SEG-Y_r2.0: SEG-Y revision 2.0 Data Exchange format[1]

SEG Technical Standards Committee[2]

January 2017

[2] Editors Rune Hagelund and Stewart A. Levin

Contents

FIGURES

TABLES

1. Introduction

Since the original SEG-Y Data Exchange Format (revision 0, - see Appendix G. References) was published in 1975 it has achieved widespread usage within the geophysical industry. The 2002 revision 1 extended the standard to 3-D acquisition and high capacity media, reducing, though far from eliminating, the use of proprietary variations. Since the publication of SEG-Y rev 1, the nature of seismic data acquisition, processing and seismic hardware has continued to evolve and the SEG Technical Standards Committee undertook further revision. In line with the 2011 SEG D rev 3.0 standard, this revision both addresses current industry data exchange needs and provides an explicit mechanism to support future expansion with both proprietary and officially-adopted extensions. The SEG Technical Standards Committee strongly encourages producers and users of SEG-Y data sets to move to the revised standard in an expeditious fashion.

Users of this standard are cautioned that SEG-Y was not explicitly designed for use as a field recording format. The SEG D or SEG 2 formats are recommended for this purpose.

2. Summary

2.1. Unchanged Items

- EBCDIC encoding allowed for text
- The size of the original 3200-byte Textual File Header, 400-byte Binary File Header and initial 240-byte Trace Header

2.2. Changes from rev 1 to rev 2

- Provide for up to 65535 additional 240 byte trace headers with bytes 233-240 of each trace header reserved for trace header names
- Allow up to 2^{32} –1 samples per trace
- Allow arbitrarily large and small sample intervals
- Permit up to 2^{64}–1 traces per line and 2^{32}–1 traces per ensemble
- Support additional data sample formats, including IEEE double precision (64 bit)
- Support little-endian and pair-wise byte swapping to improve I/O performance.
- Support microsecond accuracy in time and date stamps
- Support additional precision on coordinates, depths and elevations (especially useful for lat/long and UTM coordinates) and more options for coordinate reference system specification
- Require Extended Textual File Header stanzas to begin at 3200-byte boundaries and removed 40 80-byte line restriction
- Allow stanzas to appear after the last data trace
- Provide flexible trace header mapping options via Extended Textual File Headers. Because of this ability, we remove almost all “mandatory” and “highly recommended” header entry designations
- Allow XML-based Extended Textual File Header and Trailer stanzas for ease of machine encoding and decoding
- Include depth, velocity, EM, gravity and rotational sensor data

2.3. Changes from rev 0 to rev 1

- A SEG-Y file may be written to any medium that is resolvable to a stream of variable length records
- The data word formats are expanded to include four-byte, IEEE floating-point and one-byte integer data words
- A small number of additional fields in the 400-byte Binary File Header and the

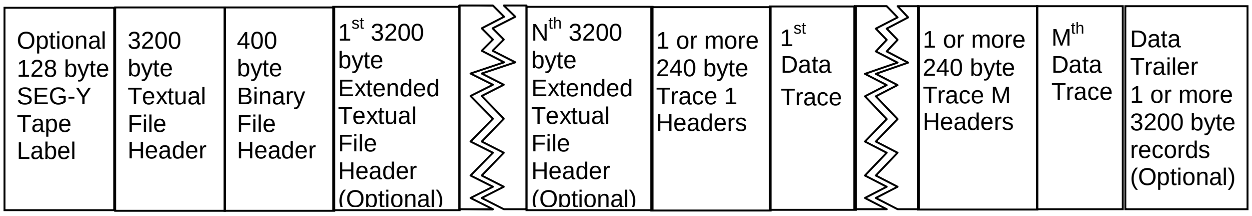

Figure 1 Byte stream structure of a SEG-Y file with N Extended Textual File Header records and M traces records

240-byte Trace Header are defined and the use of some existing entries is clarified

- An Extended Textual File Header consisting of additional 3200-byte Textual File Header blocks is introduced
- The data in the Extended Textual File Header uses a stanza layout and standard stanzas are defined
- Trace identification is expanded
- Engineering conversions are introduced
- The Textual File Header and the Extended Textual File Header can be encoded as EBCDIC or ASCII characters

2.4. Notation

The term CDP (common depth point) as used in this document is used as a synonym for the term CMP (common midpoint).

2.5. Controlling Organization

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

Society of Exploration Geophysicist
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

2.6. Acknowledgments

The SEG Technical Standards Committee would like to acknowledge the time and effort put forth by a great many individuals and organizations.

3. SEG-Y File Structure

The SEG-Y format is intended to be independent of the actual medium on which it is recorded. For this standard, the terms file and data set are synonymous. Both terms are a collection of logically related data traces or ensembles of traces and the associated ancillary data.

3.1. Recording Medium

A SEG-Y file may be written to any sequential medium. Whatever medium is used, the data must be resolvable to a stream of variable length logical records. This includes high capacity tape devices, although with these it is desirable to use some kind of blocking and/or logical encapsulation such as SEG RODE (Booth et al., 1997), to use the tape more efficiently and possibly to allow the recording of associated metadata. Obviously, when seismic data are being exchanged in SEG-Y format, the medium and any blocking and/or encapsulation scheme used must be acceptable to both the provider and recipient of the data.

One important class of media on which SEG-Y data are exchanged is the byte stream without any record structure. It is common practice to write SEG-Y data to disk, including USB sticks, CD and DVD-ROM, or streamed through a network for data distribution. Certain rules have to be followed for this to work correctly. Appendix A defines how SEG-Y data should be written as a byte stream.

In order to make SEG-Y consistent with the SEG D Rev 3.0 standard, Appendix B defines a tape label for SEG-Y tapes, using a format based on the RP66 Storage Unit Label. Labels are not mandatory for SEG-Y, but their use is highly desirable in environments such as robotic tape libraries and large scale processing centers.

Appendix C defines a simple blocking scheme for SEG-Y data to allow more efficient use of high-capacity tape media. This is based on the scheme defined in the SEG D Rev 3.0 standard.

3.2. File Structure

Figure 1 illustrates the structure of a SEG-Y file. Following the optional SEG-Y Tape Label, the next 3600 bytes of the file are the Textual File Header and the Binary File Header written as a concatenation of a 3200-byte record and a 400-byte record. This is optionally followed by Extended Textual File Header(s), which consists of zero or more 3200-byte Extended Textual File Header records. The remainder of the SEG-Y file contains a variable number of Data Trace records that are each preceded by a 240-byte Standard Trace Header and zero or more 240-byte Trace Header Extensions. The Trace Header Extension mechanism is the only structural change introduced in this revision and while not strictly backward compatible with prior SEG-Y formats, it has been carefully designed to have minimal impact on existing SEG-Y reader software. It should be simple for existing software to be modified to detect the presence of the optional trace headers and either process or ignore any Proprietary Trace Header Extensions. The format of Trace Header Extensions is described fully in section 7.

3.3. Number Formats

In earlier SEG-Y standards, all binary values were defined as using "big-endian" byte ordering. This means that, within the bytes that make up a number, the most significant byte (containing the sign bit) is written closest to the beginning of the file and the least significant byte is written closest to the end of the file. With SEG-Y rev 2, "little-endian" and "pairwise byte-swapped" byte ordering are allowed, primarily for I/O performance. This is independent of the medium to which a particular SEG-Y file is written (i.e. the byte ordering is no different if the file is written to tape on a mainframe or to disk on a PC). These alternate byte orders are identified by examining bytes 3297-3300 in the Binary File Header and apply only to the Binary File Header, Trace Headers, and Trace Samples[3].

All values in the Binary File Header and the SEG defined Trace Headers are to be treated as two's complement integers, whether two, four or eight bytes long, with the exception of the new 8-character Trace Header Extension name, an optional IEEE double precision sample rate, and fields that cannot be negative such as the number of samples per trace. *To aid in data recognition and recovery, a value of zero in any SEG or user assigned fields of these headers should indicate an unknown or unspecified value unless explicitly stated otherwise*.

Trace Data sample values are either integers or floating-point numbers. Signed integers are in two's complement format. SEG-Y revision 2 adds unsigned integers,

[3] Textual Headers and Data Trailer records are always assumed to be text and so byte ordering is left untouched.

24 and 64 bit integers and IEEE floating-point data sample types.

3.4. Varying Trace Lengths

The SEG-Y standard specifies fields for sample interval and number of samples at two separate locations in the file. The Binary File Header contains values that apply to the whole file and the Trace Headers contain values that apply to the associated trace. In SEG-Y, varying trace lengths in a file are explicitly allowed. The values for sample interval and number of samples in the Binary File Header should be for the primary set of seismic data traces in the file. This approach allows the Binary File Header to be read and say, for instance, “this is six seconds data sampled at a two-millisecond interval”. The value for the number of samples in each individual Trace Header may vary from the value in the Binary File Header and reflect the actual number of samples in a trace. The number of bytes in each trace record must be consistent with the number of samples in the Trace Header. This is particularly important for SEG-Y data written to disk files (see Appendix A).

