – TN2_0$	Channel 2, Trace Number (three bytes, unsigned binary).
14	$ST3_1$, $ST3_2$	Channel 3, Scan Type Number (one byte, two digit, BCD). Currently only values 1–99 allowed.
15,16	$CS3_{15} – CS3_0$	Channel 3, Channel Set Number (two bytes, unsigned binary).
17–19	$TN3_{23} – TN3_0$	Channel 3, Trace Number (three byte, unsigned binary).
20	$ST4_1$, $ST4_2$	Channel 4, Scan Type Number (one byte, two digit, BCD). Currently only values 1–99 allowed.
21, 22	$CS4_{15} – CS4_0$	Channel 4, Channel Set Number (two bytes, unsigned binary).
23–25	$TN4_{23} – TN4_0$	Channel 4, Trace Number. (three byte, unsigned binary).

26	$ST5_1$, $ST5_2$	Channel 5, Scan Type Number (one byte, two digit, BCD). Currently only values 1–99 allowed.
27, 28	$CS5_{15} – CS5_0$	Channel 5, Channel Set Number (two bytes, unsigned binary).
29–31	$TN5_{23} – TN5_0$	Channel 5, Trace Number (three bytes, unsigned binary).
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Source Auxiliary Channel Reference this byte is set to 21_{16}.

8.12 COORDINATE REFERENCE SYSTEM IDENTIFICATION (conditional)

This header block is mandatory if a position block is included in the record.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–31	CRID	CRS ID (31 bytes, ASCII text). The coordinate reference system (CRS) identification in a textual format as defined in Appendix D. The stanza is split into 31 byte blocks, and stored in multiple, consecutive CRS identification blocks as part of the General Header. Text is left justified, the block is padded with space (20_{16}) characters if needed.
32	$HT_7 – HT_0$	Header type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. For Coordinate Reference System Identification this byte is set to 55_{16}.

8.13 POSITION BLOCKS (optional)

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–8	$TP_{63} – TP_0$	Time of position (eight bytes, SEG-D timestamp). Time to which this position applies, i.e., sample time.
9–16	$TC_{63} – TC_0$	Time of measurement/calculation (eight bytes, SEG-D timestamp). Time the position was measured or calculated.
17–20	$VE_{31} – VE_0$	Vertical error quality estimate: 95% precision estimate in same units as vertical coordinate (four bytes, IEEE float). Set to 0 if coordinate reference type only uses the horizontal plane. Set to infinity (exponent all bits set, mantissa all zeroes) if not available.

21–24	$HEA_{31} – HEA_0$	Horizontal error quality estimate: 95% error ellipse semi-major axis in same units as horizontal coordinate (four bytes, IEEE float). Set to 0 if coordinate reference type only uses the vertical plane. Set to infinity (exponent all bits set, mantissa all zeroes) if not available. HEA, HEB and HEO must all be valid, or all be infinity.
25–28	$HEB_{31} – HEB_0$	Horizontal error quality estimate: 95% error ellipse semi-minor axis in same units as horizontal coordinate (four bytes, IEEE float). Set to 0 if coordinate reference type only uses the vertical plane. Set to infinity (exponent all bits set, mantissa all zeroes) if not available. HEA, HEB and HEO must all be valid, or all be infinity.
29–30	$HEO_{15} – HEO_0$	Horizontal error quality estimate: Bearing of error ellipse semi-major axis (two bytes, unsigned binary). Orientation to map grid north in steps of 1/100 of a degree. Set to 0 if coordinate reference type only uses the vertical plane. Set $FFFF_{16}$ if not available. HEA, HEB and HEO must all be valid, or all be infinity.
31	$PT_7 – PT_0$	Position type (one byte, unsigned binary). The type of position, the following types are supported: 01_{16} Planned/preplot 02_{16} Measured 03_{16} Processed 04_{16} Final $0F_{16}$ Unknown
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 50_{16} for Position Block 1.

INDEX BYTE	ABBREVIATION	DESCRIPTION
33–40	$T1C1_{63} – T1C1_0$	First coordinate for coordinate tuple 1 (eight bytes, IEEE double float). This coordinate and its unit is as given in the CRS definition in the location data stanza identified through ID1. Set to infinity (exponent all bits set, mantissa all zeroes) if CRS type is one-dimensional i.e. vertical.
41–48	$T1C2_{63} – T1C2_0$	Second coordinate for coordinate tuple 1 (eight bytes, IEEE double float). This coordinate and its unit is as given in the CRS definition in the location data stanza identified through ID1. Set to infinity (exponent all bits set, mantissa all zeroes) if CRS type is one-dimensional i.e. vertical.
49–56	$T1C3_{63} – T1C3_0$	Third coordinate for coordinate tuple 1 (eight bytes, IEEE double float). This coordinate and its unit is as given in the CRS definition in the location data stanza identified through ID1. Set to infinity (exponent all bits set, mantissa all zeroes) if CRS type is two-dimensional (projected or geographic 2D).
57–58	$ID1_{15} – ID1_0$	Location Data Stanza ID 1 (two bytes, unsigned binary). Location Data Stanza ID of the CRS for coordinate tuple 1. Defined in the Coordinate Reference

		System Identification blocks of the record header. Please refer to Appendix D for a description of the Coordinate Reference System Identification format. Range 1 to 65535.
59	$PV1_7 – PV1_0$	Position 1 Valid (one byte, unsigned binary). Position 1 valid flag. Set to 1 if position coordinates are valid, 0 if not.
60	$PQ1_7 – PQ1_0$	Position 1 Quality (one byte, unsigned binary). Position 1 quality flag. Supported values: 01_{16} Position is good 02_{16} Quality uncertain 03_{16} Position is bad, not to be used This field must always be filled in. The field will be an interpretation of VE, HEA, HEB and HEO (bytes 17–30) if available.
61–63	X	Undefined
64	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 51_{16} for Position Block 2.

INDEX BYTE	ABBREVIATION	DESCRIPTION
65–72	$T2C1_{63} – T2C1_0$	First coordinate for coordinate tuple 2 (eight bytes, IEEE double float). This coordinate and its unit is as given in the CRS definition in the location data stanza identified through ID2. Set to infinity (exponent all bits set, mantissa all zeroes) if CRS type is one-dimensional i.e. vertical.
73–80	$T2C2_{63} – T2C2_0$	Second coordinate for coordinate tuple 2 (eight bytes, IEEE double float). This coordinate and its unit is as given in the CRS definition in the location data stanza identified through ID2. Set to infinity (exponent all bits set, mantissa all zeroes) if CRS type is one-dimensional i.e. vertical.
81–88	$T2C3_{63} – T2C3_0$	Third coordinate for coordinate tuple 2 (eight bytes, IEEE double float). This coordinate and its unit is as given in the CRS definition in the location data stanza identified through ID2. Set to infinity (exponent all bits set, mantissa all zeroes) if CRS type is two-dimensional (projected or geographic 2D).
89–90	$ID2_{15} – ID2_0$	Location Data Stanza ID 2 (two bytes, unsigned binary). Location Data Stanza ID of the CRS for coordinate tuple 2. Defined in the Coordinate Reference System Identification blocks of the record header. Please refer to Appendix D for a description of the Coordinate Reference System Identification format. Range 1 to 65535.
91	$PV2_7 – PV2_0$	Position 2 Valid (one byte, unsigned binary). Position 2 valid flag. Set to 1 if position coordinates are valid, 0 if not.

92	$PQ2_7 – PQ2_0$	Position 2 Quality (one byte, unsigned binary). Position 2 quality flag. Supported values: 00_{16} Position is bad, not to be used 01_{16} Quality uncertain 02_{16} Position is good This field must always be filled in. The field will be an interpretation of VE, HEA, HEB and HEO (bytes 17–30) if available.
93–95	X	Undefined
96	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 52_{16} for Position Block 3.

8.14 RELATIVE POSITION BLOCK (optional)

The header block describes a position relative to the position described by the 96 byte position header. A textual description is included.

Relative offset is in projection/grid coordinates only. Follows directly after the position header to which it is relative. Multiple relative positions might be connected to the same position header, e.g. to describe the front and tail of source (Position Header describes center of source).

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–4	$OE_{31} – OE_0$	Offset Easting (four bytes, IEEE float). Offset in projected (grid) coordinates in easting direction, in units given in projected CRS definition. Note that (i) although given first here, easting may not be the first coordinate given in Position Header block, (ii) SEG-D only allows horizontal offsets to be given in projected terms.
5–8	$ON_{31} – ON_0$	Offset Northing (four bytes, IEEE float). Offset in projected (grid) coordinates in northing direction, in units given in projected CRS definition. Note that (i) although given second here, northing may not be the second coordinate given in Position Header block, (ii) SEG-D only allows horizontal offsets to be given in projected terms.
9–12	$OD_{31} – OD_0$	Offset Vertical (four bytes, IEEE float). Offset in vertical direction, in units given in vertical CRS definition. Note that whether a positive offset is a height or a depth is defined in the CRS Header.
13–31	DE	Description (19 bytes, ASCII text). Manufacturer defined, textual description of the equipment
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 56_{16} for Relative Position Block.

8.15 SCAN TYPE HEADER (channel set descriptor)

The scan type header is determined by the system configuration and consists of one or more channel set descriptors each of 96 bytes followed by a series of 32 byte sample skew fields. A channel set is defined as a group of channels operating with the same set of parameters and being sampled as part of a scan of data. A scan type header can be composed of from 1 to 99 channel set descriptors. If dynamic parameter changes are required during the recording, additional scan type headers must be added, each containing the channel set descriptors necessary to define the new parameters. Each scan type header must have the same number of channel set descriptors (see Appendix E.4).

8.16 CHANNEL SET DESCRIPTOR

INDEX BYTE	ABBREVIATION	DESCRIPTION
1	ST_1, ST_2	These two digits (1–99) identify the number of the scan type header to be described by the subsequent bytes. The first scan type header is 1 and the last scan type header number is the same value as Byte 28 (ST/R) of the General Header Block #1. If a scan type header contains more than one channel set descriptor, the scan type header number will be repeated in each of its channel set descriptors. If the system does not have dynamic parameter changes during the record, such as switched sampling intervals, there will only be one scan type header required.
2,3	$CS_{15} – CS_0$	Channel Set Number (two bytes, unsigned binary). These digits (1–65535) identify the channel set to be described in the next 94 bytes within this scan type header. The first channel set is "1" and the last channel set number is the same number as Byte 29 (CS) of the General Header Block #1. If the scan actually contains fewer channel sets than CS, then dummy channel set descriptors are included as specified in Byte 29 of General Header Block #1.
4	$C_7 – C_0$	Channel type Identification (one byte, unsigned binary).

00_{16}	Unused
10_{16}	Seis
11_{16}	Electromagnetic (EM)
20_{16}	Time break
21_{16}	Clock timebreak
22_{16}	Field timebreak
30_{16}	Up hole
40_{16}	Water break
50_{16}	Time counter
60_{16}	External Data
61_{16}	Acoustic range measurement
62_{16}	Acoustic reference measured (correlation reference)
63_{16}	Acoustic reference nominal (correlation reference)
70_{16}	Other
80_{16}	Signature/unfiltered
90_{16}	Signature/filtered
91_{16}	Source signature/unfiltered
92_{16}	Source signature/filtered

93_{16}	Source signature/estimated
94_{16}	Source signature/measured
95_{16}	Source base plate
96_{16}	Source reference sweep
97_{16}	Source other
$A0_{16}$	Auxiliary Data Trailer (no longer used)
$B0_{16}$	True reference sweep (correlation reference)
$B1_{16}$	Radio reference sweep
$B2_{16}$	Radio similarity signal
$B3_{16}$	Wireline reference sweep
$C0_{16}$	Depth
$C1_{16}$	Wind
$C2_{16}$	Current
$C3_{16}$	Voltage
$C4_{16}$	Velocity
$C5_{16}$	Acceleration
$C6_{16}$	Pressure
$C7_{16}$	Tension
$F0_{16}$	Calibration trace (time series)

5–8	$TF_{31} – TF_0$	Channel set start time (four bytes, two's complement, signed binary). This is a microsecond (1/1000000 second) unit of measurement. This number identifies the timing word of the first scan of data in this channel set. In a single scan type record, this would typically be recorded as a zero (an exception might be deep water recording). In multiple scan type records, this number represents the starting time, in microseconds, of the channel set. Start times from −2,147,483,648 to 2,147,483,647 (in 1-μsec increments) can be recorded. In Extended Recording mode (byte 29 of General Header #3 set to 1) this field is not used, and should be set to 0.
9–12	$TE_{31} – TE_0$	Channel set end time (four bytes, two's complement, signed binary). This is a microsecond (1/1000000 second) unit of measurement. This number represent the record end time of the channel set in microseconds. TE may be used to allow the termination of a particular channel set shorter than other channel sets within its scan type. In a single scan type record starting at time zero, Bytes 9–12 would be the length of the record. End times from −2,147,483,648 to 2,147,483,647 μsec (in 1-μsec increments) can be recorded. In Extended Recording mode (byte 29 of General Header #3 set to 1) this field is not used, and should be set to 0.
13–16	$NS_{31} – NS_0$	Number of samples in each trace of this channel set (four bytes, unsigned binary). It can assume a value from 0 to 4,294,967,295. For the purpose of this SEG-D standard, the number of samples NS, sampling interval SR, channel set start time TF, and channel set end time TE should be related by the formula TE=TF+NS×SR.
17–20	$DSM_{31} – DSM_0$	Sample descale multiplication factor (four bytes, IEEE float). The multiplier is be used to descale the data on tape to obtain input voltage in millivolts.

21–23	$C/S_{23} – C/S_0$	Number of channels in this channel set (three bytes, unsigned binary). It can assume a value from 0 to 16,777,215.
24–26	$SR_{23} – SR_0$	Sampling Interval (three bytes, unsigned binary). This is a microsecond (1/1000000 second) unit of measurement. It can assume a value from 0 to 16,777,215.
27	$ARY_7 – ARY_0$	Array Forming (one byte, unsigned binary). Identifies whether the data in this channel set is the result of array forming. 01_{16} No array forming. 02_{16} 2 groups summed, no weighting. 03_{16} 3 groups summed, no weighting. 04_{16} 4 groups summed, no weighting. $0N_{16}$ N groups summed, no weighting. $1N_{16}$ N groups weighted, overlapping, summation.
28	$THE_7 – THE_0$	Number of Trace Header Extensions (one byte, unsigned binary). Must match byte 10 of the Demux Trace Header.
29	$EFH_3 – EFH_0$,	Extended Header flag (half byte, unsigned binary). Set to 1 to indicate that the extended header contains additional information on the channel set.
29	J	Channel gain control method (half byte, unsigned binary). 1_{16} Individual AGC 2_{16} Ganged AGC 3_{16} Fixed gain 4_{16} Programmed gain 8_{16} Binary gain control 9_{16} IFP gain control
30	$VS_7 – VS_0$	Vertical Stack (one byte, unsigned binary). Effective stack order. Set to zero if the trace data was intentionally set to real zero. Set to one if no stack. Set to the effective stack order if the data is the result of stacked data (with or without processing).
31	$CAB_7 – CAB_0$	Streamer Cable number (one byte, unsigned binary). Required for streamer data only. Identifies the number of the streamer cable that will be identified in this block. The starboard-most cable is identified as cable 1 while the Port most cable is N. Zero means that the Streamer Cable number has not been recorded.
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 30_{16} for Channel Set Descriptor block 1.
33–36	$AF_{31} – AF_0$	Alias filter frequency in Hz (four bytes, IEEE float).
37–40	$LC_{31} – LC_0$	Low-cut filter setting in Hz (four bytes, IEEE float).
41–44	$AS_{31} – AS_0$	Alias filter slope in dB per octave (four bytes, IEEE float). A zero indicates the filter is out. (See Appendix E3 for definition.)

45–48	$LS_{31} – LS_0$	Low-cut filter slope (four bytes, IEEE float). A zero indicates the filter is out. (See Appendix E3 for definition.)
49–52	$NT1_{31} – NT1_0$	Notch frequency setting (four bytes, IEEE float). The out filter is written as 0.0 Hz.

The following notch filters are coded in a similar manner:

53–56	$NT2_{31} – NT2_0$	Second notch frequency
57–60	$NT3_{31} – NT3_0$	Third notch frequency
61	$FPH_7 – FPH_0$	Filter phase (one byte, unsigned binary). 00_{16} Unknown 01_{16} Minimum 02_{16} Linear 03_{16} Zero 04_{16} Mixed 05_{16} Maximum
62	$PHU_7 – PHU_0$	Physical unit (one byte, unsigned binary). This is the physical unit measured by this sensor. 00_{16} Unknown 01_{16} Millibar 02_{16} Bar 03_{16} Millimeter/second 04_{16} Meter/second 05_{16} Millimeter/second/second 06_{16} Meter/second/second 07_{16} Newton 08_{16} Kelvin 09_{16} Hertz $0A_{16}$ Second $0B_{16}$ Tesla $0C_{16}$ Volt/meter $0D_{16}$ Volt meter $0E_{16}$ Ampere/meter $0F_{16}$ Volt 10_{16} Ampere
63	X	These bits are undefined by the format and may have any value. Note: In future versions these may be defined, so use with caution.
64	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 31_{16} for Channel Set Description block 2.
65–68	$FDL_{31} – FDL_0$	Filter delay (four bytes, unsigned binary). The signal delay introduced by the filter in microseconds.