Allowing variable length traces complicates random access in a disk file, since the locations of traces after the first are not known without pre-scanning the file. To facilitate the option of random access, a field in the Binary File Header defines a fixed length trace flag. If this flag is set, all traces in the file must have the same length. This will typically be the case for poststack data.

3.5. Coordinates

Knowing the source and receiver locations is a primary requirement for processing seismic data, and knowing the location of the processed data with respect to other data is essential for interpretation. Traditionally seismic coordinates have been supplied as geographic coordinates and/or grid coordinates. SEG-Y accommodates either form. However locations are ambiguous without clear coordinate reference system (CRS) definition. SEG-Y provides the ability to define the CRS used for the coordinates contained within the Binary Header, the Extended Textual Header and the Trace Headers. To avoid confusion, this standard requires that a single CRS ***must*** be used for all coordinates within an individual SEG-Y data set. Additionally, the coordinate units must be the same for all coordinates. The SEG-Y CRS definitions conform to those in SEG D Rev 3.0 or the OGP P1/11 standards referenced in Appendix G.

4. Textual File Header

The first 3200-byte, Textual File Header record contains 40 lines of textual information, providing a human-readable description of the seismic data in the SEG-Y file. This information is free form and is the least well-defined of the headers in the 1975 standard, although the standard did provide a suggested layout for the first 20 lines. While there would be distinct advantages in making the layout of this header more rigid, it was decided that it would not be practicable to produce a layout that would be universally acceptable in the light of how it is currently used. It originally was encoded in the EBCDIC character set (see Appendix F) but ASCII is now allowed for all Textual File Headers.

The SEG-Y standard defines a separate textual header with a more comprehensively defined structure, where textual information can be stored in a machine-readable way. This Extended Textual File Header is described in detail in section 6. Note that the “traditional” Textual File Header is completely separate from the Extended Textual File Header and will still be the primary location for human readable information about the contents of the file. In particular, it should contain information about any unusual features in the file, such as if the delay recording time in trace header bytes 109-110 is non-zero. The revision level of the SEG-Y format (Binary File Header bytes 3501-3502) being used ***must*** be included for all files written in the SEG-Y rev 2 format. It is mandatory that the SEG-Y revision level now be included in the Textual File Header. Table 1 is an example Textual File Header with the SEG-Y revision level included in the 39th record.

Table 1 Textual File Header

3200-byte Textual File Header							
Cols 1-10	**Cols 11-20**	**Cols 21-30**	**Cols 31-40**	**Cols 41-50**	**Cols 51-60**	**Cols 61-70**	**Cols 71-80**
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
C 1 CLIENT			COMPAN	Y		CREW N	O
C 2 LINE		AREA		MA	P ID		
C 3 REEL N	O	DAY-STAR	T OF REEL	YEAR	OBSERV	ER	
C 4 INSTRU	MENT: MFG		MODEL	SE	RIAL NO		
C 5 DATA T	RACES/RECO	RD	AUXILIARY	TRACES/REC	ORD	CDP FOLD	
C 6 SAMPLE	INTERVAL	SA	MPLES/TRAC	E BI	TS/IN	BYTES/SAMP	LE
C 7 RECORD	ING FORMAT	FO	RMAT THIS	REEL	MEASUREM	ENT SYSTEM	
C 8 SAMPLE	CODE: FLO	ATING PT	FIXED P	T FIXE	D PT-GAIN	CORREL	ATED
C 9 GAIN	TYPE: FIXE	D BINA	RY FLO	ATING POIN	T OTHE	R	
C10 FILTER	S: ALIAS	HZ NOT	CH HZ	BAND	- HZ	SLOPE -	DB/OCT
C11 SOURCE	: TYPE	NU	MBER/POINT	PO	INT INTERV	AL	
C12 PA	TTERN:			LENGTH	WID	TH	
C13 SWEEP:	START	HZ END	HZ LEN	GTH M	S CHANNEL	NO TY	PE
C14 TAPER:	START LEN	GTH	MS END LE	NGTH	MS TYPE		
C15 SPREAD	: OFFSET	MAX	DISTANCE	GROU	P INTERVAL		
C16 GEOPHO	NES: PER G	ROUP S	PACING	FREQUENCY	MFG	MO	DEL
C17 PA	TTERN:			LENGTH	WID	TH	
C18 TRACES	SORTED BY	: RECORD	CDP	OTHER			
C19 AMPLIT	UDE RECOVE	RY: NONE	SPHERI	CAL DIV	AGC	OTHER	
C20 MAP PR	OJECTION			ZONE ID	COORDI	NATE UNITS	[4]
C21 PROCES	SING:						
C22 PROCES	SING:						
C23							
...							
C38							
C39 SEG-Y_	REV2.0						
C40 END TE	XTUAL HEAD	ER [5]					

[4] C20 is overridden by the contents of location data stanza in an Extended Textual Header record

[5] C40 END EBCDIC is also acceptable but C40 END TEXTUAL HEADER is the preferred encoding.

5. Binary File Header

The 400-byte Binary File Header record contains binary values relevant to the whole SEG-Y file. The values in the Binary File Header are defined as two-byte or four-byte, two's complement or unsigned integers, with the exception of IEEE double precision sample intervals in the optional SEG00001 trace header. Certain values in this header are crucial for the processing of the data in the file, particularly the sample interval, trace length and format code. Revision 2.0 defines a few additional fields in the optional portion, as well as providing some clarification on the use of some existing entries.

Table 2 Binary File Header[6]

400-byte Binary File Header	
Byte	**Description**
3201–3204	Job identification number.
3205–3208	Line number. For 3-D poststack data, this will typically contain the in-line number.
3209–3212	Reel number.
3213–3214	Number of data traces per ensemble. Mandatory for prestack data.
3215–3216	Number of auxiliary traces per ensemble. Mandatory for prestack data.
3217–3218	Sample interval. Microseconds (µs) for time data, Hertz (Hz) for frequency data, meters (m) or feet (ft) for depth data.
3219–3220	Sample interval of original field recording. Microseconds (µs) for time data, Hertz (Hz) for frequency data, meters (m) or feet (ft) for depth data.
3221–3222	Number of samples per data trace. Note: The sample interval and number of samples in the Binary File Header should be for the primary set of seismic data traces in the file.
3223–3224	Number of samples per data trace for original field recording.

[6] Where not otherwise indicated, a value of zero indicates an unknown or unspecified value.

400-byte Binary File Header	
Byte	**Description**
3225–3226	Data sample format code. Mandatory for all data. These formats are described in Appendix E. 1 = 4-byte IBM floating-point 2 = 4-byte, two's complement integer 3 = 2-byte, two's complement integer 4 = 4-byte fixed-point with gain (obsolete) 5 = 4-byte IEEE floating-point 6 = 8-byte IEEE floating-point 7 = 3-byte two's complement integer 8 = 1-byte, two's complement integer 9 = 8-byte, two's complement integer 10 = 4-byte, unsigned integer 11 = 2-byte, unsigned integer 12 = 8-byte, unsigned integer 15 = 3-byte, unsigned integer 16 = 1-byte, unsigned integer
3227–3228	Ensemble fold — The expected number of data traces per trace ensemble (e.g. the CMP fold).
3229–3230	Trace sorting code (i.e. type of ensemble) : –1 = Other (should be explained in a user Extended Textual File Header stanza) 0 = Unknown 1 = As recorded (no sorting) 2 = CDP ensemble 3 = Single fold continuous profile 4 = Horizontally stacked 5 = Common source point 6 = Common receiver point 7 = Common offset point 8 = Common mid-point 9 = Common conversion point
3231–3232	Vertical sum code: 1 = no sum, 2 = two sum, …, N = M–1 sum (M = 2 to 32,767)
3233–3234	Sweep frequency at start (Hz).
3235–3236	Sweep frequency at end (Hz).
3237–3238	Sweep length (ms).

400-byte Binary File Header	
Byte	**Description**
3239–3240	Sweep type code: 1 = linear 2 = parabolic 3 = exponential 4 = other
3241–3242	Trace number of sweep channel.
3243–3244	Sweep trace taper length in milliseconds at start if tapered (the taper starts at zero time and is effective for this length).
3245–3246	Sweep trace taper length in milliseconds at end (the ending taper starts at sweep length minus the taper length at end).
3247–3248	Taper type: 1 = linear 2 = cosine squared 3 = other
3249–3250	Correlated data traces: 1 = no 2 = yes
3251–3252	Binary gain recovered: 1 = yes 2 = no
3253–3254	Amplitude recovery method: 1 = none 2 = spherical divergence 3 = AGC 4 = other
3255–3256	Measurement system: If Location Data stanzas are included in the file, this entry would normally agree with the Location Data stanza. If there is a disagreement, the last Location Data stanza is the controlling authority. If units are mixed, e.g. meters on surface, feet in depth, then a Location Data stanza is mandatory. 1 = Meters 2 = Feet
3257–3258	Impulse signal polarity 1 = Increase in pressure or upward geophone case movement gives negative number on trace. 2 = Increase in pressure or upward geophone case movement gives positive number on trace.

400-byte Binary File Header	
Byte	**Description**
3259–3260	Vibratory polarity code: Seismic signal lags pilot signal by: 1 = 337.5°to 22.5° 2 = 22.5°to 67.5° 3 = 67.5°to 112.5° 4 = 112.5°to 157.5° 5 = 157.5°to 202.5° 6 = 202.5°to 247.5° 7 = 247.5°to 292.5° 8 = 292.5°to 337.5°
3261–3264	Extended number of data traces per ensemble. If nonzero, this overrides the number of data traces per ensemble in bytes 3213–3214.
3265–3268	Extended number of auxiliary traces per ensemble. If nonzero, this overrides the number of auxiliary traces per ensemble in bytes 3215–3216.
3269–3272	Extended number of samples per data trace. If nonzero, this overrides the number of samples per data trace in bytes 3221–3222.
3273–3280	Extended sample interval, IEEE double precision (64-bit). If nonzero, this overrides the sample interval in bytes 3217–3218 with the same units.
3281–3288	Extended sample interval of original field recording, IEEE double precision (64-bit) . If nonzero, this overrides the sample interval of original field recording in bytes 3219–3220 with the same units.
3289–3292	Extended number of samples per data trace in original recording. If nonzero, this overrides the number of samples per data trace in original recording in bytes 3223–3224.
3293–3296	Extended ensemble fold. If nonzero, this overrides ensemble fold in bytes 3227–3228.
3297–3300	The integer constant 16909060_{10} (01020304_{16}). This is used to allow unambiguous detection of the byte ordering to expect for this SEG-Y file. For example, if this field reads as 67305985_{10} (04030201_{16}) then the bytes in every Binary File Header, Trace Header and Trace Data field must be reversed as they are read, i.e. converting the endian-ness of the fields. If it reads 33620995_{10} (02010403_{16}) then consecutive pairs of bytes need to be swapped in every Binary File Header, Trace Header and Trace Data field. The byte ordering of all other portions (the Extended Textual Header and Data Trailer) of the SEG-Y file is not affected by this field.
3301–3500	Unassigned

400-byte Binary File Header	
Byte	**Description**
3501	Major SEG-Y Format Revision Number. This is an 8-bit unsigned value. Thus for SEG-Y Revision 2.0, as defined in this document, this will be recorded as 02_{16}. This field is mandatory for all versions of SEG-Y, although a value of zero indicates "traditional" SEG-Y conforming to the 1975 standard.
3502	Minor SEG-Y Format Revision Number. This is an 8-bit unsigned value with a radix point between the first and second bytes. Thus for SEG-Y Revision 2.0, as defined in this document, this will be recorded as 00_{16}. This field is mandatory for all versions of SEG-Y.
3503–3504	Fixed length trace flag. A value of one indicates that all traces in this SEG-Y file are guaranteed to have the same sample interval, number of trace header blocks and trace samples, as specified in Binary File Header bytes 3217–3218 or 3281–3288, 3517–3518, and 3221–3222 or 3289–3292. A value of zero indicates that the length of the traces in the file may vary and the number of samples in bytes 115–116 of the Standard SEG-Y Trace Header and, if present, bytes 137–140 of SEG-Y Trace Header Extension 1 must be examined to determine the actual length of each trace. This field is mandatory for all versions of SEG-Y, although a value of zero indicates "traditional" SEG-Y conforming to the 1975 standard. Irrespective of this flag, it is strongly recommended that corect values for the number of samples per trace and sample interval appear in the appropriate trace Trace Header locations.
3505–3506	Number of 3200-byte, Extended Textual File Header records following the Binary Header. If bytes 3521–3528 are nonzero, that field overrides this one. A value of zero indicates there are no Extended Textual File Header records (i.e. this file has no Extended Textual File Header(s)). A value of -1 indicates that there are a variable number of Extended Textual File Header records and the end of the Extended Textual File Header is denoted by an ((SEG: EndText)) stanza in the final record (Section 6.2). A positive value indicates that there are exactly that many Extended Textual File Header records. Note that, although the exact number of Extended Textual File Header records may be a useful piece of information, it will not always be known at the time the Binary Header is written and it is not mandatory that a positive value be recorded here or in bytes 3521–3528. It is however recommended to record the number of records if possible as this makes reading more effective and supports direct access to traces on disk files. In the event that this number exceeds 32767, set this field to –1 and bytes 3521–3528 to 3600+3200*(number of Extended Textual File Header records). Add a further 128 if a SEG-Y Tape Label is present.
3507–3510	Maximum number of additional 240 byte trace headers. A value of zero indicates there are no additional 240 byte trace headers. The actual number for a given trace may be supplied in bytes 157–158 of SEG-Y Trace Header Extension 1.

3511–3512	Time basis code: 1 = Local 2 = GMT (Greenwich Mean Time) 3 = Other, should be explained in a user defined stanza in the Extended Textual File Header 4 = UTC (Coordinated Universal Time) 5 = GPS (Global Positioning System Time)
3513–3520	Number of traces in this file or stream. (64-bit unsigned integer value) If zero, all bytes in the file or stream are part of this SEG-Y dataset.
3521–3528	Byte offset of first trace relative to start of file or stream if known, otherwise zero. (64-bit unsigned integer value) This byte count will include the initial 3600 bytes of the Textual and this Binary File Header plus the Extended Textual Header if present. When nonzero, this field overrides the byte offset implied by any nonnegative number of Extended Textual Header records present in bytes 3505–3506.
3529–3532	Number of 3200-byte data trailer stanza records following the last trace (4 byte signed integer). A value of 0 indicates there are no trailer records. A value of -1 indicates an undefined number of trailer records (0 or more) following the data. It is, however, recommended to record the number of trailer records if possible as this makes reading more efficient.
3533–3600	Unassigned

6. Extended Textual File Header

If bytes 3505–3506 of the Binary File Header are non-zero, then an Extended Textual File Header[7] is present in the SEG-Y file. The Extended Textual File Header follows the Binary File Header record and precedes the first Data Trace record. Similarly, if bytes 3529–3532 are nonzero, a trailer with the same formatting appears after the last Data Trace. An Extended Textual File Header consists of one or more 3200-byte records and provides additional space for recording information about the SEG-Y file in a flexible but well defined way. The kind of information recorded here will include trace header mappings, coordinate reference system details, 3-D bin grids, processing history and acquisition parameters. It is recommended that stanza information be included only once per SEG-Y rev 2 file. In the event multiple or conflicting data entries are included in the SEG-Y rev 2 file, the last data entry is assumed to be correct.

The data in the Extended Textual File Header are organized in the form of stanzas, a format also used for trailer records. Appendix D defines a set of predefined stanzas. It is intended that additional stanzas will be defined in the future revisions to thisd other SEG-defined standards. However, the stanza mechanism is intended to be flexible and extensible and it is perfectly acceptable to define private stanzas. For the sake of usability, data exchange and maximum benefit, a standard SEG defined stanza should be used if it exists for the type of information required.

[7] While binary data may possibly appear in User Header stanzas, all SEG-defined Extended Textual Header stanzas have their numeric values encoded in text.

To avoid clashes of stanza names, a stanza name will be prefixed with the name of the company or organization that has defined the stanza. The company or organization name and the stanza name are separated by the character ":" (EBCDIC $7A_{16}$ or ASCII $3A_{16}$). Examples are ((SEG: Location Data)) and ((JJ Example Seismic: Microseismic Geometry Definition)). The company or organization name can be an abbreviation or acronym; but the name must be sufficiently unique so as to unambiguously identify the originator of the stanza definition. If there is any question that the name may become non-unique, the first stanza keyword/value pair should be "Stanza Definer = Full Company Name".

All stanza names should be uniquely associated with a single parameter set, typically keyword/value pairs.[8] To ensure that there is always a unique association between the stanza names and the stanza content, revision numbering and/or stanza name modification should be employed for all user defined stanzas.

For stanza naming, the Society of Exploration Geophysicists reserves the acronym SEG and all variants of SEG for use by the SEG Technical Standards Committee.

A SEG-Y reader must be capable of ignoring stanzas that the reader does not comprehend (which may be the whole Extended Textual File Header). The data within stanzas will typically use keywords and values, which can be produced and read by machines, as well as remaining human-readable.

Possible user supplied stanzas which have been suggested are:

- General Data Parameters (e.g. License Block, Date, Operator, Line etc.)
- General Acquisition Parameters
- SP to CDP relationship
- Usage of Optional parts of Trace Headers
- Decoded Binary Header

It is strongly recommended that the SEG-Y format be used principally to exchange seismic data. As part of that exchange, the SEG-Y file should contain sufficient information to identify the seismic data contained within the file and allow that seismic data to be processed. *The SEG-Y file is not intended as a comprehensive ancillary data exchange format.* The Extended Textual Header provides a means to include almost unlimited ancillary data in the SEG-Y file; but restraint should be exercised when selecting ancillary data to be included in the Extended Textual File Header. If significant amounts of ancillary data need to be exchanged, it is recommended that SEG Ancillary Data Standard data set(s) be used.