69–95	DSC	Description (27 bytes, ASCII text). Channel set description 4 byte free-text as defined by the recording system. Left justified, padded with spaces (20_{16}) for unused characters. zero (00_{16}) is not allowed.
96	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 32_{16} for Channel Set Descriptor block 3.

8.17 DEMUX TRACE HEADER

INDEX BYTE	ABBREVIATION	DESCRIPTION
1,2	$F_1 – F_4$	File Number (two bytes, four digits, BCD). These bytes must be set to $FFFF_{16}$ when the Extended File Number (bytes 18–20) is used.
3	$ST_1 – ST_2$	Scan Type Number (one byte, two digits, BCD).
4	$CN_1 – CN_2$	Channel Set Number (one byte, two digits, BCD). Set to FF_{16} if more than 99 channel sets are required, i.e. Extended Channel Set Number (byte 16–17) is used.
5,6	$TN_1 – TN_4$	Trace Number (two bytes, four digits, BCD). Set to $FFFF_{16}$ if more than 9999 channels are required, i.e. Extended Trace Number (byte 22–24 of Trace Header Extension) is used.
7–9	$T_{15} – T_{-8}$	First Timing Word. These bytes comprise the timing word that would accompany the first sample if these data were written in the multiplexed format. To obtain the exact sample timing, the actual sample skew time (byte 11 multiplied by the base can interval) must be added to the time recorded in Bytes 7–9.
10	$THE_7 – THE_0$	Trace Header Extensions (one byte, unsigned binary). Indicates the number of Trace Header Extension blocks (32 bytes each). Set to zero when no extensions are used. Maximum allowed is 255. Channels within the same channel set must have the same number of Trace Header Extensions.
11	$SSK_{-1} – SSK_{-8}$	Sample Skew (one byte, binary fraction). The fractional skew value represents the fractional part of the base Scan Interval (Byte 23 of General Header Block #1.)
12	$TR_7 – TR_0$	Trace edit (one byte, unsigned binary). 00_{16} No edit applied to this trace. 01_{16} Trace part of dead channels for roll-on or roll-off spread. Trace intentionally zeroed. 02_{16} Trace intentionally zeroed. 03_{16} Trace has been edited. This flag will indicate that the acquisition system has modified one or more samples of this trace. Other codes are undefined in Rev 3.0.
13–15	$TW_{15} – TW_{-8}$	Time Break Window (three bytes, unsigned binary with two bytes integer and one byte fraction). Bytes 13, 14, and 15 are included as an integrity check on time break. They comprise the timing word of the scan in which TWI changed to a one.

16,17	$EN_{15} – EN_0$	Extended Channel Set Number (two bytes, unsigned binary). Allows Channel Set Numbers beyond the 99 which can be indicated in byte 4. To allow Channel Set Numbers greater than 99, or to allow use of a binary channel set number, set byte 4 to FF_{16} and use bytes 16 and 17 for the Channel Set Number.
18–20	$EFN_{23} – EFN_0$	Extended File Number (three bytes, unsigned binary). Allows File Numbers beyond the 9999 which can be indicated in bytes 1 and 2. To allow File Numbers greater than 9999, or to allow use of a binary file numbers, set bytes 1 and 2 to $FFFF_{16}$ and use bytes 18, 19, and 20 for the File Number.

8.18 TRACE HEADER EXTENSION

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$RLN_{23} – RLN_0$	Receiver Line Number (three bytes, two's complement, signed binary).
4–6	$RPN_{23} – RPN_0$	Receiver Point Number (three bytes, two's complement, signed binary).
7	$RPI_7 – RPI_0$	Receiver Point Index (one byte, two's complement, signed binary). This index allows several locations for the receiver group in the grid, the original value is 1 and that value is incremented by 1 every time the receiver is moved, even when it is moved back to the previous location). Receiver Point Index may also be used for downhole seismic to extend the numerical range of the Depth Index.
8	$RI_7 – RI_0$	Re-shoot Index (one byte, unsigned binary). The re-shoot index is used to indicate that this receiver has been moved back here for a re-acquisition (shot has been recorded earlier, and this record should replace it). Re-shoot index starts at 0 (indicates recorded for first time).
9	$GI_7 – GI_0$	Group Index (one byte, unsigned binary). Group index is used to indicate this receiver is part of a set of receivers needed to be processed as a unit. This could be a component of a three-component sensor at a given location. 0 means receiver is not part of a group.
10	$DI_7 – DI_0$	Depth Index (one byte, unsigned binary). The depth index is used to indicate receivers recorded at different depths on the same location. 0 is not allowed. Starts at 1 closest to the surface, and counts up when moving downwards.
11–15	$ERLN_{23} – ERLN_{-16}$	Extended Receiver Line Number (Signed binary, three bytes integer, two bytes fractional). Allows fractional Receiver Line Numbers. Only valid if bytes 1–3 in this Trace Header Extension are set to $FFFFFF_{16}$. The representation of a negative receiver line number is the two's complement of the integer formed by multiplying the absolute value of the receiver line number by 65536.
16–20	$ERPN_{23} – ERPN_{-16}$	Extended Receiver Point Number (Signed binary, three bytes integer, two bytes fractional). Allows fractional Receiver Point Numbers. Only valid if bytes 4–6 in this Trace Header Extension are set to $FFFFFF_{16}$. The representation of a

		negative receiver line number is the two's complement of the integer formed by multiplying the absolute value of the receiver line number by 65536.
21	$SEN_7 - SEN_0$	Sensor Type recorded on this trace (one byte, unsigned binary) 00_{16} Not defined 01_{16} Hydrophone (pressure sensor) 02_{16} Geophone (velocity sensor) Vertical 03_{16} Geophone, Horizontal, inline 04_{16} Geophone, Horizontal, cross-line 05_{16} Geophone, Horizontal, other 06_{16} Accelerometer, Vertical 07_{16} Accelerometer, Horizontal, inline 08_{16} Accelerometer, Horizontal, crossline 09_{16} Accelerometer, Horizontal, other 15_{16} Electric Dipole 16_{16} Magnetic coil other values are not defined at the present time
22–24	$ETN_{23} - ETN_0$	Extended Trace Number (three bytes, unsigned binary). Extended trace number to allow up to 16,777,215 traces in one channel set. Field is only valid if bytes 5–6 in Demux Trace Header is set to $FFFF_{16}$.
25–28	$NS_{31} - NS_0$	Number of samples in this trace (four bytes, unsigned binary) It can assume a value from 0 to 4,294,967,295. Whenever possible, this should be the same as bytes 13–16 of channel set descriptor.
29	$MV_7 - MV_0$	Sensor moving (one byte, unsigned binary). Sensor moving or stationary. Set to 1 if sensor is moving during the record, 0 if it is stationary.
30	X	Undefined
31	$PHU_7 - PHU_0$	Physical unit (1 byte, unsigned binary). This is the physical unit measured by this sensor. Same as byte 60 of the channel set header
32	$HT_7 - HT_0$	Header block type (1 byte, unsigned binary) Set to 40_{16} for Trace header extension 1.

8.19 SENSOR INFO TRACE HEADER EXTENSION (optional)

This header contains measurements and other information used for

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–8	$ITT_{63} - ITT_0$	Instrument Test Time (SEG-D 8 byte timestamp). Time of last instrument test. Set to 0 if not recorded.

9–12	$STY_{31} – STY_0$	Sensor Sensitivity (four bytes, IEEE float). This is the signal sensitivity for the sensor. Divide the sample value by this number to achieve a physical unit (as specified in bytes 31). Set to 0 if not specified (physical unit unknown).
13	$ITR_7 – ITR_0$	Instrument Test Result (one byte, unsigned binary). The following values are allowed 00_{16} Unknown (untested) 01_{16} Passed 02_{16} Failed 03_{16} Uncertain (warning during IT)
14–31	SN	Serial Number (18 bytes, ASCII text) . Serial number of sensor for this receiver. Left-justified, padded with space (20_{16}) characters.
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 41_{16} for Sensor Info Trace Header Extension.

8.20 TIMESTAMP HEADER (optional)

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–8	$TZD_{63} – TZD_0$	Time Zero for this data block (eight bytes, SEG-D timestamp). The time of sample 0 in the data described by this header. Typically inserted into Trace Header in Extended Recording Mode, the timestamp is then the time of first sample in the trace. If the recorder does not know the absolute time, this timestamp must be relative to the timestamp in General Header Block #3 (bytes 1–8), i.e. number of microseconds since time zero of the record.
9–31	X	These byte fields are undefined by the format and may have any value. Note: In future versions these may be defined, so use with caution.
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. See Table 2 for an overview of header types. Set to 42_{16} for Timestamp Header.

8.21 SENSOR CALIBRATION HEADER (optional)

Use of the sensor calibration header supports between 1 and a maximum of approximately 500 points (the maximum size of the trace header). The maximum depends on what other blocks are present in the trace header. If a higher resolution calibration is required, a calibration channel/channelset must be created.

INDEX BYTE	ABBREVIATION	DESCRIPTION

1–4	$FR1_{31} – FR1_0$	Frequency 1 (four bytes, IEEE float). The frequency of first calibration value. Set to 0.0 if not used.
5–8	$AM1_{31} – AM1_0$	Amplitude 1 (four bytes, IEEE float). The amplitude of first calibration value.
9–12	$PH1_{31} – PH1_0$	Phase 1 (four bytes, IEEE float). The phase of first calibration value.
13–16	$FR2_{31} – FR2_0$	Frequency 2 (four bytes, IEEE float). The frequency of second calibration value. Set to 0.0 if not used.
17–20	$AM2_{31} – AM2_0$	Amplitude 2 (four bytes, IEEE float). The amplitude of second calibration value.
21–24	$PH2_{31} – PH2_0$	Phase 2 (four bytes, IEEE float). The phase of second calibration value.
32	$CA_7 – CA_0$	Calibration applied (one byte, unsigned binary). Set to 1 if calibration has already been applied, i.e. the trace has been corrected with the calibration function described by the values in the Sensor Calibration Header blocks, set to 0 if trace is not corrected. Note: Multiple Sensor Calibration Header blocks are usually needed to describe a calibration function.
26–31	X	Undefined
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). The header type indicates what type of information is contained in this header. See Table 2 for an overview of header types. Set to 43_{16} for Sensor Calibration Header.

8.22 *TIME DRIFT HEADER (optional)*

This block may be used when data from sensors with drifting clocks are merged into one SEGD record. Data must be clock drift corrected prior to creating a SEGD shot, i.e. SEGD only supports recording data with one single clock reference system.

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–8	$TD_{63} – TD_0$	Time of deployment (eight bytes, SEG-D timestamp). Time of sensor unit deployment, i.e. when time drift was measured before deployment.
9–16	$TR_{63} – TR_0$	Time of retrieval (eight bytes, SEG-D timestamp). Time of sensor unit retrieval, i.e. when the time drift was measured after retrieval.
17–20	$OD_{31} – OD_0$	Time offset at deployment (four bytes, signed binary). Time offset value at deployment in number of microseconds. Range approximately ±2147 seconds.
21–24	$OR_{31} – OR_0$	Time offset at retrieval (four bytes, signed binary). Time offset value at retrieval in number of microseconds. Range approximately ±2147 seconds.

25	$TDC_7 – TDC_0$	Time drift corrected (one byte, unsigned binary). Set to 1 if time drift correction has been applied, 0 if not. Note: SEGD only supports recording data from one clock reference system. *Data must be time drift corrected prior to record creation if data acquired from sensors with decoupled (drifting) clocks is merged into one SEGD record.*
26	$CM_7 – CM_0$	Time drift correction method (one byte, unsigned binary). Method used to correct for time drift. The following methods are supported: 00_{16} Uncorrected 01_{16} Linear correction (values in this header used) FF_{16} Other, manufacturer defined method used for correction
27–31	X	Undefined
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 44_{16} for Time Drift Header.

8.23 ORIENTATION HEADER (optional)

The orientation header provides the means to specify the orientation at a given position for 3C data.

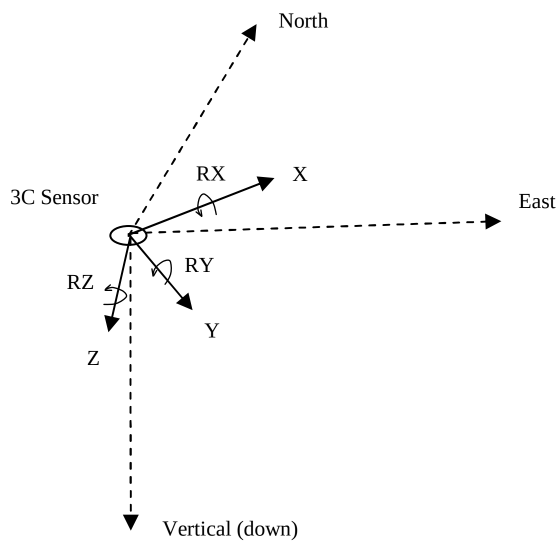


Figure shows multicomponent measurements and their respective rotation measurements relative to the reference system (north/east/vertical)

X is defined as in-line, Y is cross-line and Z is vertical. Positive vertical is down. The rotation follows the right-hand rule. RX is defined as rotation around X axis, RY around Y axis, and RZ around Z axis. Positive value as defined by right-hand rule.

X axis will for cable acquisition (towed streamers, seabed cables etc.) be along the cable.

The rotation measurement in the sensor is expected to result in a rotation to the horizontal plane, however rotation to true north will in many cases require an external measurement. This value is hence extracted to a separate field in the rotation header, and may also be recorded as 0 if not available at the time of acquisition.

$$\begin{bmatrix} C_{north} \\ C_{east} \\ C_{vertical} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} * \begin{bmatrix} \mathrm{C}_{\mathrm{x}} \\ C_y \\ C_z \end{bmatrix} \quad (1)$$

The rotation described in bytes 1–16 in the orientation header must fulfill equation (1) above. The group index for multicomponent data must hence be numbered such that

1 = X axis
2 = Y axis
3 = Z axis

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–4	RX_{31} – RX_0	Rotation angle around the X axis in radians (four bytes, IEEE float).
5–8	RY_{31} – RY_0	Rotation angle around the Y axis in radians (four bytes, IEEE float).
9–12	RZ_{31} – RZ_0	Rotation angle around the Z axis in radians (four bytes, IEEE float).
13–16	RO_{31} – RO_0	Reference Orientation (four bytes, IEEE float). Rotation angle for the system described in bytes 1–12 relative to true north, i.e. the rotation angle around the Z axis for the X direction to point towards true north, This will be the cable direction or housing orientation, depending of the value of byte 25. Recorded as 0 if not applicable. If value is valid, byte 26 should be set to 1. May be recorded as 0 if the value is unknown at the time of recording. The value must then be made available to the processing/QC systems through other means. Byte 26 should then be set to 0.
17–24	TS_{63} – TS_0	Time Stamp (eight bytes, SEG-D timestamp). Timestamp for which the rotation applies to, i.e. which sample the rotation applies to. Allows multiple rotation measurements during a record. May be recorded as 0, the values then applies to first sample in trace.
25	OT_7 – OT_0	Orientation Type horizontal plane (one byte, unsigned binary). The rotation angle in bytes 9–12 is referenced to this coordinate system. The following values are defined 00_{16} None/Unknown 01_{16} Earth true north 02_{16} Cable direction (e.g. marine/seabed streamer) 03_{16} Sensor housing orientation 04_{16} Survey orientation 05_{16} Magnetic north 09_{16} Other

26	$ROV_7 – ROV_0$	Reference Orientation Valid (one byte, unsigned binary). Set to 1 if bytes 13–16 contains a valid value, otherwise recorded as 0.
27	$RA_7 – RA_0$	Rotation Applied (one byte, unsigned binary). Set to 1 if rotation in bytes 1–12 has already been applied, otherwise recorded as 0. Allows recording of orientation values for QC purposes even though the rotation has already been done
28	$RNA_7 – RNA_0$	Rotation to North Applied (one byte, unsigned binary). Set to 1 if values in traces has already been rotated to true north, otherwise recorded as 0. Rotation in bytes 1–16 has been applied to trace values. Allows recording of orientation values for QC purposes even though the rotation has already been done. If byte 28 is 1, byte 27 must also be set to 1.
29–31	X	These bytes are undefined by the format and may have any value. Note: In future versions these may be defined, so use with caution.
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 60_{16} for Orientation Header.

8.24 MEASUREMENT BLOCK (optional)

This block is used to standardize storage of different types of common measurements.

NOTE: This block should not be used until final agreement with Energistics is reached regarding the unit of measurement system, and the proper resources are available.