6.1. Structure of Extended Textual Header

The Extended Textual File Header consists of one or more 3200-byte records. Typically each record contains lines[9] of textual card-image text. Note that, unlike the Textual File Header, lines in the Extended Textual File Header do not start with the character "C" (EBCDIC $C3_{16}$ or ASCII 43_{16}). For processing purposes, all of the Extended Textual File Header records shall be considered as being concatenated into a single logical file (i.e., the inter-record gaps between the 3200-byte records are not significant).

Text within the Extended Textual File Header is organized into stanzas. A stanza begins on a 3200-byte boundary with a stanza header, which is a line containing the

[8] The User Data stanza allows XML 1.0 format in Extended Textual Headers, permitting complex names and values. OGP stanzas adhere to OGP formatting.

[9] A "line" is terminated by carriage return and linefeed characters. It need not be 80 characters long.

name of the defining organization or company and the name of the stanza. A stanza ends with the start of a new stanza, or the end of the Extended Textual File Header and consists of one or more 3200-byte records. The stanza header begins with double left parentheses "((", EBCDIC $4D_{16}$ or ASCII 28_{16} characters, and ends with double right parentheses "))", EBCDIC $5D_{16}$ or ASCII 29_{16} characters. The first left parenthesis at the beginning of a stanza must be in column one. The case of stanza names shall not be significant. To aid readability, spaces (" ", EBCDIC 40_{16} or ASCII 20_{16}) within stanza names shall be allowed but ignored. Thus the stanza name ((SEG: Recording Parameters)) shall refer to the same stanza as ((seg:RECORDINGPARAMETERS)).

The format of the information within a stanza depends on the type of the data contained in the stanza, which is implicitly and uniquely defined by the name of the stanza. However, many stanzas will contain data organized as keyword/value pairs. 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.
- For lines that are not comments, if the last non-blank character on the 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). Note that blank lines and comments are bypassed when continuing the line.
- Each line in an Extended Textual File Header ends in carriage return and linefeed (EBCDIC $0D25_{16}$ or ASCII $0D0A_{16}$)

6.2. EndText stanza

The EndText stanza is required[10] if the Binary File Header value in bytes 3505–3506 is –1. If that value is greater than zero, the EndText stanza is optional but must in that case be included in the count of Extended Textual Header records if present. The stanza ((SEG: EndText)) is treated specially with regard to stanza concatenation. This stanza must appear as the final 3200-byte record in the Extended Textual File Header. The stanza header shall be on the first line in the record and must be the only non-blank text in the record (i.e. the stanza must be empty). This allows the end of the Extended Textual File Header to be located easily by SEG-Y readers and simplifies decoding for SEG-Y readers that do not wish to process the Extended Textual File Header.

[10] In principle, the EndText stanza could be omitted when bytes 3521–3528 of the Binary File Header are nonzero, but we feel it best to continue to require it for SEG-Y rev 1 software compatibility.

6.3. Stanza Examples

((JJ ESeis: Microseismic Geometry Definition ver 1.0))
Definer name = J and J Example Seismic Ltd.
Line Name Convention = CDA
Line Name = Sample MicroSeismic 1
First Trace In Data Set = 101
Last Trace In Data Set = 1021
First SP In Data Set = 2001
Last SP In Data Set = 6032
… *additional blank lines to end of 3200-byte Extended Textual Header record*

((OGP: P1/11 Data Geographic Extent))

```
H1,5,0,0,Survey Perimeter Definition… ,1,Full Fold Boundary,2,1,3,Full Fold Coverage,0,
M1,0,1,1,1,1,391194.94,4092809.86,,54.2344345434,-9.2344345434,,
M1,0,1,1,2,1,392747.34,4093232.60,,54.2655123423,-9.2435354534,,
M1,0,1,1,3,1,393576.45,4094267.73,,54.2834225677,-9.2578834354,,
M1,0,1,1,4,1,391243.56,4095786.14,,54.2535353553,-9.2367002431,,
M1,0,1,1,1, ,391194.94,4092809.86,,54.2344345434,-9.2344345434,,
```

… *additional blank lines to end of 3200-byte Extended Textual Header record*

((SEG: Measurement Units ver 1.0))
Data Sample Measurement Unit =Millivolts
Volt conversion =0.001
… *additional blank lines to end of 3200-byte Extended Textual Header*

7. Data Traces

7.1. Trace Header

The SEG-Y trace header contains trace attributes, most of which are defined with two-byte or four-byte, two's complement integers. The values in bytes 1–180 were defined in the 1975 standard and these entries remain unchanged, although clarification and extensions may be supplied where appropriate. Bytes 181–240 were for optional information in the 1975 standard and this has been the main area of conflict between different flavors of SEG-Y.

SEG-Y rev 1 defined standard locations in bytes 181–232 of the Standard Trace Header for certain values that are needed in modern data processing. In particular, standard locations for a shotpoint number and ensemble (CDP) coordinates are defined. Bytes 203 to 210 allow the measurement units for the Data Trace samples to be defined and transduction constants to be specified. These entries allow the Data Trace values to be converted to engineering units.

The present revision includes an optional SEG-Y Trace Header Extension to provide extra precision and range for certain fields (coordinates, depths and elevations, time stamps, sample rate and number of samples per trace) in the Standard Trace Header. When used, any SEG-Y Trace Header Extensions must immediately follow the SEG-Y Standard Trace Header.

This revision further provides for zero or more User-defined Proprietary Trace Header Extensions to appear after SEG-defined trace headers. If more than zero User-defined Proprietary Trace Header Extensions are provided, SEG-Y Trace Header Extension 1 is mandatory. Also, unless the fixed length flag in Binary File Header bytes 3503–3504 is set, User-defined Proprietary Trace Header Extensions may appear in any order and number from trace to trace after all present SEG-defined trace headers. In the event the fixed length flag is set, the same sequence of trace headers and the same number and type of data samples must be present in every trace.

The values included in the SEG-defined Trace Headers are limited and intended to provide information that may change on a trace-by-trace basis and the basic information needed to process and identify the trace. The trace headers are not intended to be a repository for exceptional amounts of ancillary data. If great amounts of ancillary data need to be exchanged, it is recommended that one or more SEG Ancillary Data Standard datasets be used.

Table 3 Standard Trace Header[11]

240-byte Standard Trace Header	
Byte	**Description**
1–4	Trace sequence number within line — Numbers continue to increase if the same line continues across multiple SEG-Y files.
5–8	Trace sequence number within SEG-Y file — Each file starts with trace sequence one.
9–12	Original field record number.
13–16	Trace number within the original field record. If supplying multi-cable data with identical channel numbers on each cable, either supply the cable ID number in bytes 153–156 of SEG-Y Trace Header Extension 1 or enter (cable–1)*nchan_per_cable+channel_no here.
17–20	Energy source point number — Used when more than one record occurs at the same effective surface location. It is recommended that the new entry defined in Trace Header bytes 197–202 be used for shotpoint number.
21–24	Ensemble number (i.e. CDP, CMP, CRP, etc.)
25–28	Trace number within the ensemble — Each ensemble starts with trace number one.
29–30	Trace identification code: –1 = Other 0 = Unknown 1 = Time domain seismic data

[11] Where not otherwise indicated, a value of zero indicates an unknown or unspecified value.

240-byte Standard Trace Header	
Byte	**Description**
	2 = Dead 3 = Dummy 4 = Time break 5 = Uphole 6 = Sweep 7 = Timing 8 = Waterbreak 9 = Near-field gun signature 10 = Far-field gun signature 11 = Seismic pressure sensor 12 = Multicomponent seismic sensor – Vertical component 13 = Multicomponent seismic sensor – Cross-line component 14 = Multicomponent seismic sensor – In-line component 15 = Rotated multicomponent seismic sensor – Vertical component 16 = Rotated multicomponent seismic sensor – Transverse component 17 = Rotated multicomponent seismic sensor – Radial component 18 = Vibrator reaction mass 19 = Vibrator baseplate 20 = Vibrator estimated ground force 21 = Vibrator reference 22 = Time-velocity pairs 23 = Time-depth pairs 24 = Depth-velocity pairs 25 = Depth domain seismic data 26 = Gravity potential 27 = Electric field – Vertical component 28 = Electric field – Cross-line component 29 = Electric field – In-line component 30 = Rotated electric field – Vertical component 31 = Rotated electric field – Transverse component 32 = Rotated electric field – Radial component 33 = Magnetic field – Vertical component 34 = Magnetic field – Cross-line component 35 = Magnetic field – In-line component 36 = Rotated magnetic field – Vertical component 37 = Rotated magnetic field – Transverse component 38 = Rotated magnetic field – Radial component 39 = Rotational sensor – Pitch 40 = Rotational sensor – Roll 41 = Rotational sensor – Yaw 42 … 255 = Reserved 256 … N = optional use, (maximum N = 16,383) N+16,384 = Interpolated, i.e. not original, seismic trace.
31–32	Number of vertically summed traces yielding this trace. (1 is one trace, 2 is two summed traces, etc.)

<table>
<tr><th colspan="3">240-byte Standard Trace Header</th></tr>
<tr><th>Byte</th><th colspan="2">Description</th></tr>
<tr><td>33–34</td><td colspan="2">Number of horizontally stacked traces yielding this trace. (1 is one trace, 2 is two stacked traces, etc.)</td></tr>
<tr><td>35–36</td><td colspan="2">Data use:
1 = Production
2 = Test</td></tr>
<tr><td>37–40</td><td colspan="2">Distance from center of the source point to the center of the receiver group (negative if opposite to direction in which line is shot).</td></tr>
<tr><td>41–44</td><td>Elevation of receiver group. This is, of course, normally equal to or lower than the surface elevation at the group location.</td><td rowspan="7">The scalar in Trace Header bytes 69–70 applies to these values.
The units are feet or meters as specified in Binary File Header bytes 3255–3256.
Elevations and depths and their signs (+ve or –ve) are tied to a vertical CRS defined through an Extended Textual Header (see Appendix D-1).
Historical usage had been that all elevations above the vertical datum were positive and below were negative. Elevations should now be defined with respect to the CRS.</td></tr>
<tr><td>45–48</td><td>Surface elevation at source location.</td></tr>
<tr><td>49–52</td><td>Source depth below surface[12].</td></tr>
<tr><td>53–56</td><td>Seismic Datum[13] elevation at receiver group. (If different from the survey vertical datum, Seismic Datum should be defined through a vertical CRS in an extended textual stanza.)</td></tr>
<tr><td>57–60</td><td>Seismic Datum elevation at source. (As above)</td></tr>
<tr><td>61–64</td><td>Water column height at source location (at time of source event).</td></tr>
<tr><td>65– 68</td><td>Water column height at receiver group location (at time of recording of first source event into that receiver).</td></tr>
<tr><td>69–70</td><td colspan="2">Scalar to be applied to all elevations and depths specified in Standard Trace Header bytes 41–68 to give the real value. Scalar = 1, ±10, ±100, ±1000, or ±10,000. If positive, scalar is used as a multiplier; if negative, scalar is used as a divisor. A value of zero is assumed to be a scalar value of 1.</td></tr>
</table>

[12] Historically, taken as positive below the surface. We also note that bytes 41–48 of SEG-Y Trace Header Extension 1 provide a method for calculating receiver depth below the surface.

[13] Typically a floating or flat seismic processing datum that has been used in preprocessing the data in this SEG-Y dataset.

240-byte Standard Trace Header		
Byte	**Description**	
71–72	Scalar to be applied to all coordinates specified in Standard Trace Header bytes 73–88 and to bytes Trace Header 181–188 to give the real value. Scalar = 1, ±10, ±100, ±1000, or ±10,000. If positive, scalar is used as a multiplier; if negative, scalar is used as divisor. A value of zero is assumed to be a scalar value of 1.	
73–76	Source coordinate – X.	The coordinate reference system should be identified through an Extended Textual Header (see Appendix D-1). If the coordinate units are in seconds of arc, decimal degrees or DMS, the X values represent longitude and the Y values latitude. A positive value designates east of Greenwich Meridian or north of the equator and a negative value designates south or west.
77–80	Source coordinate – Y.	
81–84	Group coordinate – X.	
85–88	Group coordinate – Y.	
89–90	Coordinate units: 1 = Length (meters or feet as specified in Binary File Header bytes 3255-3256 and in Extended Textual Header if Location Data are included in the file) 2 = Seconds of arc (deprecated) 3 = Decimal degrees (preferred degree representation) 4 = Degrees, minutes, seconds (DMS)[14] Note: To encode ±DDDMMSS set bytes 73–88 = $\pm DDD*10^4 + MM*10^2 + SS$ with bytes 71–72 set to 1; To encode ±DDDMMSS.ss set bytes 73–88 = $\pm DDD*10^6 + MM*10^4 + SS*10^2 + ss$ with bytes 71–72 set to –100.	
91–92	Weathering velocity. (ft/s or m/s as specified in Binary File Header bytes 3255–3256)	
93–94	Subweathering velocity. (ft/s or m/s as specified in Binary File Header bytes 3255–3256)	
95–96	Uphole time at source in milliseconds.	Time in milliseconds as scaled by the scalar specified in Standard Trace Header bytes 215-216.
97–98	Uphole time at group in milliseconds.	
99–100	Source static correction in milliseconds.	
101–102	Group static correction in milliseconds.	
103–104	Total static applied in milliseconds. (Zero if no static has been applied,)	

[14] With two decimal places (±DDMMSS.ss) resolution is approximately ±0.3 meters. If longitudes are in the range ±180 degrees, a third decimal place is available.

240-byte Standard Trace Header		
Byte	**Description**	
105–106	Lag time A — Time in milliseconds between end of 240-byte trace identification header and time break. The value is positive if time break occurs after the end of header; negative if time break occurs before the end of header. Time break is defined as the initiation pulse that may be recorded on an auxiliary trace or as otherwise specified by the recording system.	
107–108	Lag Time B — Time in milliseconds between time break and the initiation time of the energy source. May be positive or negative.	
109–110	Delay recording time — Time in milliseconds between initiation time of energy source and the time when recording of data samples begins. In SEG-Y rev 0 this entry was intended for deep-water work if data recording did not start at zero time. The entry can be negative to accommodate negative start times (i.e. data recorded before time zero, presumably as a result of static application to the data trace). If a non-zero value (negative or positive) is recorded in this entry, a comment to that effect should appear in the Textual File Header.	
111–112	Mute time — Start time in milliseconds.	
113–114	Mute time — End time in milliseconds.	
115–116	Number of samples in this trace. The number of bytes in a trace record must be consistent with the number of samples written in the Binary File Header and/or the SEG-defined Trace Header(s). This is important for all recording media; but it is particularly crucial for the correct processing of SEG-Y data in disk files (see Appendix A). If the fixed length trace flag in bytes 3503–3504 of the Binary File Header is set, the number of samples in every trace in the SEG-Y file is assumed to be the same as the value recorded in the Binary File Header and this field is ignored. If the fixed length trace flag is not set, the number of samples may vary from trace to trace.	
117–118	Sample interval for this trace. Microseconds (µs) for time data, Hertz (Hz) for frequency data, meters (m) or feet (ft) for depth data. If the fixed length trace flag in bytes 3503–3504 of the Binary File Header is set, the sample interval in every trace in the SEG-Y file is assumed to be the same as the value recorded in the Binary File Header and this field is ignored. If the fixed length trace flag is not set, the sample interval may vary from trace to trace.	

240-byte Standard Trace Header	
Byte	**Description**
119–120	Gain type of field instruments: 1 = fixed 2 = binary 3 = floating point 4 … N = optional use
121–122	Instrument gain constant (dB).
123–124	Instrument early or initial gain (dB).
125–126	Correlated: 1 = no 2 = yes
127–128	Sweep frequency at start (Hz).
129–130	Sweep frequency at end (Hz).
131–132	Sweep length in milliseconds.
133–134	Sweep type: 1 = linear 2 = parabolic 3 = exponential 4 = other
135–136	Sweep trace taper length at start in milliseconds.
137–138	Sweep trace taper length at end in milliseconds.
139–140	Taper type: 1 = linear 2 = $\cos^2$ 3 = other
141–142	Alias filter frequency (Hz), if used.
143–144	Alias filter slope (dB/octave).
145–146	Notch filter frequency (Hz), if used.
147–148	Notch filter slope (dB/octave).
149–150	Low-cut frequency (Hz), if used.
151–152	High-cut frequency (Hz), if used.
153–154	Low-cut slope (dB/octave)
155–156	High-cut slope (dB/octave)

240-byte Standard Trace Header	
Byte	**Description**
157–158	Year data recorded — The 1975 standard was unclear as to whether this should be recorded as a 2-digit or a 4-digit year and both have been used. For SEG-Y revisions beyond rev 0, the year should be recorded as the complete 4-digit Gregorian calendar year, e.g., the year 2001 should be recorded as 2001_{10} ($07D1_{16}$).
159–160	Day of year[15] (Range 1–366 for GMT, UTC, and GPS time basis).
161–162	Hour of day (24 hour clock).
163–164	Minute of hour.
165–166	Second of minute.
167–168	Time basis code. If nonzero, overrides Binary File Header bytes 3511–3512. 1 = Local 2 = GMT (Greenwich Mean Time) 3 = Other, should be explained in a user defined stanza in the Extended Textual File Header 4 = UTC (Coordinated Universal Time) 5 = GPS (Global Positioning System Time)
169–170	Trace weighting factor — Defined as 2^{-N} units (volts unless bytes 203–204 specify a different unit) for the least significant bit. (N = 0, 1, …, 32767)
171–172	Geophone group number of roll switch position one.
173–174	Geophone group number of trace number one within original field record.
175–176	Geophone group number of last trace within original field record.
177–178	Gap size (total number of groups dropped).
179–180	Over travel associated with taper at beginning or end of line: 1 = down (or behind) 2 = up (or ahead)
181–184	X coordinate of ensemble (CDP) position of this trace (scalar in Standard Trace Header bytes 71–72 applies). The coordinate reference system should be identified through an Extended Textual Header (see Appendices D-1 or D-3).
185–188	Y coordinate of ensemble (CDP) position of this trace (scalar in Standard Trace Header bytes 71–72 applies). The coordinate reference system should be identified through an Extended Textual Header (see Appendices D-1 or D-3).
189–192	For 3-D poststack data, this field should be used for the in-line number. If one in-line per SEG-Y file is being recorded, this value should be the same for all traces in the file and the same value will be recorded in bytes 3205–3208 of the Binary File Header.