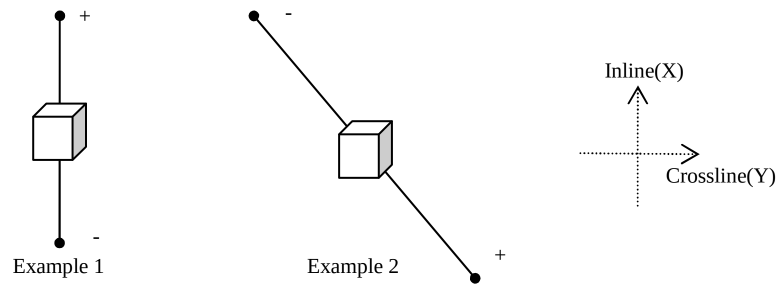
INDEX BYTE	ABBREVIATION	DESCRIPTION
1–8	$TS_{63} – TS_0$	Time Stamp (eight bytes, SEG-D timestamp). Timestamp for the measurement. Timestamp for time related measurements like Uphole time should be set to time zero for record (same as timestamp in General Header #3) if they are measuring the offset from start of record.
9–12	$VA1_{31} – VA1_0$	Measurement value 1 (four bytes, IEEE float). Value of the measurement.
13–14	$UOM1_{15} – UOM1_0$	Unit Of Measure 1 (two bytes, unsigned binary). This is the Energistics Unit Of Measure identifier for used for measurement 1. *Link to the Energistics unit of measure tables will be inserted when available. Until then, please do not use the Measurement block.*

15	$MT1_7 – MT1_0$	Measure type 1 (one byte, unsigned binary). Type of measurement as defined in ….. (*proper reference to be added later*). This is a SEG specific ID defining the context in which the Energistics Unit Of Measure is used. If the Energistics Unit Of Measure is used directly, i.e. no additional SEG-D specific context information is needed, this field should be set to 0. *A couple of examples will be added later* To add entries to the measurement list, please enter a request at the technical standards page at www.seg.org.
16–19	$VA2_{31} – VA2_0$	Measurement value 2 (four bytes, IEEE float). Value of the measurement.
20–21	$UOM2_{15} – UOM2_0$	Unit Of Measure 2 (two bytes, unsigned binary). See UOM1.
22	$MT2_7 – MT2_0$	Measure type 2 one byte, unsigned binary). See MT1
23–26	$VA3_{31} – VA3_0$	Measurement value 3 (four bytes, IEEE float). Value of the measurement.
27–28	$UOM3_{15} – UOM3_0$	Unit Of Measure 3 (two bytes, unsigned binary). See UOM1.
29	$MT3_7 – MT3_0$	Measure type 3 one byte, unsigned binary). See MT1
30–31	X	Undefined
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 61_{16} for Measurement Block.

8.25 ELECTROMAGNETIC SRC/RECV DESC BLOCK (optional)

INDEX BYTE	ABBREVIATION	DESCRIPTION
1–3	$LX_{23} – LX_0$	Equipment Dimension X direction (three bytes, unsigned binary). Length of measurement equipment (e.g. antenna) in X (inline) direction in number of centimeters (1/100 meters). Maximum length is 16777215 centimeters, or approximately 16777 meters.
4–6	$LY_{23} – LY_0$	Equipment Dimension Y direction (three bytes, unsigned binary). Length of measurement equipment (e.g. antenna) in Y (crossline) direction in number of centimeters (1/100 meters). Maximum length is 16777215 centimeters, or approximately 16777 meters.
7–9	$LZ_{23} – LZ_0$	Equipment Dimension Z direction (three bytes, unsigned binary Length of measurement equipment (e.g. antenna) in Z (height) direction in number of centimeters (1/100 meters). Maximum length is 16777215 centimeters, or approximately 16777 meters.

10	$PT_7 – PT_0$	Positive terminal (one byte, unsigned binary). Determines the positive terminal for the antenna. Bits 0–2 determines if the positive end (i.e. antenna end with largest positive value) in X, Y and Z direction respectively is the positive terminal. Set bit to 1 if positive end is positive terminal, 0 if it is negative. Example: The antenna is 15 meters in inline direction, positive terminal is at the positive end of the antenna. LX is set to $0005DC_{16}$, LY to 000000_{16}, LZ to 000000_{16}, PT to 01_{16}. Example 2: Antenna is 25 meters diagonally in X-Y direction, positive terminal is negative X, positive Y value. Length in X and Y direction is $\mathrm{sqrt}(25^2/2)=17.68$ meters. LX is set to $0006E8_{16}$, LY to $0006E8_{16}$, LZ to 000000_{16}, PT to 02_{16}.



11–13	$OX_{23} – OX_0$	Equipment Offset X direction (three bytes, signed binary). Measurement equipment (e.g. antenna) offset from center of equipment in X (inline) direction in number of centimeters (1/100 meters). Maximum offset is +/-8388607 centimeters, or approximately 8388 meters.
14–16	$OY_{23} – OY_0$	Equipment Offset Y direction (three bytes, signed binary). Measurement equipment (e.g. antenna) offset from center of equipment in Y (crossline) direction in number of centimeters (1/100 meters). Maximum offset is +/-8388607 centimeters, or approximately 8388 meters.
17–19	$OZ_{23} – OZ_0$	Equipment Offset Z direction (three bytes, signed binary). Measurement equipment (e.g. antenna) offset from center of equipment in Z (height) direction in number of centimeters (1/100 meters). Maximum offset is +/-8388607 centimeters, or approximately 8388 meters.
20–31	X	Undefined.
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 45_{16} for Electromagnetic Src/Recv Desc block

8.26 GENERAL TRAILER (Optional)

The general trailer consists of data block units of varying size. Each data block starts with a description block (32 bytes) followed by data (either binary or ASCII) in a multiple of 32 byte.

The content of the trailer data blocks is completely recording system defined. However, some types have been predefined to simplify exchange of data. The predefined types (e.g. edits) can be found in table listed for Byte 1 in the General Trailer Description block below.

Binary data padded with zeros (00_{16}), ASCII data padded with spaces (20_{16}), until 32 byte limit.

Layout of the general trailer will be like this:

[GT desc block (32 byte)][Block 1 (x*32 byte)][GTDB][Block 2][GTDB][Block 3]…

This is done to simplify appending data.

8.26.1 GENERAL TRAILER DESCRIPTION BLOCK (optional)

Field	Description	Bytes	Start – end byte
1	Trailer Block type/id	1	1
2	ASCII/Binary data	1	2
	Undefined (zero-filled)	2	3–4
3	Size (unsigned binary)	4	5–8
4	Description (ASCII) if type other	16	9–24
	Undefined (zero-filled)	6	25–31
5	Header block type (31_{16})	1	32

INDEX BYTE	ABBREVIATION	DESCRIPTION
1	$BT_7 – BT_0$	Block type, (one byte, unsigned binary). This describes the type of data in this block. 01_{16} Edits 02_{16} Positions 03_{16} Positions raw 10_{16} Text comments 11_{16} Observer log FF_{16} Other (user defined)
2	$AB_7 – AB_0$	ASCII or binary (one byte, unsigned binary). States if data in this block is binary or ASCII text. 01_{16} Binary 02_{16} ASCII
3–4	X	Undefined, set to 0.

5–8	$SI_{31} – SI_0$	Block size (four bytes, unsigned binary). Number of 32 byte blocks in this data block. Block size in bytes = 32 * SI
9–24	DE	Description (16 bytes, ASCII). A description of the contents of this data block. Must be filled out if block type is Other (11_{16}). Unused characters are filled with ASCII space (20_{16}) characters.
25–31	X	Undefined, set to 0.
32	$HT_7 – HT_0$	Header block type (one byte, unsigned binary). Set to 70_{16} for General Trailer Description Block.

Note: All types of trailer records are optional.

Appendix A: Manufacturers of Seismic Field Recorders

Code No.	Manufacturer Name and Address
01	Alpine Geophysical Associates, Inc. (Obsolete) 65 Oak St. Norwood, New Jersey
02	Applied Magnetics Corporation (See 09) 75 Robin Hill Rd. Goleta, California 93017
03	Western Geophysical Exploration Products (formerly Litton Resources Systems) 3600 Briarpark Drive, Houston, Texas 77042
04	SIE, Inc. (Obsolete) 5110 Ashbrook Houston, Texas 77036
05	Dyna-Tronics Mfg. Corporation (Obsolete) 5820 Star Ln., Box 22202 Houston, Texas 77027
06	Electronic Instrumentation, Inc. (Obsolete) 601 Dooley Rd., Box 34046 Dallas, Texas 75234
07	Halliburton Geophysical Services, Inc.,(formerly, Electro-Technical Labs, Div. of Geosource, Inc.) 6909 Southwest Freeway Houston, Texas 77074
08	Fortune Electronics, Inc. (Obsolete) 5606 Parkersburg Dr. Houston, Texas 77036
09	Geo Space Corporation 7334 Gessner Houston, Texas 77040
10	Leach Corporation (Obsolete) 405 Huntington Dr. San Marino, California

11	Metrix Instrument Co. (Obsolete) 8200 Westglen Box 36501 Houston, Texas 77063	
12	Redcor Corporation (Obsolete) 7800 Deering Ave., Box 1031 Canoga Park, California 91304	
13	Sercel (Societe d'Etudes, Recherches Et Constructions Electroniques) 25 X, 44040 Nantes Cedex, France	
14	Scientific Data Systems (SDS), (Obsolete) 1649 Seventeenth St. Santa Monica, California 90404	
15	Texas Instruments, Inc. P.O. Box 1444 Houston, Texas 77001	
17	GUS Manufacturing, Inc. P.O. Box 10013 El Paso, Texas 79991	
18	Input/Output, Inc. 12300 Parc Crest Dr. Stafford, Texas 77477	
19	Geco-Prakla Transition Zone Product Development (formerly Terra Marine Engineering) 10420 Miller Road Dallas, Texas 75238	
20	Fairfield Industries, Incorporated 10627 Kinghurst Houston, Texas 77099	
22	Geco-Prakla Buckingham Gate, Gatwick Airport West Sussex, RH6 ONZ, UK	
31	Japex Geoscience Institute Akasaka Twin Towers Bldg. 2; 2-17-22, Akasaka Minato-ku; Tokyo 107, Japan	1991
32	Halliburton Geophysical Services, Inc. 6909 Southwest Freeway Houston, Texas 77074	1991

33	Compuseis, Inc. 8920 Business Park Dr, Ste 275, Austin, Texas 78759	1993
34	Syntron, Inc. 17200 Park Row Houston, Texas 77084	1993
35	Syntron Europe Ltd. Birchwood Way Cotes Park Industrial Estates Somercotes, Alfreton, Dergyshire DE55 4QQ, U.K.	1993
36	Opseis 7700 E. 38th St. Tulsa, OK 74145	1994
39	Grant Geophysical 16850 Park Row Houston, Tx 77084	1995
40	Geo-X Suite 900, 425 1st St SW Calgary, Alberta, Canada T2P3L8	1996
41	PGS Inc. 16010, Barkers Point Lane, Houston Texas 77079	2001
42	SeaMap UK Ltd. Unit 31 The Maltings Charlton Estate Shepton Mallet BA4 5QE UK	2003
43	Hydroscience Hydroscience Technologies, Inc. 5101 Airport Road Mineral Wells, Texas 77478 USA	2004
44	JSC "SPECIAL DESIGN BUREAU FOR SEISMIC INSTRUMENTATION" 129, Krainyaya Str., 410019 Saratov, RUSSIA	2006
45	Fugro Hoffsveien 1C N-0213 Oslo	2006

	NORWAY	
46	ProFocus Systems AS Spelhaugen 20 5147 Fyllingsdalen NORWAY	2006
47	Optoplan AS Haakon VII's gate 17 7041 Trondheim NORWAY	2008
48	Wireless Seismic Inc 12996 Somerset Drive Grass Valley, CA 95945 USA	2008

Appendix B: Glossary

Attribute - A named item of information or data pertaining to an object.
Base scan interval - The time between timing words. A base scan interval usually contains one scan but under some conditions may contain multiple subscans.
Beginning of tape mark (BOT) - An indelible mark (e.g., reflector) near the beginning of the tape that indicates the start of the region in which recorded data is permitted.
Block - The data between gaps on tape. On modern tape devices block may be a logical marks only, not an actual gap on the tape. A block usually refers to the smallest amount of data possible to read from the device.
Channel set - One or more channels sampled at the same sampling interval and containing the same filter, fixed gain, and other fixed parameter information.
Channel set descriptor - A unit of the scan type header describing the parameters of a channel set.
Data recording method - The arrangement of bits to represent samples on tape.
End of data flag (EOD) - A special record or condition on the tape used to indicate the end of data on the tape.
End of file mark (EOF) - A special record or condition on the tape that indicates the end of a tape file. Also called File Mark.
End of tape warning (ETW) - An indelible mark (e.g., reflector), located a required minimum distance from the physical end of the tape that serves as a warning.
Field record – The data between two filemarks on media that supports this (e.g. tape). The field record may consists of multiple SEG-D records. For media without filemarks the field record equals a SEG-D record. (See also **File**)
File - Literally, it is all of the blocks between file marks. The term **Field File** is sometimes used instead of File. It is all data recorded from a one or more energy impulses or sweeps, stored in one or more SEG-D records. It may also be the sum of a number of energy impulses or sweeps, or data recorded without any source actions. (See also **Field record.**)
Filemark - A special record or state of a media index that indicates the end of a physical file.
Format Data - recording method combined with a multiplexed/demultiplexed indicator (see general header Bytes 3 and 4).
General header - The first header in the header block. It contains information common to the entire record.
Index byte - The byte number of some particular parameter within the general or scan type header.
Organization code - A number assigned by the Petroleum Open Standards Consortium (POSC) to an organization that identifies the organization and represents schemas and dictionaries defined and administered by the organization.
Packed BCD - Binary coded decimal digits represented by four data bits.
Partition - An independent recording area resulting from physical formatting of the media that can be mounted as though it were a single volume.
Physical blocks (on tape) - A collection of contiguous bytes recorded as a unit on a longitudinal tape or a track set on helical scan tape. Physical blocks on longitudinal tape are separated by interblock gaps.
Record – A SEG-D record is a collection of data traces and meta-data describing those traces. It is also allowed to create a record with the meta-data only. The record consists of a header (mandatory), data traces and a record trailer. Sometimes the terms **Shot record** or **Seismic record** is used to describe a SEG-D record.
Sample skew - The fraction of the base scan interval between the timing word and the actual time the sample was taken in a base scan interval (not related to position on tape).
Sampling interval - The interval between readings such as the time between successive samples of a digital seismic trace.
Scan - One complete sequence of events, such as sampling all channels. Data recorded during a base scan interval.
Scan interval - The interval between readings of all samples contained in a scan type.

Scan type - One complete set of channel sets which make up a scan. A seismic record contains multiple scans, and may or may not contain more than one scan type.
Scan type header - A header containing one or more channel set descriptors and the skew information.
Schema - A formalized description of the encoding of information defined by a logical model, typically in terms of a data model.
Storage set - An ordered set of storage units, the position in the set is specified in the storage unit label.
Storage unit - A logical volume of data containing on or more logical files.
Subscan - A set of samples containing one sample for each channel in a channel set.
Tape file - Is the data contained between two File Marks or between an File Mark at its beginning and an EOD at its end. A typical implementation of EOD is an empty tape file, i.e., two consecutive File Marks. Some systems implement EOD as two or more (possibly many) File Marks.
Tape record - A sequence of data bytes treated as a unit by the tape I/O subsystem. The application provides the number of bytes in the tape record when writing and is returned the number of bytes in the tape record when reading. A tape record has an identifiable beginning on the tape, which need not be on the boundary of a physical block, and which is locatable by the tape I/O subsystem.

Time break window - Time interval in which time break is expected. If time break does not occur by the end of the window, internal time break is generated.
Trace - A record of one seismic channel within a scan type. A collection of a sequential set of points from one seismic channel.
Trace block - A block containing the data of one trace or a part of a trace with constant parameters.

Appendix C: API Producer Organization Code

C.1 Scope

Table 1 below contains a list of *organization codes* assigned by the *Petrotechnical Open Standards Consortium (POSC)* for use in API Recommended Practice 66. *Several of the organization codes in this appendix are historical in nature and reflect the well log origins of API Recommended Practice 66.*

This specification was originally prepared by the Subcommittee on Standard Format for Digital Well data. In June 1996 it was designated a recommended practice by the American Petroleum Institute [API] Exploration & Production Department's Executive Committee on Drilling and Production Practices. In June 1998 stewardship of the document was transferred from API to Petroleum Open Standards Consortium [POSC]. This organization was relaunched as Energistics in December 2006.

C.2 Assignment of Organization Codes

Organization codes are assigned by Energistics, which maintains the current list of codes. To request a new organization code, contact:

Energistics
24 E. Greenway Plaza
Suite 1315
Houston, TX 77046-2414 USA
+1 713 784-1880 telephone
+1 713 784-9219 fax
info@energistics.org
http://www.energistics.org/

The following is copied from

http://www.posc.org/technical/data_exchange/RP66/V2/rp66v2.html

This document, *POSC RP66 V2*, is a specification of Petrotechnical Open Standards Consortium.