[15] Also known as Julian Day, albeit distinguished from the Julian Day in the Julian Date system that astronomers use.

240-byte Standard Trace Header	
Byte	**Description**
193–196	For 3-D poststack data, this field should be used for the cross-line number. This will typically be the same value as the ensemble (CDP) number in Standard Trace Header bytes 21–24, but this does not have to be the case.
197–200	Shotpoint number — This is probably only applicable to 2-D poststack data. Note that it is assumed that the shotpoint number refers to the source location nearest to the ensemble (CDP) location for a particular trace. If this is not the case, there should be a comment in the Textual File Header explaining what the shotpoint number actually refers to.
201–202	Scalar to be applied to the shotpoint number in Standard Trace Header bytes 197–200 to give the real value. If positive, scalar is used as a multiplier; if negative as a divisor; if zero the shotpoint number is not scaled (i.e. it is an integer. A typical value will be –10, allowing shotpoint numbers with one decimal digit to the right of the decimal point).
203–204	Trace value measurement unit: –1 = Other (should be described in Data Sample Measurement Units Stanza) 0 = Unknown 1 = Pascal (Pa) 2 = Volts (v) 3 = Millivolts (mV) 4 = Amperes (A) 5 = Meters (m) 6 = Meters per second (m/s) 7 = Meters per second squared (m/s^2) 8 = Newton (N) 9 = Watt (W) 10-255 = reserved for future use 256 … N = optional use. (maximum N = 32,767)
205–210	Transduction Constant — The multiplicative constant used to convert the Data Trace samples to the Transduction Units (specified in Standard Trace Header bytes 211–212). The constant is encoded as a four-byte, two's complement integer (bytes 205–208) which is the mantissa and a two-byte, two's complement integer (bytes 209–210) which is the power of ten exponent (i.e. Bytes 205–208 * 10**Bytes 209–210).
211–212	Transduction Units — The unit of measurement of the Data Trace samples after they have been multiplied by the Transduction Constant specified in Standard Trace Header bytes 205–210. –1 = Other (should be described in Data Sample Measurement Unit stanza, page 114) 0 = Unknown 1 = Pascal (Pa) 2 = Volts (v) 3 = Millivolts (mV) 4 = Amperes (A) 5 = Meters (m)

240-byte Standard Trace Header	
Byte	**Description**
	6 = Meters per second (m/s) 7 = Meters per second squared (m/s^2) 8 = Newton (N) 9 = Watt (W)
213–214	Device/Trace Identifier — The unit number or id number of the device associated with the Data Trace (i.e. 4368 for vibrator serial number 4368 or 20316 for gun 16 on string 3 on vessel 2). This field allows traces to be associated across trace ensembles independently of the trace number (Standard Trace Header bytes 25–28).
215–216	Scalar to be applied to times specified in Trace Header bytes 95–114 to give the true time value in milliseconds. Scalar = 1, +10, +100, +1000, or +10,000. If positive, scalar is used as a multiplier; if negative, scalar is used as divisor. A value of zero is assumed to be a scalar value of 1.
217–218	Source Type/Orientation — Defines the type and the orientation of the energy source. The terms vertical, cross-line and in-line refer to the three axes of an orthogonal coordinate system. The absolute azimuthal orientation of the coordinate system axes should be defined in the CRS (see Appendix D-1). –1 to –n = Other (should be described in Source Type/Orientation stanza, page 115) 0 = Unknown 1 = Vibratory - Vertical orientation 2 = Vibratory - Cross-line orientation 3 = Vibratory - In-line orientation 4 = Impulsive - Vertical orientation 5 = Impulsive - Cross-line orientation 6 = Impulsive - In-line orientation 7 = Distributed Impulsive - Vertical orientation 8 = Distributed Impulsive - Cross-line orientation 9 = Distributed Impulsive - In-line orientation
219–224	Source Energy Direction with respect to the source orientation — Three two-byte two's complement binary integers for vertical, cross-line and in-line inclinations respectively. The positive orientation direction is defined in Bytes 217–218 of the Standard Trace Header. The energy direction is encoded in tenths of degrees (i.e. 347.8º is encoded as 3478_{10} ($0D96_{16}$)).

240-byte Standard Trace Header	
Byte	**Description**
225–230	Source Measurement — Describes the source effort used to generate the trace. The measurement can be simple, qualitative measurements such as the total weight of explosive used or the peak air gun pressure or the number of vibrators times the sweep duration. Although these simple measurements are acceptable, it is preferable to use true measurement units of energy or work. The constant is encoded as a four-byte, two's complement integer (bytes 225–228) which is the mantissa and a two-byte, two's complement integer (bytes 209–230) which is the power of ten exponent (i.e. Bytes 225–228 * 10**Bytes 229–230).
231–232	Source Measurement Unit — The unit used for the Source Measurement, Standard Trace header bytes 225–230. –1 = Other (should be described in Source Measurement Unit stanza, page 116) 0 = Unknown 1 = Joule (J) 2 = Kilowatt (kW) 3 = Pascal (Pa) 4 = Bar (Bar) 4 = Bar-meter (Bar-m) 5 = Newton (N) 6 = Kilograms (kg)
233–240	Either binary zeros or the eight character trace header name "SEG00000". May be ASCII or EBCDIC text.

Table 4 details the contents of SEG-Y Trace Header Extension 1. This extension allows one to override or supplement entries in the SEG-Y Standard Trace Header, providing more numeric precision or additional information. If used, it should appear immediately after the SEG-Y Standard Trace Header. In the event that any Proprietary User-defined Trace Headers are present, then SEG-Y Trace Header Extension 1 bytes 157-158 are used to account for their number.

Table 4 Trace Header Extension 1[16]

240-byte SEG-Y Trace Header Extension 1	
Byte	**Description**
1–8	Extended trace sequence number within line — Numbers continue to increase if the same line continues across multiple SEG-Y files. If nonzero, overrides trace sequence number within line (SEG-Y Standard Trace Header bytes 1–4). 64 bit unsigned integer.
9–16	Extended trace sequence number within SEG-Y file — Each file starts with trace sequence one. If nonzero, overrides trace sequence number within SEG-Y file (SEG-Y Standard Trace Header bytes 5–8). 64 bit unsigned integer.

[16] Where not otherwise indicated, a value of zero indicates an unknown or unspecified value.

240-byte SEG-Y Trace Header Extension 1	
Byte	**Description**
17–24	Extended original field record number. If nonzero, overrides original field record number (SEG-Y Standard Trace Header bytes 9–12). 64 bit two's-complement integer.
25–32	Extended ensemble number (i.e. CDP, CMP, CRP, etc.) If nonzero, overrides ensemble number (SEG-Y Standard Trace Header bytes 21–24). 64 bit two's-complement integer.
33–40	Extended elevation of receiver group. If nonzero, overrides receiver group elevation (SEG-Y Standard Trace Header bytes 41–44). IEEE double precision (64-bit) value.
41–48	Receiver group depth below the surface location of receiver group. Positive for receiver group below surface[17]. IEEE double precision (64-bit) value.
49–56	Extended surface elevation at source location. If nonzero, overrides surface elevation at source location (SEG-Y Standard Trace Header bytes 45–48). IEEE double precision (64-bit) value.
57-64	Extended source depth below surface. If nonzero, overrides source depth below surface (SEG-Y Standard Trace Header bytes 49-52). IEEE double precision (64-bit) value.
65-72	Extended Seismic Datum elevation at receiver group. If nonzero, overrides Seismic Datum elevation at receiver group (SEG-Y Standard Trace Header bytes 53–56). IEEE double precision (64-bit) value.
73-80	Extended Seismic Datum elevation at source. If nonzero, overrides Seismic Datum elevation at source (SEG-Y Standard Trace Header bytes 57–60). IEEE double precision (64-bit) value.
81-88	Extended water column height at source location (at time of source event). If nonzero, overrides water column height at source location (SEG-Y Standard Trace Header bytes 61–64). IEEE double precision (64-bit) value.
89-96	Extended water column height at receiver group location (at time of recording of first source event into that receiver). If nonzero, overrides water column height at receiver group location (SEG-Y Standard Trace Header bytes 65–68). IEEE double precision (64-bit) value.
	For values in bytes 33–96, signs of elevation or depth values (+ve or –ve) must be in accordance with the respective vertical CRS defined through a Location Data Stanza (see Appendices D-1 or D-3).
97-104	Extended source coordinate - X. If nonzero, overrides Source coordinate - X (SEG-Y Standard Trace Header bytes 73–76). Will be negative for negative coordinates. IEEE double precision (64-bit) value.

[17] Note that there is no corresponding entry in the SEG-Y Standard Trace Header.