This specification is based on and is equivalent in content to the document *Recommended Practices for Exploration and Production Data Digital Interchange: API RECOMMENDED PRACTICE 66, V2: Second Edition, June 1996* published by the American Petroleum Institute (API). The acronym ***RP66V2*** is retained for historical reasons; POSC does not designate specifications as "Recommended Practices".

http://www.posc.org/technical/data_exchange/RP66/V2/rp66v2_appa.html

Last Updated 30 Sep 2008

Table 1— Organization Codes

Table 1 - Organization Codes	
Code	Organization

0	Subcommittee On Recommended Format For Digital Well Data, Basic Schema
1	Operator
2	Driller
3	Mud Logger
9	Amerada Hess
10	Analysts, The
15	Baker Hughes Inteq
20	Baroid
30	Birdwell
40	Reeves (1 Jan 99; formerly BPB)
50	Brett Exploration
60	Cardinal
65	Center Line Data
66	Subcommittee On Recommended Format For Digital Well Data, DLIS Schema
70	Century Geophysical
77	CGG Logging, Massey France
80	Charlene Well Surveying
90	Compagnie de Services Numerique
95	Comprobe
100	Computer Data Processors
110	Computrex
115	COPGO Wood Group
120	Core Laboratories
125	CRC Wireline, Inc.
126	Crocker Data Processing Pty Ltd
127	Tucker Wireline Services (formerly Davis Great Guns Logging, Wichita, KS)
130	Digigraph
137	Tucker Technologies (formerly Digital Logging Inc.), Tulsa, OK.
140	Digitech
145	Deines Perforating
148	Drillog Petro-Dynamics Limited
150	Baker Atlas (formerly Dresser Atlas)
160	Earthworm Drilling
170	Electronic Logging Company
180	Elgen
190	El Toro

200	Empire
205	Encom Technology, Ltd.
206	Ensign Geophysics, Ltd.
210	Frontier
215	Geolog
217	Geoshare
218	GEO·X Systems Ltd.
220	G O International
230	Gravilog
240	Great Guns Servicing
250	Great Lakes Petroleum Services
260	GTS
268	Guardian Data Seismic Pty. Ltd.
270	Guns
280	Halliburton Logging
285	Horizon Production Logging
290	Husky
300	Jetwell
305	Landmark Graphics
310	Lane Wells
315	Logicom Computer Services (UK) Ltd
320	Magnolia
330	McCullough Tool
332	Mitchell Energy Corporation
335	Paradigm Geophysical (formerly Mincom Pty Ltd)
337	MR-DPTS Limited
338	NRI On-Line Inc
339	Oilware, Inc.
340	Pan Geo Atlas
342	Pathfinder Energy Services
345	Perfco
350	Perfojet Services
360	Perforating Guns of Canada
361	Petcom, Inc.
362	Petroleum Exploration Computer Consultants, Ltd.
363	Petrologic Limited
366	Phillips Petroleum Company
368	Petroleum Geo-Services (PGS)

370	Petroleum Information
380	Petrophysics
390	Pioneer
392	The Practical Well Log Standards Group
395	IHS Energy Log Services (formerly Q. C. Data Collectors)
400	Ram Guns
410	Riley's Datashare
418	RODE
420	Roke
430	Sand Surveys
440	Schlumberger
450	Scientific Software
460	Seismograph Service
462	SEGDEF
463	SEG Technical Standards High Density Media Format Subcommittee
464	Shell Services Company
465	Stratigraphic Systems, Inc.
467	Sperry-Sun Drilling Services
468	SEPTCO
469	Sercel, Inc.
470	Triangle
475	Troika International
480	Welex
490	Well Reconnaissance
495	Wellsite Information Transfer Specification (WITS)
500	Well Surveys
510	Western
520	Westronics
525	Winters Wireline
530	Wireline Electronics
540	Worth Well
560	Z & S Consultants Limited
999	***Reserved for local schemas***
1000	POSC

Appendix D: Coordinate Reference System identification

The following stanzas are extended versions of the SEG-Y Rev 1 Location Data stanza. The structure for Extended Textual stanzas is described in section 6.1 of SEG-Y Revision 1 (http://www.seg.org/SEGportalWEBproject/prod/SEG-Publications/Pub-Technical-Standards/Documents/seg_y_rev1.pdf) and summarized in the following:

The ground rules for stanzas that use this schema are as follows:

- Each line consists of a keyword/value pair in the form “keyword = value”.
- The keywords and values can contain any printable character except double right or double left parentheses or the equal sign. However, the use of punctuation characters in keywords is not recommended.
- The case of a keyword is not significant.
- For readability, spaces within a keyword are allowed but ignored. Thus the keyword “Line Name” refers to the same keyword as “LINENAME”.
- The value associated with a keyword begins with the first non-blank character following the equal sign and extends to the last non-blank character on the line.
- The value field for a keyword may consist of multiple subfields, separated by commas (“,”, EBCDIC $6B_{16}$ or ASCII $2C_{16}$).
- Blank lines are ignored.
- If the first non-blank character in a line is the hash sign (“#”, EBCDIC $7B_{16}$ or ASCII 23_{16}), the line is treated as a comment and ignored.
- If the last non-blank character on a line is an ampersand (“&”, EBCDIC 50_{16} or ASCII 16_{16}), the next line is considered to be a continuation of the current line (i.e. the next line is concatenated with the current line, with the ampersand removed).

Each line in an Extended Textual File Header ends in carriage return and linefeed (EBCDIC $0D25_{16}$ or ASCII $0D0A_{16}$)

D.1 Location Data

Location stanzas provide the definitions for the system to which coordinates given in this SEG-D format file are referenced. Without this information the coordinates are ambiguous. The stanzas for this coordinate reference system definition are described in D.1.1. Examples are given in D.1.2.

If during acquisition a coordinate transformation has been applied to derive the coordinates defined through the location stanza (for example when the location data has been transformed from the GPS system's WGS 84 coordinates to coordinates referenced to a local Coordinate Reference System), details of the transformation applied should additionally be given through a Coordinate Transformation stanza. The stanzas for this transformation are defined in D.1.3. Examples are given in D.1.4.

Some keywords in both the Location Data stanza (D.1.1 table 1.2) and the Location Data Coordinate Transformation stanza (D.1.3 table 1.4) require entry of a parameter from the EPSG Geodetic Parameter Dataset. In some cases, such as unit of measure, the user has the ability to specify the Unit Name, Unit Code and/or Conversion Factor (to canonical Unit of Measure). The EPSG dataset is available at www.epsg.org or online at www.epsg-registry.org.

<u>Degree Representation</u>
A **sexagesimal degree** is defined as “a plane angle represented by a sequence of values in degrees, minutes and seconds. In the case of latitude or longitude, may also include a character indicating hemisphere.” For example: 50.0795725 degrees is represented as 50º04'46.461"N sexagesimal degrees. To store this in numeric form requires three or four fields.

The EPSG dataset includes a pseudo-unit named sexagesimal DMS which allows the storage of a sexagesimal degree in one numeric field. The format is:

> *signed degrees - period - minutes (two digits) - integer seconds (two digits) - fraction of seconds (any precision). Must include leading zero in minutes and seconds and exclude decimal point for seconds.*

For example 50º04'46.461"S may be represented as −50.0446461 in sexagesimal DMS.
This "unit" is used for documenting (mainly map projection) parameter values where the values must be exactly preserved and rounding through conversion to decimal degrees cannot be tolerated.
<u>Note that the value given in decimal degrees (e.g. 50.0795725) will differ from its equivalent value given in sexagesimal DMS (50.0446461) only in the decimal part and that the two representations may be easily confused.</u> Any value given in sexagesimal DMS must be clearly labeled as such.
Sexagesimal DMS must be used <u>*only*</u> for geodetic parameter values in the extended textual location data stanzas. For latitude and longitude coordinates given in General Header records, signed decimal degrees must be used.

D.1.1 Location Data stanza

The Location Data stanza identifies the Coordinate Reference System (CRS) to which data recording locations, such as source and group, and derived locations, for example CDP, are referenced. The CRS serves as a reference for both horizontal positioning, such as easting and northing, and vertical positioning, such as elevations and depths. Without this identification these coordinates are ambiguous. This information may be given implicitly[2] through reference to the EPSG Geodetic Parameter Dataset (Table 1.1) or explicitly[3] (Table 1.2).

Whenever the Coordinate Reference System used is included in the EPSG dataset both the implicit and explicit definition Textual Location Data stanzas may be provided to ease decoding in environments that lack direct internet access to the EPSG dataset. If a user extended version of the EPSG dataset is utilized with a code outside of the EPSG reserved range, then explicit definition is required.

If more than one CRS is used for a given seismic survey, then multiple Location Data Stanzas with unique non-ambiguous IDs must also be populated. Multiple Location Data Stanza IDs may also be used for continuation of stanzas over 3200 bytes to ensure proper continuity and repetition for oversized stanzas.

---

[2] Implicit definition of the Coordinate Reference System (CRS) to which a set of coordinates is referenced requires the user to specify only the appropriate CRS code and the dataset version number of the EPSG dataset from which that CRS definition was obtained. See Table 1.1 for required data for an implicit CRS definition.
[3] Explicit definition of the Coordinate Reference System (CRS) to which a set of coordinates is referenced requires specifying all of the key attributes and parameters necessary to define the CRS. See Table 1.2 for required data in an explicit CRS definition.

Table 1.1 Stanza for implicit identification of Location Data

Stanza Header and Keyword	Format	Resolution / Limits	Comment
((SEG: Location Data EPSG Reference ver 3.0))	Text		Stanza name
Location Data Stanza ID =	Long Integer	ID is user defined;. range 1 to 65535.	Reference from coordinate tuple given in Position block..
CRS code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	The code of the Coordinate Reference System as given in the EPSG Geodetic Parameter Dataset, www.epsg.org
CRS name =	Text	80 character limit	The name of the Coordinate Reference System as given in the EPSG Geodetic Parameter Dataset, www.epsg.org
Dataset version[4] =	Text	80 character limit	The release number for the EPSG Geodetic Parameter Dataset.

[4] For all EPSG dataset versions from EPSG v6.0 onward the code itself (without version number) would be sufficient, as from v6.0 onward code number will always remain unique. However, adding the version number is a good bit of insurance.

Table 1.2 Stanza for explicit definition of Location Data

Stanza Header and Keyword	**Format**	**Resolution / Limits**	**Comment**
((SEG: Location Data ver 3.0))	Text		Stanza name
The following keywords apply to all Coordinate Reference Systems (CRS):			
Location Data Stanza ID =	Long Integer	ID is user defined; range 1 to 65535.	Reference from coordinate tuple given in Position block.
CRS type =	Text from list projected geographic2D geographic3D vertical geocentric compound	24 character limit, but must be from specified "look up" list.	See EPSG dataset www.epsg.org and accompanying guidance notes for information on CRS type.
CRS name =	Text	80 character limit	The name of the Coordinate Reference System.
The following keywords are additionally required if CRS type = compound:			
Horizontal CRS name =	Text	80 character limit	The name of the CRS forming the horizontal component of the compound CRS. The CRS type may be projected or geographic2D.
Vertical CRS name =	Text	80 character limit	The name of the CRS forming the vertical component of the compound CRS. The CRS type will be vertical.
The definitions of these two component CRSs should then be provided through the relevant keywords below.			
The following keywords are additionally required if CRS type = projected or geographic2D, or compound including one of these types, or geographic3D or geocentric:			
Geodetic Datum name =	Text	80 character limit	The name of the Geodetic Datum.
Prime Meridian name =	Text	80 character limit	Mandatory if not "Greenwich". Note: most, but not all, Coordinate Reference Systems use Greenwich as the prime meridian (PM).
PM Greenwich longitude =	Real Number	IEEE double precision normally represented/provided to seven decimal places of a degree. Range $-180 <= \lambda_G <= +180$ degrees or equivalent in other units. See EPSG dataset for examples of values / ranges	The longitude of the CRS's prime meridian relative to the Greenwich meridian, positive if east of Greenwich. Not required if Prime Meridian name = "Greenwich".

Stanza Header and Keyword	Format	Resolution / Limits	Comment
PM Greenwich longitude unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	Not required if Prime Meridian name = “Greenwich”. If Prime Meridian name is not "Greenwich" then at least one of EPSG unit code, unit name or unit conversion ratio to radian is required. Example conversion ratio: if unit = degree, conversion ratio ≈ 0.01745329…
PM Greenwich longitude unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
PM Greenwich longitude unit conversion =	Real Number	IEEE double precision.	
Ellipsoid name =	Text	80 character limit	
Ellipsoid semi-major axis =	Real Number	IEEE double precision, normally given to 10 significant figures. Range 6350 < a < 6400 km or equivalent in other units.	See EPSG dataset for examples of values / ranges.
Semi-major axis unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to meter is required. Example conversion ratio: if unit = US Survey foot, conversion ratio ≈ 0.3048006096…
Semi-major axis unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
Semi-major axis unit conversion =	Real Number	IEEE double precision.	
Ellipsoid inverse flattening =	Real Number	IEEE double precision normally given to 10 significant figures. Range 250 < 1/f < 350. See EPSG dataset for examples of values / ranges	If a sphere, 1/f is infinite. In this case enter value of 0.
The following keyword is additionally required when CRS type = vertical or compound:			
Vertical Datum name =	Text	80 character limit	The name of the Vertical Datum. Not required if ellipsoidal heights are given - these are part of a geographic3D CRS. (Most heights and depths are gravity-related, not ellipsoidal).

Stanza Header and Keyword	Format	Resolution / Limits	Comment
The following keywords are additionally required when CRS type = projected or when a Bin Grid Definition stanza or a Data Geographic Extent stanza or a Coverage Perimeter stanza is included in the extended file header:			
Projection name =	Text	80 character limit	
Projection method name =	Text	50 character limit	For example: “Transverse Mercator”, “Lambert Conic Conformal (1SP)”, “Lambert Conic Conformal (2SP)”.
Projection parameter 1 name =	Text	80 character limit	The number and name of projection defining parameters is dependent upon the map projection method. See the EPSG Geodetic Parameter Dataset coordinate operation method table for parameters required by each projection method.
Projection parameter 1 value =	Real Number	IEEE double precision. See EPSG dataset for examples of values / ranges	
Projection parameter 1 unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = degree, conversion ratio ≈ 0.01745329…; if unit = US Survey foot, conversion ratio ≈ 0.3048006096…; if unit = unity, conversion ratio = 1.0.
Projection parameter 1 unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
Projection parameter 1 unit conversion =	Real Number	IEEE double precision.	
: :			
Projection parameter n name =	Text	80 character limit	
Projection parameter n value =	Real Number	IEEE double precision. See EPSG dataset for examples	

Stanza Header and Keyword	Format	Resolution / Limits	Comment
Projection parameter n unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = grad, conversion ratio ≈ 0.01570796…; if unit = International foot, conversion ratio = 0.3048; if unit = unity, conversion ratio = 1.0.
Projection parameter n unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
Projection parameter n unit conversion =	Real Number	IEEE double precision.	
The following keywords are additionally required if CRS type = projected or geographic2D or geographic3D or geocentric or compound (they are not required if CRS type = vertical):			
Coordinate System axis 1 name =	Text	80 character limit	The name or abbreviation of the Coordinate System (CS) axis for the coordinate in Position block Coord 1. For example: easting, latitude, X, E, or geocentric X.
CS axis 1 orientation =	Text	24 character limit	The positive direction for axis 1. For example: “east”, or “north”.
CS axis 1 unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = grad, conversion ratio ≈ 0.01570796…; if unit = International foot, conversion ratio = 0.3048; if unit = unity, conversion ratio = 1.0.
CS axis 1 unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
CS axis 1 unit conversion =	Real Number	IEEE double precision.	
Coordinate System axis 2 name =	Text	80 character limit	The name or abbreviation of the axis for the coordinate in Position block Coord 2. For example: northing, Y, N, or longitude.
CS axis 2 orientation =	Text	24 character limit	The positive direction for axis 2. For example: “north” or “east”.
CS axis 2 unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = grad, conversion ratio ≈ 0.01570796…; if unit = International foot, conversion

Stanza Header and Keyword	Format	Resolution / Limits	Comment
CS axis 2 unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	ratio = 0.3048; if unit = unity, conversion ratio = 1.0.
CS axis 2 unit conversion =	Real Number	IEEE double precision.	
The following keywords are additionally required when CRS type = geographic3D or geocentric or vertical or compound (they are not required if CRS type = projected or geographic 2D):			
Coordinate System axis 3 name =	Text	80 character limit	The name or abbreviation of the axis for the elevations and depths in Position block Coord3. For example: gravity-related height, ellipsoidal height.
CS axis 3 orientation =	Text	24 character limit	The positive direction for axis 3. For example: “up”.
CS axis 3 unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = grad, conversion ratio ≈ 0.01570796…; if unit = International foot, conversion ratio = 0.3048; if unit = unity, conversion ratio = 1.0.
CS axis 3 unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
CS axis 3 unit conversion =	Real Number	IEEE double precision.	

D.1.2 Example Stanzas for Location Data

a) Example of implicit identification of Location Data through a geographic 2D CRS

((SEG: Location Data EPSG Reference ver 3.0))	
Location Data Stanza ID =	1
CRS code =	4267
CRS name =	NAD27
Dataset version =	6.13

b) Example of implicit identification of Location Data through a projected CRS (i.e. through a Map Grid)

This is the same CRS as described in full in example (e).

((SEG: Location Data EPSG Reference ver 3.0))	
Location Data Stanza ID =	2
CRS code =	34020
CRS name =	NAD27 / Texas South Central
Dataset version =	6.13

c) Example of implicit identification of Location Data through compound coordinate reference system consisting of projected CRS (map grid) with vertical CRS

This is the same CRS as described in full in example (i).

((SEG: Location Data EPSG Reference ver 3.0))	
Location Data Stanza ID =	3
CRS code =	7407
CRS name =	NAD27 / Texas North + NGVD29
Dataset version =	6.13

d) Example of implicit identification of Location Data through a compound coordinate reference system consisting of geographic2D CRS with vertical CRS

((SEG: Location Data EPSG Reference ver 3.0))	
Location Data Stanza ID =	4
CRS code =	7406
CRS name =	NAD27 + NGVD29
Dataset version =	6.13

e) Example of explicit definition of Location Data through a projected CRS (i.e. a map grid)

This is the same CRS as identified implicitly in example (b).