240-byte SEG-Y Trace Header Extension 1	
Byte	**Description**
105-112	Extended Source coordinate - Y. If nonzero, overrides Source coordinate - Y (SEG-Y Standard Trace Header bytes 77–80). Will be negative for negative coordinates. IEEE double precision (64-bit) value.
113-120	Extended group coordinate - X. If nonzero, overrides group coordinate - X (SEG-Y Standard Trace Header bytes 81–84). Will be negative for negative coordinates. IEEE double precision (64-bit) value.
121-128	Extended group coordinate - Y. If nonzero, overrides group coordinate - Y (SEG-Y Standard Trace Header bytes 85–88). Will be negative for negative coordinates. IEEE double precision (64-bit) value.
129-136	Extended Distance from center of the source point to the center of the receiver group (negative if opposite to direction in which line is shot). If nonzero, overrides Distance from center of the source point to the center of the receiver group (SEG-Y Standard Trace Header bytes 37-40). IEEE double precision (64-bit) value.
137-140	Extended number of samples in this trace. If nonzero, overrides number of samples in this trace (SEG-Y Standard Trace Header bytes 115–116). 4 byte unsigned integer value.
141-144	Nanoseconds to add to Second of minute (SEG-Y Standard Trace Header bytes 165–166). May be negative. 4 byte signed integer value.
145-152	If nonzero, IEEE double precision (64-bit) microsecond sample interval, overriding bytes 117–118 of the Standard Trace Header.
153-156	Cable number for multi-cable acquisition or Recording Device/Sensor ID number. 4 byte signed integer value.
157-158	Number of additional trace header extension blocks including this one. If zero, the value in the Binary File Header bytes 3507–3510 is assumed. 2 byte unsigned integer value.
159-160	Last trace flag — a sum of the appropriate integers, leave zero otherwise. 1 = Last trace in ensemble (CDP, Shot Record, …) 2 = Last trace in line 4 = Last trace in this data file or data stream 8 = Last trace in current survey
161–168	Extended X coordinate of ensemble (CDP) position of this trace. If nonzero, overrides X coordinate of ensemble (CDP) (SEG-Y Standard Trace Header bytes 181–184). Will be negative for negative coordinates. IEEE double precision (64-bit) value.
169–176	Extended Y coordinate of ensemble (CDP) position of this trace. If nonzero, overrides Y coordinate of ensemble (CDP) (SEG-Y Standard Trace Header bytes 185–188). Will be negative for negative coordinates. IEEE double precision (64-bit) value.
177–232	Reserved

240-byte SEG-Y Trace Header Extension 1	
Byte	**Description**
233–240	Eight character trace header name “SEG00001”. May be ASCII or EBCDIC text.

All nonstandard 240-byte trace header extensions must take the following form. A corresponding record or records defining the proprietary layout should always be included in a SEG-Y Extended Textual Header ((SEG: Trace Header Mapping)) and include the trace header extension name in its stanza.

Table 5 Proprietary Trace Header Extension

240-byte Proprietary Trace Header Extensions	
Byte	**Description**
1–232	User defined
233–240	Eight character trace header extension name, left justified and blank padded. May be ASCII or EBCDIC text and must consist of printable characters in the selected character set. (Refer to Appendix F, p. 143.) Names “SEG00000” through “SEG99999” are reserved.

7.2. Trace Data

Trace Data immediately follow their attached Trace Header(s), with the trace data arranged in samples of fixed size (1, 2, 3, 4, or 8 bytes) described in Appendix E. The format of the data sample is specified in the Binary File Header (bytes 3225–3226). With SEG-Y revision 2, provision has been made via bytes 3297–3300 of the Binary File Header to consistently support little-endian byte ordering or pairwise byte swapping of the Binary File Header and both Trace Headers and Trace Data.

The seismic data in a SEG-Y file is organized into ensembles of traces or as a series of stacked traces. When the trace data is organized into ensembles of traces, the ensemble type may be identified (Binary File Header bytes 3229–3230).

8. User Header stanza and Data Trailer

The User Header and Data Trailer allow for arbitrary binary or textual data within a SEG-Y file. The User Header is provided as an Extended Textual Header stanza option and the Data Trailer follows the last trace.

The User Header and Data Trailer use the same basic format, and consist of a set of user-defined blocks separated by a description block identifying the producer, contents, size and data format of the data block following it.

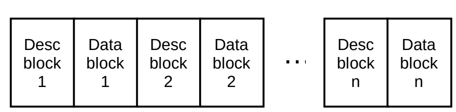

Figure 2 Structure of a User Header or Data Trailer containing *n* data blocks.

Even though the format is the same except for the preceding ((SEG:UserHeader)) stanza line, we recommended the User Header is used for information that is required for interpreting the data traces. Large data blocks like backup of positioning data, databases etc. should be stored in the Data Trailer. This will simplify decoding and speed up access to the data.

The description block is a well-formed XML document with a strictly defined format providing information about the following data block. It must be written in ASCII text (1 byte characters), and contain English text. This is done to simplify decoding of the description block on all platforms and parsing in any language by all readers of the SEG-Y format. Please refer to D-11 for a detailed description of the description block.

The data block following the description block can be of any format and have any contents in any language (binary, text, Unicode, big-endian, little-endian etc.)

The format is designed to make it easy to append and insert new blocks, and delete existing blocks. This allows systems to add information to the record as processing stages refine and update the data.

Examples of data that can be stored in the trailer are information about survey, contract, processing system, edits, trace data description, deliverables, processing notes - decisions and result evaluation, etc.

The User Header and Data Trailer can also be used as a backup of information like positioning files (P1, P6, SPS etc.) or equipment/sensor information (from e.g. SEG D).

They also allow bundling of any meta data with the seismic traces/cubes (e.g. database files, velocity models, processing scripts, observer logs). This can be used to automate and simplify the workflows, and improve consistency, correctness, robustness and repeatability of the seismic processing and interpretation.

Please refer to D-11 for more detailed description of the User header and Data trailer.

Appendix A. Writing SEG-Y Data to a Disk File

On modern UNIX and PC systems, a disk file is defined at the operating system level as a byte stream without any structure. It has become common practice for SEG-Y data to be streamed into a disk file, without any kind of encapsulation to recover record boundaries. Such a disk file can only be read by software that comprehends the SEG-Y format, since it must use certain values in the SEG-Y headers to reconstruct the original record stream. This appendix describes the rules that must be followed when un-encapsulated SEG-Y data is written to a disk file or network byte stream.

The first 3600 bytes of the file[18] are the "traditional" SEG-Y File Header (i.e. the 3200-byte Textual File Header followed by the 400-byte Binary Header). The Binary Header may be followed by zero or more 3200-byte Extended Textual File Header records, as indicated in bytes 3505–3506 of the Binary Header.

The first Data Trace record, beginning with the 240-byte SEG-Y Standard Trace Header, immediately follows the Binary File Header or if supplied, the last Extended Textual File Header. The number of bytes of Data Trace sample values that follow the Trace Header is determined from the value for number of samples in bytes 115–116 in the Trace Header and, if present, bytes 129–132 of SEG-Y Trace Header Extension 1, together with the sample format code in bytes 3225–3226 of the Binary Header. For

[18] Beware that a tape label may have been copied in front if the Textual File Header.

format codes 1, 2, 4, 5 and 10, the number of bytes of sample data is four times the number of samples. For format codes 6, 9 and 12, the number of bytes of sample data is eight times the number of samples. For format codes 3 and 11, the number of bytes of sample data is twice the number of samples. For format codes 7 and 15, the number of bytes of sample data is three times the number of samples. For format codes 8 and 16, the number of bytes of sample data is the same as the number of samples.

The Trace Header for the second Data Trace in the file follows immediately after the sample data for the first trace and so on for subsequent traces in the file.

As with tape, all values may be written to the disk file using either the traditional “big-endian” byte ordering or the “little-endian” byte ordering now overwhelmingly used in modern computer architectures. For exchange purposes text in the Textual File Header and Extended Textual File Headers may be written in EBCDIC or ASCII (UTF-8) character code.

Appendix B. SEG-Y Tape Labels

In order to bring SEG-Y into line with SEG D Rev 3.0, a label may, and should, be written at the front of a SEG-Y file on record-oriented removable media such as magnetic tape. This is a single record consisting of 128 bytes of ASCII characters, the same length as the SEG D label and sharing a similar format description. A SEG-Y tape label is optional and is only valid on SEG-Y files written to unformatted, removable media. However, a label must be present if the blocking scheme described in Appendix C is being used. In this case the label must appear as a separate 128-byte record at the beginning of the file. There must be no file mark between the label record and the first data record.

If the recording medium supports multiple partitions, each partition is treated in isolation as if it were a separate unit. Thus, if labels are being used, each partition must begin with a label. Data from one partition cannot "run-over" into a subsequent partition. Each partition must be capable of being decoded in isolation. On one recording medium, it is permissible to mix partitions containing SEG-Y data with partitions containing non-SEG-Y formatted information.

The format of a SEG-Y Tape Label is summarized in Table 6.