((SEG: LOCATION DATA VER 3.0))	
Location Data Stanza ID	5
CRS type =	projected

CRS name = NAD27 / Texas South Central
Geodetic Datum name = North American Datum 1927
Ellipsoid name = Clarke 1866
Ellipsoid semi-major axis = 6378206.4
Semi-major axis unit name = meter
Ellipsoid inverse flattening = 294.9786982
Projection name = Texas CS27 South Central zone
Projection method name = Lambert Conic Conformal (2SP)
Projection parameter 1 name = latitude of false origin
Projection parameter 1 value = 27.5
Projection parameter 1 unit name = sexagesimal DMS
Projection parameter 2 name = longitude of false origin
Projection parameter 2 value = −99
Projection parameter 2 unit name = degree
Projection parameter 3 name = latitude of first standard parallel
Projection parameter 3 value = 28.23
Projection parameter 3 unit name = sexagesimal DMS
Projection parameter 4 name = latitude of second standard parallel
Projection parameter 4 value = 30.17
Projection parameter 4 unit name = sexagesimal DMS
Projection parameter 5 name = easting at false origin
Projection parameter 5 value = 2000000.0
Projection parameter 5 unit name = US survey foot
Projection parameter 6 name = northing at false origin
Projection parameter 6 value = 0.0
Projection parameter 6 unit name = US survey foot
Coordinate System axis 1 name = X
CS axis 1 orientation = East
CS axis 1 unit code = 9003
CS axis 1 unit name = US Survey foot
CS axis 1 unit Conversion Ratio = 0.304800609601
Coordinate System axis 2 name = Y
CS axis 2 orientation = North
CS axis 2 unit name = US Survey foot

f) Example of explicit definition of Location Data through a geographic2D CRS

((SEG: LOCATION DATA VER 3.0))
Location Data Stanza ID = 6
CRS type = geographic 2D
CRS name = NTF (Paris)
Geodetic Datum name = Nouvelle Triangulation Francaise (Paris)
The following three keyword/value pairs are not needed when Prime Meridian = Greenwich.

See examples (a) and (c) for instances where these keyword/value pairs are omitted.

Prime Meridian name =	Paris
PM Greenwich longitude =	2.5969213
PM Greenwich longitude unit name	Grad
Ellipsoid name =	Clarke 1880 (IGN)
Ellipsoid semi-major axis =	6378249.2
Semi-major axis unit name =	meter
Ellipsoid inverse flattening =	293.466021293
Coordinate System axis 1 name =	latitude
CS axis 1 orientation =	north
CS axis 1 unit name =	grad
Coordinate System axis 2 name =	longitude
CS axis 2 orientation =	east
CS axis 2 unit code =	9105
CS axis 2 unit name =	grad
CS axis 2 unit Conversion Ratio =	0.015707963267949

g) Example of explicit definition of Location Data through a geographic3D CRS

((SEG: Location Data ver 3.0))

Location Data Stanza ID =	7
CRS type =	geographic 3D
CRS name =	WGS 84
Geodetic Datum name =	World Geodetic System 1984
Ellipsoid name =	WGS 84
Ellipsoid semi-major axis =	6378137.0
Semi-major axis unit name =	meter
Ellipsoid inverse flattening =	298.2572236
Coordinate System axis 1 name =	latitude
CS axis 1 orientation =	north
CS axis 1 unit name =	degree
Coordinate System axis 2 name =	longitude
CS axis 2 orientation =	east
CS axis 2 unit name =	degree
Coordinate System axis 3 name =	ellipsoidal height
CS axis 3 orientation =	up
CS axis 3 unit name =	meter

h) Example of explicit definition of Location Data through a geocentric CRS

((SEG: Location Data ver 3.0))

Location Data Stanza ID =	8
CRS type =	geocentric
CRS name =	WGS 84

Geodetic Datum name =	World Geodetic System 1984
Ellipsoid name =	WGS 84
Ellipsoid semi-major axis =	6378137.0
Semi-major axis unit name =	meter
Ellipsoid inverse flattening =	298.2572236
Coordinate System axis 1 name =	geocentric X
CS axis 1 orientation =	from geocenter to intersection of equator and prime meridian
CS axis 1 unit name =	meter
Coordinate System axis 2 name =	geocentric Y
CS axis 2 orientation =	from geocenter to intersection of equator and meridian of 90 degrees E
CS axis 2 unit name =	meter
Coordinate System axis 3 name =	geocentric Z
CS axis 3 orientation =	from geocenter to north pole
CS axis 3 unit name =	meter

i) Example of explicit definition of Location Data through a compound coordinate reference system: projected CRS (map grid) with vertical CRS

This is the same CRS as identified implicitly in example (c).

((SEG: Location Data ver 3.0))

Location Data Stanza ID =	9
CRS type =	compound
CRS name =	NAD27 / Texas South Central + NGVD29
Horizontal CRS name =	NAD27 / Texas South Central
Vertical CRS name =	NGVD29
Geodetic Datum name =	North American Datum 1927
Ellipsoid name =	Clarke 1866
Ellipsoid semi-major axis =	6378206.4
Semi-major axis unit name =	meter
Ellipsoid inverse flattening =	294.9786982
Vertical Datum name =	North American Vertical Datum 1929
Projection name =	Texas CS27 South Central zone
Projection method name =	Lambert Conic Conformal (2SP)
Projection parameter 1 name =	latitude of false origin
Projection parameter 1 value =	27.5
Projection parameter 1 unit name =	sexagesimal DMS
Projection parameter 2 name =	longitude of false origin
Projection parameter 2 value =	−99
Projection parameter 2 unit name =	degree
Projection parameter 3 name =	latitude of first standard parallel
Projection parameter 3 value =	28.23
Projection parameter 3 unit name =	sexagesimal DMS
Projection parameter 4 name =	latitude of second standard parallel

Projection parameter 4 value =	30.17
Projection parameter 4 unit name =	sexagesimal DMS
Projection parameter 5 name =	easting at false origin
Projection parameter 5 value =	2000000
Projection parameter 5 unit name =	US survey foot
Projection parameter 6 name =	northing at false origin
Projection parameter 6 value =	0
Projection parameter 6 unit name =	US survey foot
Coordinate System axis 1 name =	X
CS axis 1 orientation =	east
CS axis 1 unit name =	US Survey foot
Coordinate System axis 2 name =	Y
CS axis 2 orientation =	north
CS axis 2 unit name =	US Survey foot
Coordinate System axis 3 name =	height
CS axis 3 orientation =	up
CS axis 3 unit name =	foot

j) Example of implicit definition of Location Data through redundant Geographic and Projected CRSs

In all of the examples (a) through (i) above, one CRS identification is given. This is sufficient when one coordinate tuple is given in the header block. Where two coordinate tuples are given in the header block two CRS definitions are required. In the header blocks, coordinate tuples 1 and 2 will reference these two CRSs. In this example two CRSs are identified implicitly, the first for a geographical (latitude/longitude) coordinate tuple and the second for a projected (map grid) coordinate tuple.

((SEG: Location Data EPSG Reference ver 3.0))	
Location Data Stanza ID =	1
CRS code =	4267
CRS name =	NAD27
Dataset version =	6.13
((SEG: Location Data EPSG Reference ver 3.0))	
Location Data Stanza ID =	2
CRS code =	34020
CRS name =	NAD27 / Texas South Central
Dataset version =	6.13
((SEG: EndText))	

D.1.3 Location Data Coordinate Transformation stanzas

If during acquisition a coordinate transformation has been applied to derive the coordinates described through a given Location Data stanza or a given Location Data EPSG Reference stanza (for example when the location data has been transformed from the GPS system's WGS 84 coordinates to a local Coordinate Reference System), details of the coordinate transformation which has been applied should be included as a Location Data Coordinate Transformation. A coordinate transformation may be either "single", i.e. directly from source CRS *A* to target CRS *B*, or a "concatenated operation", that is indirectly from *A* to *B* via one or more intermediate CRSs, for example from CRS *A* to CRS *C* to CRS *B*. In either case the final target CRS, i.e. CRS *B*, is that defined in the Location Data or Location Data EPSG Reference stanza.

As with location CRS data the coordinate transformation may be given implicitly[5] through reference to the EPSG Geodetic Parameter Dataset (Table 1.3) or described explicitly[6] (Table 1.4).

If more than one CRS is used for a given survey, then an appropriate Location Data Coordinate Transformation Stanza with a unique ID must be given for each of those CRSs. The Location Data Stanza ID for that unique CRS should be referenced in the Location Data Coordinate Transformation Stanza (as shown below).

Multiple Location Data Transformation Stanza IDs may also be used for continuation of stanzas over 3200 bytes to ensure proper continuity and repetition for oversized stanzas.

[5] Implicit definition of a transformation applied to a given set of coordinates requires the user to specify only the appropriate transformation code and dataset version number of the EPSG dataset from which that transformation definition was obtained. See Table 1.3 for required data for an implicit transformation definition.

[6] Explicit definition of a transformation applied to a given set of coordinates requires specifying all of the key attributes and parameters necessary to define that transformation. See Table 1.4 for required data in an explicit transformation definition.

Table 1.3 Stanza for implicit identification of Location Data Transformation

Stanza Header and Keyword	Format	Resolution / Limits	Comment
((SEG: Location Data Coordinate Transformation EPSG Reference ver 3.0))	Text		Stanza name
Location Data Coordinate Transformation Stanza ID =	Long Integer	User-defined.	
Location Data Stanza ID =	Long Integer	See tables 1.1 and 1.2.	The Location Data Stanza ID for the CRS that is the target of this transformation.
Transformation code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	The code of Transformation, as given in the EPSG Geodetic Parameter Dataset, www.epsg.org
Transformation name =	Text	80 character limit	The name of the Transformation, as given in the EPSG Geodetic Parameter Dataset, www.epsg.org
Dataset version[7] =	Text	80 character limit	The release number for the EPSG Geodetic Parameter Dataset.

[7] For all EPSG versions from EPSG v6.0 onward the code itself (without version number) would be sufficient, as code number from v6.0 onward will always remain unique. However, adding the version number is a good bit of insurance.

Table 1.4 Stanza for explicit definition of Location Data Transformation

Stanza Header and Keyword	Format	Resolution / Limits	Comment
((SEG: Location Data Coordinate Transformation ver 3.0))	Text		Stanza name
Location Data Coordinate Transformation Stanza ID =	Long Integer	User-defined.	Normally all seismic work will be done with a single transformation and thus the Stanza ID for this specific transformation would be the only one populated.
Location Data Stanza ID =	Long Integer	See tables 1.1 and 1.2.	The Location Data Stanza ID for the CRS that is the target of this transformation.
Transformation type =	From enumerated list: transformation concatenated operation	24 character limit, but must be from enumerated list.	Transformation = a single operation that has been applied to initial coordinates to derive values referred to the CRS identified in the Location Data stanza. Concatenated operation = a set of multiple transformations that have been applied sequentially.
Transformation name =	Text	80 character limit	The name of the Transformation
Source CRS name =	Text	80 character limit	The name of the CRS from which coordinates have been transformed, for example that used within the navigation system (usually "WGS 84").
Target CRS name =	Text	80 character limit	The name of the CRS to which location data is referred. A Location Data stanza or a Location Data EPSG Reference stanza containing this name must precede this stanza.
Transformation version =	Text	24 character limit	The version of the transformation between the source and target CRSs.
The following keywords are additionally required when transformation type = transformation:			
Transformation method name =	Text	50 character limit	For example "Geocentric translations", "Position Vector 7-param. Transformation", " Coordinate Frame rotation", "NADCON", "NTv2".
Then either (a) the following keyword is additionally required for transformation methods which use grid files:			
Transformation parameter file name =	1 or more comma-separated text strings	254 character limit	Containing as many file names as the method requires. For example for the NTv2 method one file name is required, for the NADCON method two file names are required.
Or (b) the following keywords are additionally required for transformation methods other than those which use grid files:			

Stanza Header and Keyword	Format	Resolution / Limits	Comment
Transformation parameter 1 name =	Text	80 character limit	The number and name of parameters is dependent upon the transformation method. For example the Position Vector and Coordinate Frame methods the seven parameters required are: • X-axis translation • Y-axis translation • Z-axis translation • X-axis rotation • Y-axis rotation • Z-axis rotation • Scale difference See example (a) in A-1.4 below for the three parameters required when the transformation method is Geocentric Translations. See EPSG dataset transformation method table for parameters required for other methods.
Transformation parameter 1 value =	Real Number	IEEE double precision.	See EPSG dataset for examples of values / ranges.
Transformation parameter 1 unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = degree, conversion ratio ≈ 0.01745329…; if unit = US Survey foot, conversion ratio ≈ 0.3048006096…; if unit = parts per million, conversion ratio = 0.000001.
Transformation parameter 1 unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
Transformation parameter 1 unit conversion ratio =	Real Number	IEEE double precision.	
: :			Repeat above sequence for each transformation parameter
Transformation parameter n name =	Text	80 character limit	
Transformation parameter n value =	Real Number	IEEE double precision.	See EPSG dataset for examples of values / ranges.
Transformation parameter n unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = grad, conversion ratio ≈ 0.01570796…; if unit = International foot, conversion ratio =

Stanza Header and Keyword	Format	Resolution / Limits	Comment
Transformation parameter n unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	0.3048; if unit = parts per million, conversion ratio = 0.000001.
Transformation parameter n unit conversion ratio =	Real Number	IEEE double precision.	
The following keywords are additionally required when transformation type = concatenated operation. They should be repeated in a block for every step with the step counter m being incremented:			
Concatenated transformation step =	Integer	Integer (typically between 1 and 4)	The value *m* is used in the following keywords.
Step *m* source CRS name =	Text	80 character limit	The name of the CRS used within the navigation system (usually "WGS 84").
Step *m* target CRS name =	Text	80 character limit	The name of the CRS to which location data is referred. A Location Data stanza or a Location Data EPSG Reference stanza containing this name must precede this stanza.
Step *m* transformation version =	Text	24 character limit	The version of the transformation between the source and target CRSs.
Step *m* transformation method name =	Text	50 character limit	For example "Geocentric translations", "Position Vector 7-param. Transformation", " Coordinate Frame rotation", "NADCON", "NTv2".
Then either (a) the following keyword is additionally required for steps using transformation methods which use grid files:			
Step *m* transformation parameter file name 1 =	Text	254 character limit	Containing primary file name as the method requires.
. . . . . .			Repeat above sequence for each transformation parameter
Step *m* transformation parameter file name *n* =	Text	254 character limit	Containing "nth" file name required by specific method. For the NTv2 method one file name is required, for the NADCON method two file names are required.
Or (b) the following keywords are additionally required for steps using transformation methods other than those which use grid files:			

Stanza Header and Keyword	Format	Resolution / Limits	Comment
Step *m* transformation parameter 1 name =	Text	80 character limit	The number and name of parameters is dependent upon the transformation method. For example the Position Vector and Coordinate Frame methods the seven parameters required are: • X-axis translation • Y-axis translation • Z-axis translation • X-axis rotation • Y-axis rotation • Z-axis rotation • Scale difference See example (a) in A-1.4 below for the three parameters required when the transformation method is Geocentric Translations. See EPSG dataset transformation method table for parameters required for other methods.
Step *m* transformation parameter 1 value =	Real Number	IEEE double precision.	See EPSG dataset for examples of values / ranges.
Step *m* transformation parameter 1 unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, and unity for scale) is required. Example conversion ratios: if unit = degree, conversion ratio ≈ 0.01745329…; if unit = US Survey foot, conversion ratio ≈ 0.3048006096…; if unit = parts per million, conversion ratio = 0.000001.
Step *m* transformation parameter 1 unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
Step *m* transformation parameter 1 unit conversion ratio =	Real Number	IEEE double precision.	
: :			Repeat above sequence for each transformation parameter
Step *m* transformation parameter n name =	Text	80 character limit	
Step *m* transformation parameter n value =	Real Number	IEEE double precision.	See EPSG dataset for examples of values / ranges

Stanza Header and Keyword	Format	Resolution / Limits	Comment
Step *m* transformation parameter n unit code =	Long integer	EPSG code range is limited to 1–32767, but other database users may add their own code extensions outside of this range.	At least one of EPSG unit code, unit name or unit conversion ratio to standard unit (radian for angle, meter for length, unity for scale) is required. Example conversion ratios: if unit = grad, conversion ratio ≈ 0.01570796…; if unit = International foot, conversion ratio = 0.3048; if unit = parts per million, conversion ratio = 0.000001.
Step *m* transformation parameter n unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
Step *m* transformation parameter n unit conversion ratio =	Real Number	IEEE double precision.	

D.1.4 Example Stanzas for Location Data Transformation

k) Example of implicit identification of Location Data Coordinate Transformation through a single transformation

((SEG: Location Data Coordinate Transformation EPSG Reference ver 3.0))	
Location Data Coordinate Transformation Stanza ID =	10
Location Data Stanza ID =	1
Transformation code =	15853
Transformation name =	NAD27 to WGS 84 (81)
Dataset version =	6.13

l) Example of implicit identification of Location Data Coordinate Transformation through a concatenated operation

((SEG: Location Data Coordinate Transformation EPSG Reference ver 3.0))	
Location Data Coordinate Transformation Stanza ID =	11
Location Data Stanza ID =	1
Transformation code =	8635
Transformation name =	NAD27 to NAD83(CSRS) (3)
Dataset version =	6.13

m) Example of explicit definition of Location Data Transformation through a single transformation with a method using parameters

((SEG: Location Data Coordinate Transformation ver 3.0))	
Location Data Coordinate Transformation Stanza ID =	12
Location Data Stanza ID =	1
Transformation type =	transformation
Transformation name =	WGS 84 to NAD27
Source CRS name =	WGS 84
Target CRS name =	NAD27
Transformation version =	JECA-Usa GoM C
Transformation method name =	Geocentric translations
Transformation parameter 1 name =	X-axis translation
Transformation parameter 1 value =	7
Transformation parameter 1 unit name =	meter
Transformation parameter 2 name =	Y-axis translation
Transformation parameter 2 value =	−151
Transformation parameter 2 unit name =	meter
Transformation parameter 3 name =	Z-axis translation
Transformation parameter 3 value =	−175
Transformation parameter 3 unit name =	meter

n) Example of explicit definition of Location Data Transformation through a single transformation with a method using grid files

((SEG: Location Data Coordinate Transformation ver 3.0))	
Location Data Coordinate Transformation Stanza ID =	13
Location Data Stanza ID =	18
Transformation type =	transformation
Transformation name =	NAD27 to NAD83 (1)
Source CRS name =	NAD27
Target CRS name =	NAD83
Transformation version =	NGS-Usa Conus
Transformation method name =	NADCON
Transformation parameter file name 1 =	conus.las,
Transformation parameter file name 2=	conus.los

o) Example of explicit definition of Location Data Transformation through a concatenated operation

((SEG: Location Data Coordinate Transformation ver 3.0))	
Location Data Coordinate Transformation Stanza ID =	14
Location Data Stanza ID =	1
Transformation type =	Concatenated operation
Transformation name =	NAD83(CSRS) to NAD27
Source CRS name =	NAD83(CSRS
Target CRS name =	NAD27
Transformation version =	EPSG-Can AB
Concatenated transformation step =	1
Step 1 transformation step name =	NAD83(CSRS) to NAD83
Step 1 source CRS =	NAD83(CSRS)
Step 1 target CRS =	NAD83
Step 1 transformation version =	AB Env-Can AB
Step 1 transformation method name =	NTv2
Step 1 transformation parameter file name =	AB_CSRS.gsb
Concatenated transformation step =	2
Step 2 transformation step name =	NAD83 to NAD27
Step 2 source CRS =	NAD83
Step 2 target CRS =	NAD27
Step 2 transformation version =	GC-Can NT2
Step 2 transformation method name =	NTv2
Step 2 transformation parameter file name =	NTV2_0.GSB

Appendix E: Examples and Calculations

E.1 Samples per scan type

$$S/S = \sum_{1}^{cs} C/S \times 2^{s/c}$$

where

S/S = samples per scan type

C/S = channels in this channel set (channel set descriptor Bytes 9 and 10)

$2^{s/c}$ = samples per channel (in this channel set) (channel set descriptor Byte 12)

CS = number of channel sets in this scan type (general header Byte 29)

For example, for a 2-msec base scan interval with 4 auxiliary channels at 2 msec, 96 channels at 2 msec and 12 channels at ½ msec. There are three channel sets, so CS = 3.

$$S/S = C/S \times 2^{s/c} \Big|_{cs=1} + C/S \times 2^{s/c} \Big|_{cs=2} + \ldots$$

$$S/S = 4 \times 1 + 96 \times 1 + 12 \times 4$$

$$S/S = 4 + 96 + 48 = 148$$

Note that all scan types must have the same number of data samples.

E.2 Skew fields per scan type

$SK = \dfrac{S/S}{32}$ (If the quotient is not a whole number, round up to the next largest whole number)

where

SK = skew fields (of 32 bytes each) per scan type (general header Byte 30)

S/S = samples per scan (Appendix E1)

Substituting for S/S from Appendix E.1:

$$SK = \frac{1}{32} \sum_{1}^{CS} C/S \times 2^{s/c}$$

(If the quotient is not a whole number, round up to the next largest whole number.)

where

CS = the number of channel sets in each scan type (general header Byte 29)

C/S = channels in this channel set (channel set descriptor Bytes 9 and 10)

$2^{s/c}$ = samples per channel in this channel set (channel set descriptor Byte 12).

For example, for a 2-msec base scan with 4 auxiliary channels at 2 msec, 96 channels at 2 msec and 12 channels at ½ msec

$$SK = \frac{4 \times 1 + 96 \times 1 + 12 \times 4}{32}$$

$$= \frac{148}{32} = 4\,{}^{20}\!/\!_{32} \text{ roundup} = 5 \text{ fields of 32 bytes each}$$

E.3 Filter slope calculation

Modern filters may not have a constant slope, so it is necessary to define this parameter. The slope is defined as the asymptote of effective performance as it would be in a constant slope filter. This slope is zero dB attenuation at the cut-off frequency and a specific attenuation at the beginning of the stop band. The chosen values are 40 dB for a low-cut filter and 60 dB for an anti-alias filter.

Low-cut filter slope calculation.

$$LS = \frac{40}{\log_2 f_{LCO}/f_{40}} = \frac{40}{3.322 \log_{10} f_{LCO}/f_{40}} = \frac{12.04}{\log_{10} f_{LCO}/f_{40}}$$

LS = low-cut filter slope (channel set descriptor Bytes 19 and 20),
f_{40} = the frequency of 40 dB low-cut filter attenuation,
f_{LCO} = low-cut filter cut-off frequency usually 6 or 12 dB attenuation.

Alias-filter slope calculation.

$$AS = \frac{60}{\log_2 f_{60}/f_{ACO}} = \frac{60}{3.322 \log_{10} f_{60}/f_{ACO}} = \frac{18.06}{\log_{10} f_{60}/f_{ACO}}$$

AS = alias filter slope (channel set descriptor Bytes 15 and 16)
f_{60} = the frequency of 60 dB alias-filter attenuation
f_{ACO} = alias-filter cut-off frequency usually 3 or 6 dB attenuation

E.4 SEG-D file storage on non-tape media

Modern acquisition and processing systems commonly require storing SEG-D data on disk. These are a few examples of how this can be done:

Raw disk copy of SEG-D records

As stated in chapter 3, starting with Revision 2, SEG-D data are treated as a stream of bytes. When storing Rev 3.0 data on tape, tape block and file marks are inserted at points defined by the manufacturer. Interpretation of data is done on the raw byte data stream. It is important to maintain the byte stream view when storing SEG-D on disk as well, such that decoding and encoding of data is performed in the same way, independent of its storage format. (For fixed blocksize devices, trailing padding at the end of some blocks slightly violates this treatment and we strongly recommend that such data should be first converted, as detailed below, from FIXREC to RECORD format before storing on disk or transmitting through a network.)

From the definitions in chapter 3 it follows that a disk file is a storage media similar to a tape, hence a tape label must be placed at the beginning of the file, followed by a set of SEG-D records. A TOC indexing file may or may not be a part of the file.

A good solution is to use the filename as indication of the contents of the SEG-D records in the file. To speed up access to SEG-D data on disk, storing only one SEG-D record per file may be the best option with, for example, the file name being *<filenumber>.segd* or *<linename>_<filenumber>.segd*.

The fields in the Tape Label should be set thusly:

- Storage unit structure will be RECORD
- Maximum block size will be 1
- Serial Number must be unique, however as disk storage is normally temporary, making it unique within the disk might be sufficient. For single record files, using a combination of file number and e.g. line/sequence/swath number might be a good solution. (For temporary storage, if the Serial Number is not used as part of an indexing system, the producer has flexibility in defining its contents.)

The file layout:

|Tape Label|SEG-D Header|Trace Data|General Trailer|EOF|

The advantage of this solution is it allows direct access of data on disk (through a memory mapped file) in the same way as data are accessed in memory. For certain tasks, especially quality control inspection, this may speed up the process with a factor of 10 or 100.

E.5 SEG-D record index interpretation for marine, land, seabed, transition-zone, and VSP surveys

SEG-D uses six indices to uniquely assign a trace with a unique logical position within the record. The interpretation of indices depends on the type of survey. The basic difference is between surveys where the receivers are stationary (land/seabed), and where the receivers are moving (towed marine). Transition zone surveys use a combination of the two. Other conceivable combinations are seabed and towed marine, VSP (vertical seismic profile, i.e. receivers in one or more boreholes) and land/marine/seabed/TZ. The indices need to be valid for all these survey types, including combinations.

The indices are:

- Line
- Point
- Point Index

- Group Index
- Depth Index
- Reshoot Index

Indices exists for both source (i.e. Source Point, Source Line), and receiver (i.e. Receiver Point, Receiver Line).

The *Group Index* is used to indicate a trace belonging to a set of traces that need to be processed as a unit (and are placed at the same position), e.g. three-component data that need to be rotated together.

The *Reshoot Index* is used to indicate a trace has been recorded earlier, and the data with the highest Reshoot Index should be used in processing (normally).

Towed marine survey

A Marine survey is normally divided up in lines, and the survey is acquired by one or more vessels towing one or more sources and one or more streamers. The lines are normally assigned an integer number. The example below shows a complex survey using two streamer vessels with two sources each, and one source vessel travelling behind one of the streamer vessels.

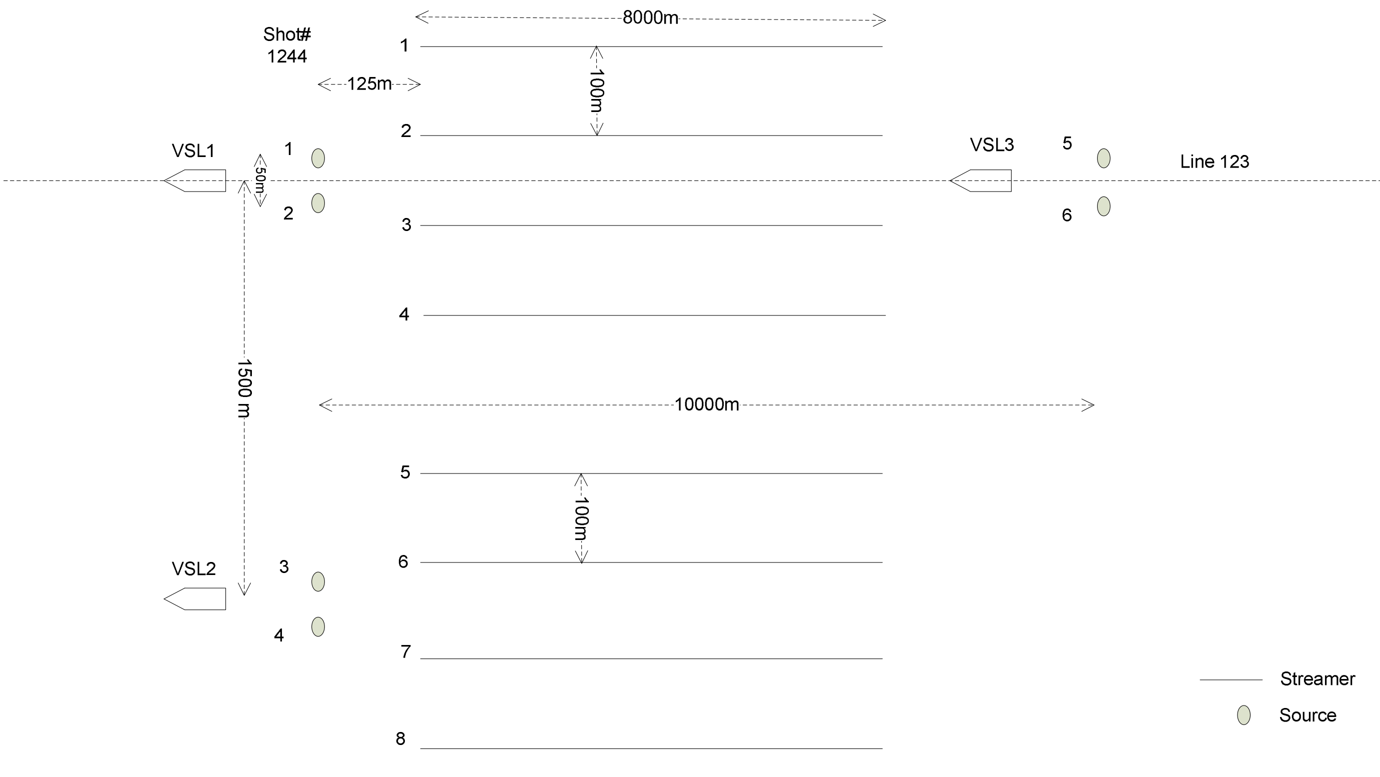


Figure showing a complex vessel configuration for a marine seismic survey. Note, figure is not to scale

Source / receiver line number

The receiver and source line number is give by the line number, in this case 123. In this example, the fractional part of the source /receiver number will be used to indicate the nominal position of the equipment.

Using a pure logical numbering scheme is allowed, i.e. there are 12 separate equipment lines in the figure (8 streamer and 4 source). Equipment is e.g. allocated line number in steps of 0.5, starting counting from 123.0. Streamer 1 has receiver line 123.0, Streamer 2 123.05, Source 1 and 5 has source line 123.1, Source 2 and 6 123.15, ... Streamer 8 has 123.55.

However for this example, the logical positions are selected to reflect the actual offsets. The offset between the outer streamers (1 and 8) are 1300 meters. 1 meter is selected to be 0.0005 logical units. Assuming increasing line numbers downwards in the figure, Streamers are assigned the following line numbers:
1: 122.925, 2: 122.955, 3: 123.025, 4: 123.075, 5: 123.675, 6: 123.725, 7: 123.775, 8: 123.725.
Sources are assigned the following source line numbers:
1: 122.9875, 2: 123.0125, 4: 123.4875, 5: 123.5125, 5: 122.9875, 6: 123.0125
It is worth noting here that the number resolution in the receiver/source line numbers is not very high, so the numbers above will be approximate.

<u>Source / receiver point number</u>
A similar strategy is used for source/receiver point number. The distance between sources 1 and 5 is 10000 meters. Receivers in the streamers are spaced at 12.5 meter intervals. 12.5 meters is chosen to be 1/1024 logical point units (The reason behind selecting 1/1024 as a unit is that this number is well represented in the format). The marine acquisition line is divided up into shots (source events), with 25 meter spacing, the shot# in the figure is 1244. The numbering starts from the first equipment (sources 1-4), counting backwards until the last equipment (sources 5-6). The logical numbering will be the same independent of the line direction (upline, i.e. shot# increasing, or downline, i.e shot# decreasing).
Sources 1-4 get Source Point Number 1244.0, sources 5-6 1244.78125.
Receiver 1 (on all streamers) is assigned Receiver Point Number 1244.00977, Receiver 2 1244.01074, Receiver 3 1244.01172, ... Receiver 640 1244.63379

The Receiver Point Index will always be 1 for the marine survey.

Note: The source/receiver line/point number will only reflect the nominal position, actual position due to e.g. streamer feather is not considered when determining the logical position determined by the indices. If actual position is needed, the position may be stored in Position Blocks in Trace Headers.

Land

Land seismic surveys have been using Source and Receiver lines for many years, so no large example will be give here. SEG-D may be used to map in the numbers currently used.

Seabed

Seabed surveys are a combination of a Land and Marine survey, where sensors are placed on the sea floor, and a source vessel is moving around in the area, usually along Source Lines. Seabed covers several operations, like seismic sensors placed on the sea floor temporarily in the form of streamers or separate sensor units, permanently around installations, or electrical / magnetic sensors (EM survey).

The sensors are deployed in a grid defined by the Receiver Line and Point Numbers, similar to a Land survey.

Sources are moving in Source Lines similar to the example for Marine survey, and are assigned indices accordingly. For continuous source efforts, like EM surveys, where the source fires continually, it is allowed to set the Source Point Number to 0 if it does not fit at all. However, it is highly recommended to determine a logical numbering scheme that allows the use of Source Point Number. This is one of the reasons why splitting up the data into smaller records are recommended, e.g. to match the cycle of the source (see E.10 Electromagnetic (EM) survey) for more details.

Transition-zone

Transition zone may be viewed as a combination survey being parts Marine and parts Land survey. Indices are assigned using Marine or Land methods where appropriate.

Downhole seismic / Vertical seismic profile

This covers all survey types where seismic sensors are inserted down a well. The sensors may be a dedicated seismic string, or acquired as part of other downhole equipment. The survey may be a normal seismic survey, Land or Marine, where the downhole data is acquired in conjunction with a larger system, or a dedicated well centric survey, like a walkaway vertical seismic profile. The Source Line /Point Number is assigned as normal, even with well centric surveys, the source moves in predetermined line.

Receiver Line / Point Number is assigned by giving the all receivers in the well the same Receiver Line and Receiver Point Number, and use the Depth Index to indicate the order downhole. If the downhole string contains more than 255 receivers, the Receiver Point Index is used to extend the numerical range of the Depth Index. This allows a maximum of about 65000 receivers down a well. It is worth noting that all receivers in a well are assigned the same Receiver Line /Point Number, even if the well is horizontal.

E.6 Trace edit example

Example of Trace Edits in General Trailer, built up by data from several systems, here by the acquisition system ACQSYS, the quality control system AQC, and the positioning system NAVPOS. The edit block may be extended by the different systems, or separate edit blocks may be created.

```
V SEGD Trace Edit v1.0
S 12:34:55 12-JAN-2007
W ACQSYS v2.2.3 build 231, M/S Unsinkable II
C These are the edits performed by online acquisition
H Killed traces
A killed trace
X 1;2;1-4
X 1;5;23,48-50,423
X 1;8;345
X 1;5-14;1
E END Killed traces
H Weak trace detection
```

```
A Seismic trace RMS < 1 uB
P SEIS_RMS(1,0.014)
R 1;5-14;1-558
X 1;3;123-135
X 1;15;33,55,231-234
E END Weak trace detection
S 14:15:00 12-JAN-2007
W AQC v6.4, operator John D. Operator
C These edits done by the online Quality Control system
H Noisy trace detection
A NOISY_TRACE(12,0.65)
C Noisy trace detection by comparing neighboring traces
C and performing frequency content analysis
R 1;5-14;1-558
X 1;6;1-15,234-238
X 1;8;1-15
E END Noisy trace detection
H Manual edits
A Manual edits
C Manual edits performed by operator
Y EXCLUDE
X 1;9;550-558
E END Manual edits
S 15:05:00 12-JAN-2007
W NAVPOS v1.2
C Edits done by positioning system
H Position verification
A GPS_VERIFY(4) GPS position quality verification
Y WARNING
X 1;14;1-558
C Comments entered by navigator:
C GPS positions for streamer 10 unreliable for shots 1201-1243
E END Position verification
T END SEGD Trace Edit
```

E.7 Source information in General Header or Trace Header

<u>Source IDs/sets</u>
Each source needed to be recorded must be assigned an ID, the same is required of any combined set of sources.

Examples:
1. A SEG-D record contains four vibrators firing as a group. The record contains Source Description blocks for each of the single vibrators, and the combined source group. The four vibrators have ids 2,12,13, and 16, each belonging to source set 101 (i.e. vibrator group 101). The combined source is assigned the id 101, and it does not belong to any source set, i.e. the source set is set to 0.

Source id	*Source set*

2	101
12	101
13	101
16	101
101	0

2. A marine airgun source is made up of three sub-strings, each consisting of three airguns (one cluster, and two single guns). The record contains Source Description blocks for each of the airguns, one for each of the substrings, and one for the combined source. The single/cluster guns have ids 1,2,3 (substring 1), 4,5,6 (substring 2), 7,8,9 (substring 3). Numbering should be starboard to port, front to back. The substrings are assigned the ids 51,52,53, and the combined source id 101. The combined source have set id 101 (starboard source is 101, the port source has set id 102, source id 102). The substrings have set ids 51,52,53.

Source id	*Source set*
1	51
2	51
3	51
4	52
5	52
6	52
7	53
8	53
9	53
51	101
52	101
53	101
101	0

To be able to see which sources belongs to which source sets, it is important to give the source set, and the combined source ids the same value. This means SEG-D rev 3.0 supports up to 255 different single sources and source sets/groups.

Source information should match SPS definitions.

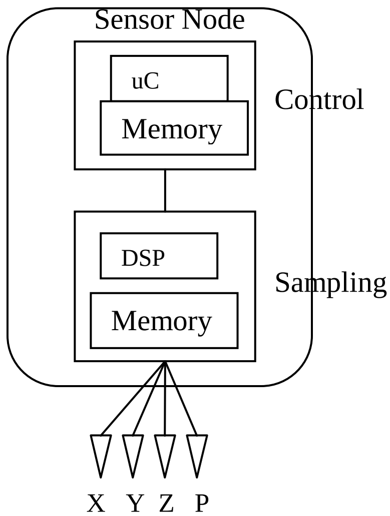
To be able to cope with multiple source events during the record, Source Information blocks can be recorded as part of the Trace Extension Header (instead of in the General Header). The Additional Source Info block *must* then be present for each Source Information block, and the timestamp (bytes 1–8) filled in properly to indicate which sample the source event happened. The relevant source description blocks must also be inserted into all traces relating to source (source signatures, baseplate measurements, raw nearfield hydrophone data etc.), though they may also be recorded as part of any other (seismic/auxiliary) trace.

E.8 Extended Recording Mode example

Extended recording mode is mostly designed for situations where lack of resources makes creation of multiple SEGD records a complicated or impossible task, e.g. in embedded systems where memory resources or memory model making storing meta-information intermixed with the data difficult.

Extended recording mode allows data to be recorded in blocks of up to 2147 seconds, containing a small trace header (84 byte minimum) followed by up to 2147 seconds of data.

Consider the following sensor node:



The node contains one microcontroller (μC) with attached memory for control, monitoring, quality control and external communication, and one digital signal processor (DSP) with attached memory for sampling and data filtering from 4 sensors (a 3 component accelerometer and one hydrophone).

The node is deployed on the sea floor, and will remain there for a maximum of two weeks (1,209,600 seconds) before being picked up. Extended recording mode allows up to 65535 channelsets (or time blocks) to be acquired in one record, each channelset being up to 4294 seconds long. As the node has 4 different channel types, which have to be stored in separate channelsets, the available number of time blocks in extended Recording mode is 65535/4=16383. To get the minimum time block/channelset length for this system we calculate 1209600/16383 = 73.8 seconds (with 4 channels, the maximum record length will be 16383 *4294 equals ca. 814 days).

One may then safely decide to choose a trace size of 120 seconds. This creates a suitable sized trace which fit the memory model of the DSP well, has a good safety margin (can be deployed for a maximum of 22 days), and has a good overhead to data ratio. In addition the system is created such that in case of a memory problem, only 120 consecutive seconds of data will be lost at a time.

The microcontroller is responsible for creating the SEG-D record header and keep it in memory during the acquisition. Whenever a new channelset is needed, the DSP notifies the microcontroller, which creates the 4 channelset descriptors. The DSP creates 4 new trace headers in its memory, each consisting of a Trace Header

(20 bytes), Trace Header Extension (32 bytes), and a Timestamp Header (32 byte). It then starts writing trace data into the new traces.

The microcontroller monitors several other sensors during the acquisition, including temperature, box pressure, leakage sensor, power status, compass readings, tilt/orientation measurements, clock drift etc. This information is stored in the SEG-D record Extended Header in a manufacturer specific format.

When the data is downloaded from the node, the microcontroller first sends the header data stored in its memory, then reads the trace data from the DSP such that the download software receives a complete SEG-D record.

E.9 SEG-D timestamp calculation

```
/* SEG-D UTC <--> GPS conversion

 * The following is ANSI C++ code.
 * It should be easily translated to most other common languages like Java, C, C#, Fortran, Perl
 */

#include <stdio.h>
#include <limits.h>
#include <stdlib.h>

#if LONG_MAX <= 2147483647L
#if defined(_MSC_VER)
typedef __int64 Timestamp;
#define TSDECL(x) x##i64
#define tsfmt "%I64d"
#else
typedef long long Timestamp;
#define TSDECL(x) x##LL
#define tsfmt "%lld"
#endif
#else
typedef long Timestamp;
#define TSDECL(x) x##L
#define tsfmt "%ld"
#endif

const int dayArr[2][12] = { { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 },
                            { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 } };

const char *monthArr[12] = { "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                         "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };
class Utc {
   public:
        int year;
        int month; /* 1..12 */
        int day; /* 1..31 */
        int hour; /* 0..23 */
        int minute; /* 0..59 */
        int sec; /* 0..60 60=leap second */
        int microsec; /* 0-999999 fraction of sec */

        Utc( int y, int m, int d, int h, int min, int s, int us ) :
        year(y), month(m), day(d), hour(h), minute(min), sec(s), microsec(us) {}

        Utc() {}

        const char* toString() const {
               /* not threadsafe code */
               static char str[28];
               sprintf( str, "%2d %s %4d %02d:%02d:%02d.%06d",
```

```
                              day,monthArr[month-1],year,hour,minute,sec,microsec);
               return str;
       }
};

struct DayLeapsec {
       int days; /* days since 6 jan 1980, may be negative */
       int leapsecs;
};

struct TimestampLeapsec {
       Timestamp timestamp;
       int leapsecs;
};

struct TimestampJan1Year {
       int year;
       Timestamp timestamp;
};

#define  NLEAP  (sizeof(dayLeapArr)/sizeof(dayLeapArr[0]))

const DayLeapsec dayLeapArr[] = {
       {-3657, -9},    /* 1 jan 1970 */
       {-2745, -8},    /* 1 jul 1972 */
       {-2561, -7},    /* 1 jan 1973 */
       {-2196, -6},    /* 1 jan 1974 */
       {-1831, -5},    /* 1 jan 1975 */
       {-1466, -4},    /* 1 jan 1976 */
       {-1100, -3},    /* 1 jan 1977 */
       { -735, -2},    /* 1 jan 1978 */
       { -370, -1},    /* 1 jan 1979 */
       {   -5, 0},     /* 1 jan 1980 */
       { 542,  1},     /* 1 jul 1981 */
       { 907,  2},     /* 1 jul 1982 */
       {1272,  3},     /* 1 jul 1983 */
       {2003,  4},     /* 1 jul 1985 */
       {2917,  5},     /* 1 jan 1988 */
       {3648,  6},     /* 1 jan 1990 */
       {4013,  7},     /* 1 jan 1991 */
       {4560,  8},     /* 1 jul 1992 */
       {4925,  9},     /* 1 jul 1993 */
       {5290,  10},    /* 1 jul 1994 */
       {5839,  11},    /* 1 jan 1996 */
       {6386,  12},    /* 1 jul 1997 */
       {6935,  13},    /* 1 jan 1999 */
       {9492,  14},    /* 1 jan 2006 */
       {10588, 15}     /* 1 jan 2009 */
};

const TimestampLeapsec tsLeapArr[] = {
       {TSDECL(-315964809000000) , -9},
       {TSDECL(-237168008000000) , -8},
       {TSDECL(-221270407000000) , -7},
       {TSDECL(-189734406000000) , -6},
       {TSDECL(-158198405000000) , -5},
       {TSDECL(-126662404000000) , -4},
       {TSDECL( -95040003000000) , -3},
       {TSDECL( -63504002000000) , -2},
       {TSDECL( -31968001000000) , -1},
       {TSDECL(    -432000000000) ,  0},
       {TSDECL(  46828801000000) ,  1},
       {TSDECL(  78364802000000) ,  2},
       {TSDECL( 109900803000000) ,  3},
       {TSDECL( 173059204000000) ,  4},
       {TSDECL( 252028805000000) ,  5},
       {TSDECL( 315187206000000) ,  6},
       {TSDECL( 346723207000000) ,  7},
       {TSDECL( 393984008000000) ,  8},
       {TSDECL( 425520009000000) ,  9},
       {TSDECL( 457056010000000) , 10},
       {TSDECL( 504489611000000) , 11},
       {TSDECL( 551750412000000) , 12},
```

```
        {TSDECL( 599184013000000) , 13},
        {TSDECL( 820108814000000) , 14},
        {TSDECL( 914803215000000) , 15}
};

const TimestampJan1Year tsJan1YearArr[] = {
       { 1970, TSDECL(-315964809000000) },
       { 1971, TSDECL(-284428809000000) },
       { 1972, TSDECL(-252892809000000) },
       { 1973, TSDECL(-221270407000000) },
       { 1974, TSDECL(-189734406000000) },
       { 1975, TSDECL(-158198405000000) },
       { 1976, TSDECL(-126662404000000) },
       { 1977, TSDECL(-95040003000000) },
       { 1978, TSDECL(-63504002000000) },
       { 1979, TSDECL(-31968001000000) },
       { 1980, TSDECL(-432000000000)  },
       { 1981, TSDECL(31190400000000) },
       { 1982, TSDECL(62726401000000) },
       { 1983, TSDECL(94262402000000) },
       { 1984, TSDECL(125798403000000) },
       { 1985, TSDECL(157420803000000) },
       { 1986, TSDECL(188956804000000) },
       { 1987, TSDECL(220492804000000) },
       { 1988, TSDECL(252028805000000) },
       { 1989, TSDECL(283651205000000) },
       { 1990, TSDECL(315187206000000) },
       { 1991, TSDECL(346723207000000) },
       { 1992, TSDECL(378259207000000) },
       { 1993, TSDECL(409881608000000) },
       { 1994, TSDECL(441417609000000) },
       { 1995, TSDECL(472953610000000) },
       { 1996, TSDECL(504489611000000) },
       { 1997, TSDECL(536112011000000) },
       { 1998, TSDECL(567648012000000) },
       { 1999, TSDECL(599184013000000) },
       { 2000, TSDECL(630720013000000) },
       { 2001, TSDECL(662342413000000) },
       { 2002, TSDECL(693878413000000) },
       { 2003, TSDECL(725414413000000) },
       { 2004, TSDECL(756950413000000) },
       { 2005, TSDECL(788572813000000) },
       { 2006, TSDECL(820108814000000) },
       { 2007, TSDECL(851644814000000) },
       { 2008, TSDECL(883180814000000) },
       { 2009, TSDECL(914803215000000) },
       { 2010, TSDECL(946339215000000) }
};

int  isLeap( int year ) {
        return (((year) % 4) == 0 && (((year) % 100) != 0 || ((year) % 400) == 0));
}

int month_dayToJulian( int day, int month, int leapIX ) {
        int julian=day;
        for( int i=0; i<(month-1); i++ ) julian += dayArr[leapIX][i];
        return julian;
}

int daysSinceT0( int year, int julianDay ) {
        int days=0;
        if( year >= 1980 ) {
               days=julianDay-6; /* correct for starting day 6 jan 1980 */
               for( int y=1980; y<year; y++ )
                      days += 365+isLeap(y);
        } else {
               days = julianDay - 365 - isLeap(year) - 6;
               for( int y=1979; y>year; y--)
                      days -= 365+isLeap(y);
        }
        return days;
}
```

```
int leapSecs( int daysSinceT0 ) {
        for( int i=NLEAP-1; i>=0; i-- ) {
                if(daysSinceT0 >= dayLeapArr[i].days ) {
                        /* fprintf(stdout, "leapSecs: %d\n", dayLeapArr[i].leapsecs); */
                        return dayLeapArr[i].leapsecs;
                }
        }
        fprintf(stderr, "Illegal number in leapSec calculation, is date after 1970?\n");
        return 0; /* should never happen with legal SEG-D timestamps, throw exception here */
}

int leapSecs( Timestamp ts ) {
        for( int i=NLEAP-1; i>=0; i-- ) {
                if(ts >= tsLeapArr[i].timestamp ) {
                        fprintf(stdout, "leapSecs: %d\n", tsLeapArr[i].leapsecs);
                        return tsLeapArr[i].leapsecs;
                }
        }
        fprintf(stderr, "Illegal number in leapSec calculation, is date after 1970?\n");
        return 0; /* should never happen with legal SEG-D timestamps, throw exception here */
}


Timestamp utcToTimestamp( const Utc& utc ) {
        Int leapIX = isLeap(utc.year) ? 1 : 0;    /* february */
        int julian = month_dayToJulian(utc.day,utc.month,leapIX);
        int nd = daysSinceT0(utc.year,julian); /* nd negative if date before 6 jan 1980 */
        Timestamp ts = 1000000*(Timestamp)((Timestamp)nd*24*3600 + utc.hour*3600 + utc.minute*60 + utc.sec) +
                utc.microsec;
        return ts + leapSecs(nd)*1000000;
}

Utc timestampToUtc( Timestamp ts ) {
    Utc result(0,0,0,0,0,0,0);
    int iy, nd, md, im;
    int leapIX;
    Timestamp tsJan1;
    for(iy = 0 ; iy < (sizeof(tsJan1YearArr)/sizeof(tsJan1YearArr[0])); ++iy) {
        if(tsJan1YearArr[iy].timestamp <= ts) {
             result.year = tsJan1YearArr[iy].year;
        } else {
             break;
        }
    }
    /* if off the end of the charts, return all zeros */
    if(result.year == 0) return result;
    else tsJan1 = tsJan1YearArr[iy-1].timestamp;

    /* reduce to number of microseconds since beginning of year */
    ts -= (1000000 * leapSecs(ts));
    tsJan1 -= (1000000 * leapSecs(tsJan1));
    ts -= tsJan1;

    /* extract microseconds */
    result.microsec = (int) (ts%1000000);
    ts -= result.microsec;
    ts /= 1000000;

    /* extract seconds */
    result.sec = (int) (ts%60);
    ts -= result.sec;
    ts /= 60;

    /* extract minutes */
    result.minute = (int) (ts%60);
    ts -= result.minute;
    ts /= 60;

    /* extract hours */
    result.hour = (int) (ts%24);
    ts /= 24;

    /* figure month and day from number of days since start of year */
```

```
    nd = (int) (ts);
    md = 0;
    leapIX = isLeap(result.year) ? 1 : 0; /* february */
    for( im = 0; im < 12; ++im ) {
         if( md > nd ) break;
         md += dayArr[leapIX][im];
    }
    result.month = im;
    result.day = 1 + nd + dayArr[lepaIX][im-1] - md;

    if(result.day == 32) { /* leap second */
       result.day = 31; result.hour = 23; result.minute = 59; result.sec = 60;
    }

    return result;
}

int main(int argc, char **argv)
{
  int yrlist[] = { 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979,
                   1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989,
                   1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
                   2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009,
                   2010 };
  int mnlist[] = {    1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
                      1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
                      1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
                      1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
                      1 };
  size_t i;
  Utc testTime(2005,12,31,23,59,60,0);
  Timestamp ts;
  ts = utcToTimestamp(testTime);
  fprintf(stdout,"%s is Timestamp " tsfmt "\n", testTime.toString(), ts);
  testTime = timestampToUtc( ts );
  fprintf(stdout,"%s is Timestamp " tsfmt "\n", testTime.toString(), ts);
  testTime.day = 1;
  testTime.hour = 0;
  testTime.minute = 0;
  testTime.sec = 0;
  testTime.microsec = 0;

  for(i=0; i<(sizeof(yrlist)/sizeof(yrlist[0])); ++i) {
      testTime.year = yrlist[i];
      testTime.month = mnlist[i];
      fprintf(stdout,"%s is Timestamp " tsfmt "\n", testTime.toString(), utcToTimestamp(testTime));
  }

  return EXIT_SUCCESS;
}
```

E.10 Electromagnetic (EM) survey

An Electromagnetic survey is acquired using 100 receivers deployed on the sea floor, one source vessel towing a 300m long source and 20 receivers connected to a streamer towed behind the source vessel. This hypothetical controlled source electromagnetic (CSEM) setup is chosen to show all aspects of an EM survey. At the same time magnetotelluric (MT) data will be recorded for all seafloor receivers.

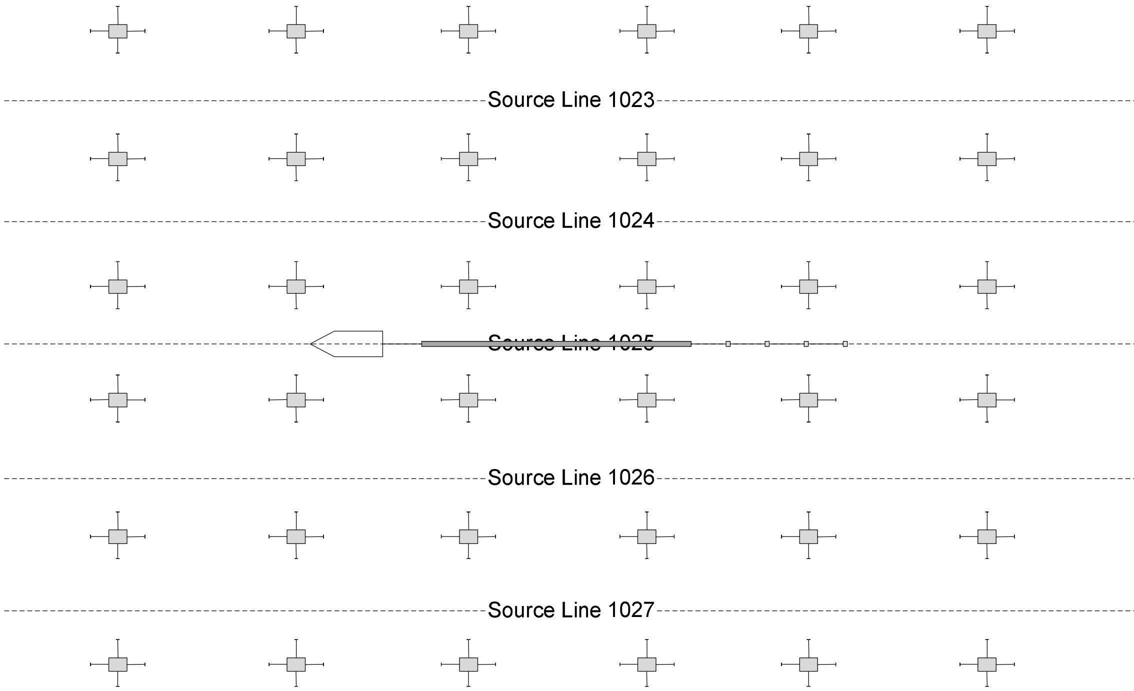


Figure shows a overview of an Electromagnetic survey using EM sensors on the sea floor, and a source vessels with towed sensors behind. Note: The relative sizes between equipment are not correct.

Each seafloor sensor unit consists of 6 different magnetic and electric field sensors, which produces 6 different data traces. The sensor units are completely independent, with no external synchronization while they are deployed. The sensor units acquire data in SEGD format similar to the example described in E.8 Extended Recording Mode example.

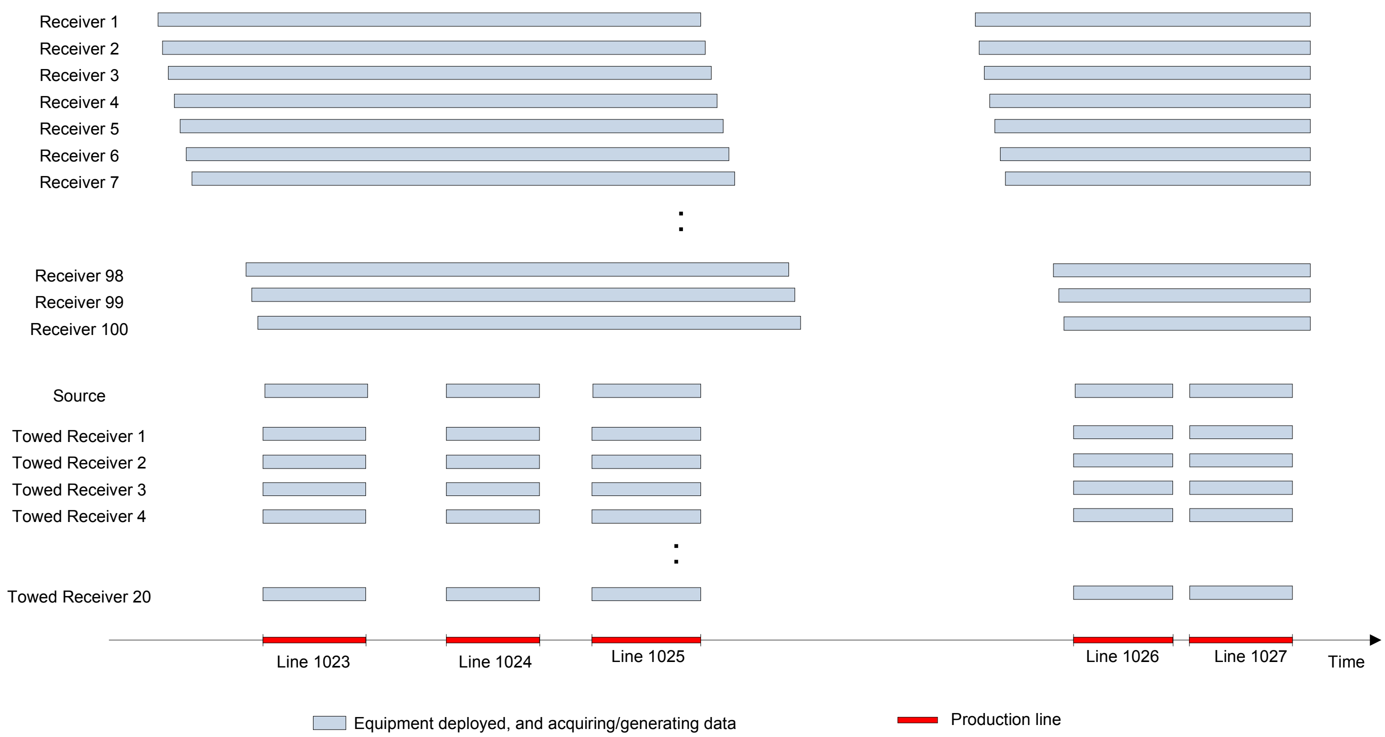


Figure shows an overview of production in an EM survey, with some sensors deployed on the seafloor, and some attached to the source vessel.

The sensors are placed on the sea floor, and data are acquired organized in source lines. After several days the sensors are retrieved, and the data from each sensor unit are read and stored in the acquisition system onboard the vessel as separate files.

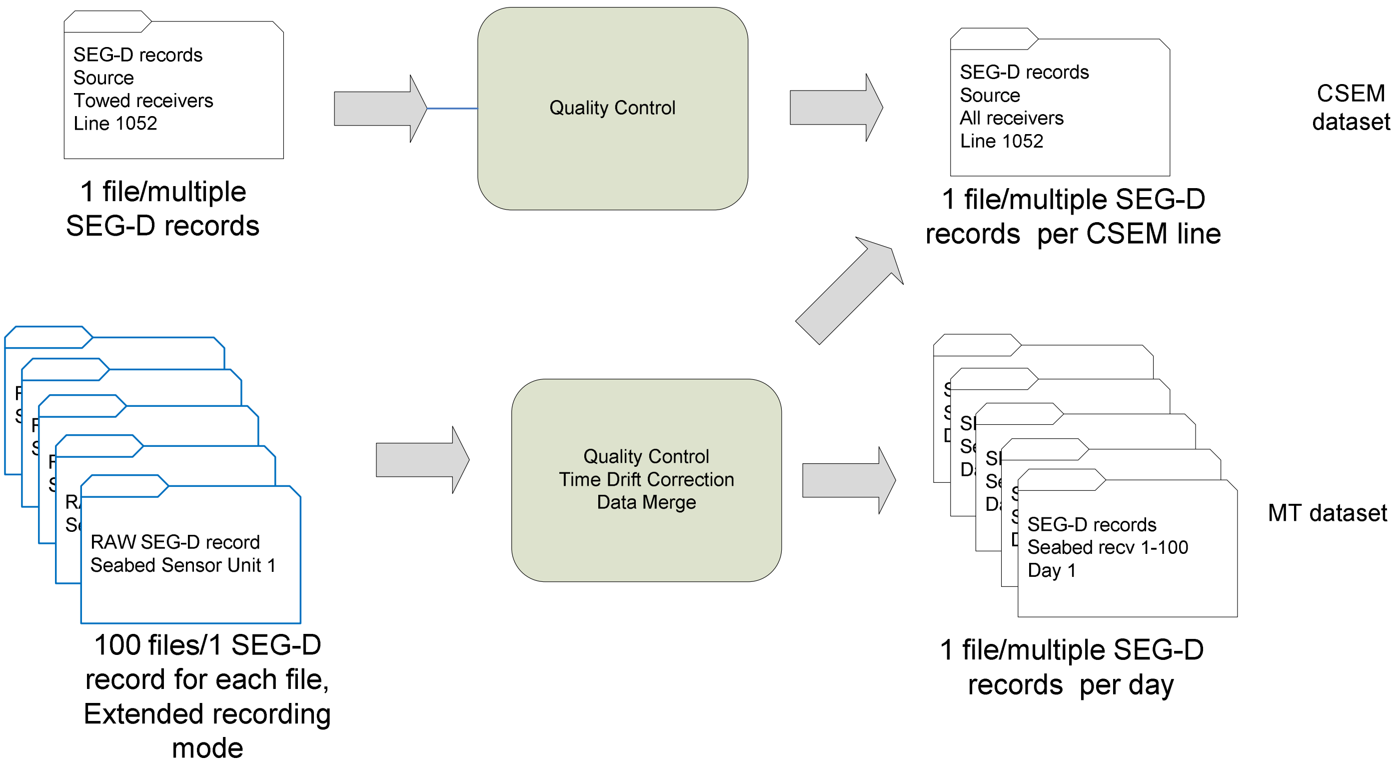
Seafloor receiver data:
The raw data records from each seabed sensor unit consists of 6 different channelsets, one for each different channel, recorded as a continuous record using continuous recording mode, splitting traces into 120 second pieces (see E.8 Extended Recording Mode example for further information). As the raw data from each sensor unit is acquired using drifting clocks, the data must be time drift corrected to match GPS time prior to merging with other sensor units.

Towed receiver data:
The data from source and towed receivers are acquired in 18 second normal SEG-D records (which matches the source signal pattern cycle), i.e. each individual SEG-D record contains all source and towed receiver data for 18 seconds. The auxiliary data for the source consists of the reference signal, and a measured signal (electrical field sensor close to the source). The data from the source (auxiliary traces), and the towed sensors are synchronized to GPS as they are acquired, and hence contains no time drift.

The client deliverable will be two datasets:

- *CSEM*: The data will be organized as SEG-D records containing data from all sensors organized into lines similar to a Land/Seabed survey. The EM source emits a continuous signal, so there are no real source events like in standard Marine/Land/Seabed surveys. The source however cycles through a signal pattern which is repeated every 18 seconds, and it is decided to create SEG-D records matching this cycle. Hence, data will be stored as normal SEG-D records each 18 sec long (not using extended recording mode).

- *MT*: The data will be organized into SEG-D records containing data from all seabed sensors for a period of time. Since there are no source events, the record size may be arbitrarily chosen, in this example 120 seconds. The data will then be grouped into one record set per day.



Summary of data flow - extended recording mode records are blue, normal recording mode records are black.

The data from the seafloor sensors are stored in files (one per unit) containing all the data from the entire time the sensor unit was deployed. Before the data from a sensor unit can be merged with other data, it is quality controlled and time drift corrected.

CSEM

For each towline the data is selected from the active sensor unit records based on towline start and end time, and merged with the records from the Source and Towed Receivers. The source data ends up as aux traces. The client deliverable SEGD record contains data from the source and all sensors for the towline time period.

MT

For MT data, a record will contain all channels for all active receivers. A fixed record length is chosen, in this example 120 seconds, and multiple 120 sec SEG-D records for a given time period (in this example one day) is stored in each disk file. If survey is a combined MT/CSEM survey, there is no need to remove CSEM data from the MT dataset.

Example CSEM SEG-D Block layout.

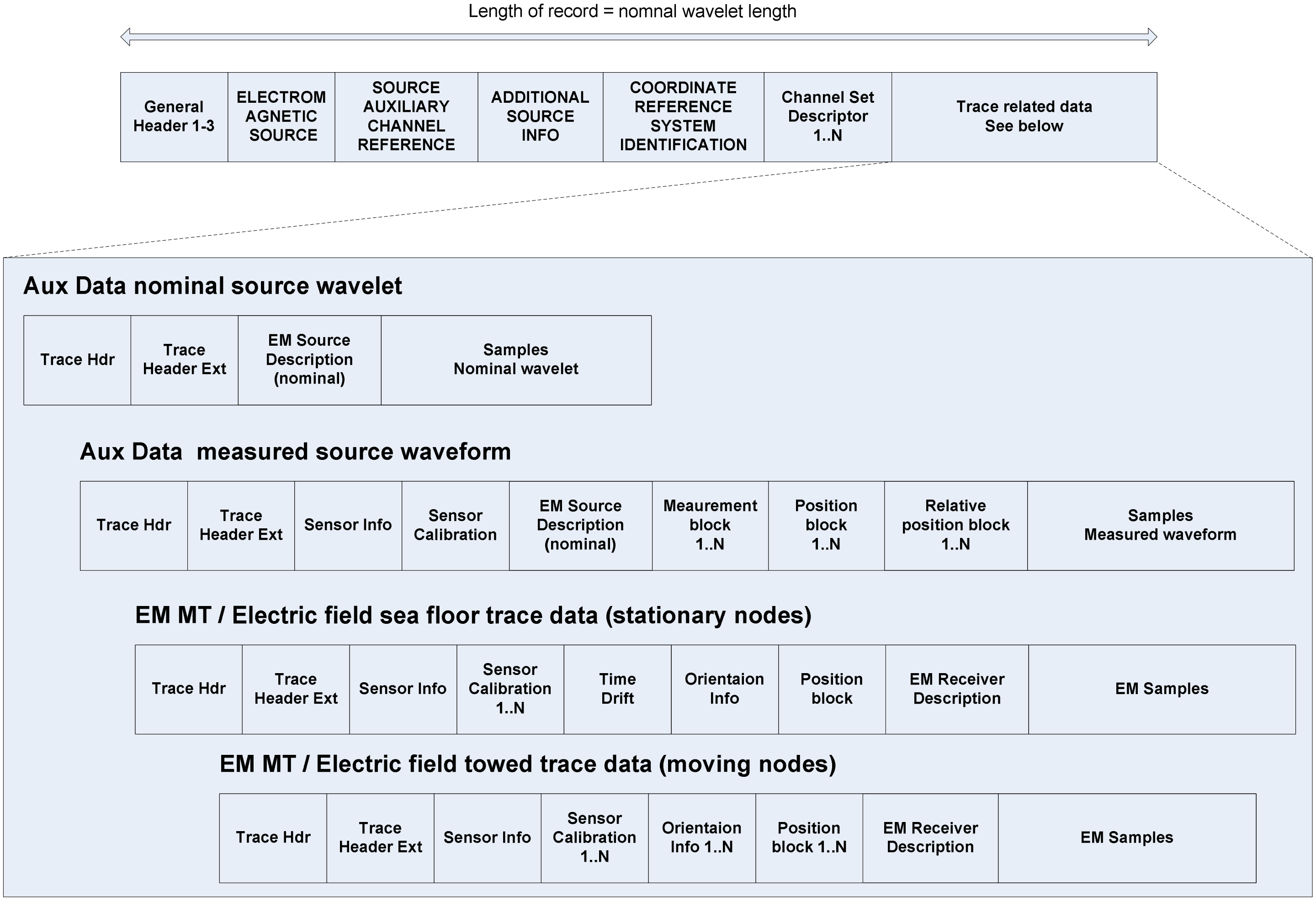


Figure shows CSEM SEG-D record block layout (block sizes are not to scale)

This figure shows how to map EM specific data and header values into SEG-D. A few comments:

- Note that source position will have dipole midpoint in Position block and electrode position in the Relative position blocks.
- Conductivity and other relevant data on the towfish will be in the Measurement blocks.
- The example covers both towed and stationary receivers. The actual selection of block configuration varies between the two situations. The main difference is towed receivers will have multiple position and orientation trace header blocks for the EM traces because the receiver is moving during the record.