Table 6 SEG-Y Tape Label

Field	Description	Bytes	Start - end byte
1	Storage Unit Sequence Number	4	1 – 4
2	SEG-Y 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

Field	Description	Bytes	Start - end byte
6	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	Reserved	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 right-most 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-Y Revision field indicates which revision of SEG-Y was used to record the data on this tape. SY2.0 indicates that the data was formatted using SEG-Y Revision 2. *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-Y Revision 2 tapes, this field must contain "RECORD". *This field is required.*

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

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[19] used to describe the physical binding of the logical format to the storage unit. 'B2' is an appropriate value. ('B1' requires a file mark after the label.) *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.*

[19] w3.energistics.org/RP66/V2/rp66v2.html

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 is required.*

Organization codes are assigned by Energistics, formerly the Petrotechnical Open Standards Consortium (POSC), which maintains the current list of codes. Please refer to Appendix C of SEG D rev 3.0 for a list of the currently assigned codes. To request a new organization code, contact:

Energistics
1 Sugar Creek Center Blvd Ste 1075
Sugar Land, TX 77478-3671
USA
+1 281 243-2121 telephone
+1 281 243-2132 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 identifier 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. This field may be empty (i.e. may contain all blanks, in which case no serial number is specified).

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

Field 10
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. It is subdivided into four individual fields.

Field 11
The External Label Name is nonblank when the tape has an external label. The characters in this field are right justified with leading blank characters as required. 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.

Field 12
The Recording Entity Name should provide the originator of these SEG-Y data, whether from field recording or subsequent processing. Leave blank if unknown.

Field 13
The User defined field may contain any other user input information. The only restriction is that the data must be in ASCII.

Field 14
Reserved and should be recorded as all blanks (code 32_{10}).

Appendix C. Blocking of SEG-Y Files on Tape

This appendix describes a simple blocking scheme for writing SEG-Y files to tape. This is effectively a special encapsulation layer for SEG-Y and may be advantageous with certain tape devices that require a large block size, e.g. megabytes, to use the tape efficiently. Note however that this is not the only way to achieve SEG-Y file blocking and it may be preferable to use another encapsulation scheme such as SEG RODE.

In the following explanation, a SEG-Y record means a record defined in the SEG-Y standard (i.e. a 3200-byte Textual File Header record, a 400-byte Binary Header record, a 3200-byte Extended Textual File Header record or the Trace Header with its associated Data Trace). A tape record means a variable length physical record written to the tape device.

A tape containing SEG-Y data written using this blocking scheme must begin with a SEG-Y tape label, as described in Appendix B. The label must be written as a separate tape record 128 bytes long. If the tape medium supports partitioning, each partition is treated in isolation and must have its own label.

Each subsequent tape record may contain one or more SEG-Y records concatenated together. Each tape record must contain an integral number of SEG-Y records (i.e. the start of a tape record must coincide with the start of a SEG-Y record). The first tape record following the SEG-Y Tape Label must begin with the 3200-byte Textual File Header record. For all tape records in a file, the record length must be less than or equal to the maximum record length for the tape medium being used.

A SEG-Y reader program that comprehends this blocking scheme must unblock the data to reproduce the original SEG-Y record stream. In particular, it must use the number of samples in each trace and the sample format to determine the actual length of the trace record.

When this blocking scheme is being used, it is permitted to end one SEG-Y file and start a new one either with or without an intervening file mark. If a file mark is present, it signifies the end of a SEG-Y file and a file mark must be followed by either a tape record beginning with a 3200-byte Textual File Header record or another file mark. Alternatively, a new SEG-Y file can be identified by the start of a 3200-byte Textual File Header record. The Textual File Header would begin with a ‘C’ character ($C3_{16}$ in EBCDIC or ASCII 43_{16}), which is taken as the beginning of the new SEG-Y data set and have the SEG-Y revision level encoded in record C39 as described in section 4. It follows that the start of a SEG-Y file must start on a tape record boundary (i.e. any tape record contains data from only one SEG-Y file). In either case, a double file mark signifies end of data.

Appendix D. Extended Textual Stanzas

The structure for Extended Textual stanzas is described in section 6.1. The following stanzas are SEG-defined, standard stanzas. User defined stanzas are permitted and provide a means to logically extend and customize the SEG-Y format to a user's particular needs. It is highly advisable to use standard SEG defined or SEG approved stanza definitions. When additional information is required beyond the standard definitions, a user defined stanza can be used to extend the standard stanza without repeating the information contained in the standard stanza.

All parameters defined in a given stanza are required unless the stanza definition specifically notes that the inclusion of a parameter is optional. Byte reordering does not apply to the Extended Textual Header.

Location Data Stanzas

Appendices D-1 through D-5 are stanzas for Location Data. Revision 2 of the SEG-Y format supports the International Association of Oil and Gas Producers (IOGP) P1/11 and P6/11 formats for defining location data. Legacy formats for location data continue to be included in Revision 2 but are deprecated in favour of the IOGP formats. Location data stanzas are detailed in the following appendices:

D-1: Coordinate Reference System definition by P1/11 (including option for Data Geographic Extent definition)

D-2: Seismic Bin Grid definition by P6/11 (including option for survey Coverage Perimeter definition)

D-3: Coordinate Reference System definition by Extended Textual Stanzas (deprecated)

D-4: Bin Grid definition by Extended Textual Stanzas (deprecated)

D-5: Data Geographic Extent & Coverage Perimeter by Extended textual Stanzas (deprecated)

D-1. Coordinate Reference System Definition: International Association of Oil and Gas Producers P1/11

The OGP P1/11 Geophysical position data exchange format is preferred for defining location data and data geographic extent. It can also be used interchangeably with OGP P6/11 for defining a survey coverage perimeter, although for a 3D survey this should more logically appear in OGP P6/11.

The stanza for OGP P1/11 Coordinate Reference System (CRS) definition takes the simple form of an initial

((OGP:P1/11 CRS))

followed by the text of the OGP P1/11 CRS definition, with sufficient trailing space characters to make the stanza a multiple of 3200 bytes. The stanza identifies the CRS to which source, receiver group and CDP coordinates given in the Standard Trace Header and Header Extension are referenced. The SEG recommends using the OGP P1/11 version 1.1 standard when backward compatibility is not an issue. Note that the “keyword = value” structure is not used in the OGP formats.

The OGP P1/11 standard is maintained by the International Association of Oil and Gas Producers (IOGP, formerly OGP). The recommended method for defining the coordinate reference system (CRS) of location data is found in the Common Header section of the P1/11 standard. A description of the format can be found in this appendix. For further details and more information on usage please refer to the IOGP website at http://www.iogp.org/Geomatics#2521696-geophysical-operations and the Geomatics Committee page http://www.iogp.org/Geomatics. On the Geophysical Operations Sub-Committee page there is a P1/11 User Guide with more examples and details of usage.

The old CRS stanza format from SEG-Y rev 1 is still supported, but deprecated (see Appendix D-3). All new systems should use the format described in Appendix D-1 as it is computer readable, and ensures compatibility between modern formats.

D-1.1 Format overview

All SEG formats, and SEG-Y and SEG D in particular, aim to use the same CRS definition format, and will also share the definition text with IOGP P-formats to ensure compatibility throughout the seismic acquisition and processing.

The CRS identification blocks contain text like this:

```
((OGP:P1/11 CRS))
P-format CRS definition records (lines of text)
     :
((SEG:EndText))
```

The P-format CRS records are the same records as can be found in a P1/11 file. Table numbering in this appendix may not be sequential with the rest of this document but replicates table numbering in the P1/11 format description document.

After removing the ((OGP:P1/11 CRS)) and optional ((SEG:EndText)) strings, the SEG-Y CRS definition string can be passed directly into any P1/11 compatible position or processing package. To that end the CRS section should begin with the ‘OGP File Identification Record’ to enable format recognition by the reading device. (See section D-1.6 below).

The P1/11 CRS format supports lines longer than 80 characters. Also note the use of # (comment fields) and & (line breaks) used in the other SEG-Y stanzas are not allowed in the P-format. In P1/11 lines starting with ‘C’ are comments.

The string is broken up into multiples of 3200 bytes. If the string does not fit into multiples of 3200 byte blocks, the last block is padded with space (code 20_{16}).

The ((SEG: EndText)) may be omitted, but is included for compatibility reasons.

Note that the text in (( )) is case and white space insensitive, so ((OGP:P1/11 CRS)) may be written ((ogp:p1/11crs)).

D-1.2 Logical File Structure

The data is stored in a series of variable length ASCII comma-separated data records, each terminated by a carriage return (Hex 0x0D) and/or a line feed (Hex 0x0A) character. Line termination shall be consistent throughout each file.

As the format is designed primarily for access by a computer program, there is no fixed limit on the length of each individual data record, and many record definitions allow multiple data items to be written into a single record. However, while it is recommended that systems make use of this facility to reduce file size where it is possible to do so, it is also recommended that records should not be written to excessive length but should instead be split across multiple records.

Although the format is primarily intended for computer access, it is also common for the file to be visually inspected, particularly the Common Header records. Thus it is recommended that, particularly for the Common Header block, systems writing the files make use of spaces to pad any repeated records to ensure the data is aligned in columns to facilitate readability.

Thus, if possible, Common Header records should be written as:

```
HC,1,5,2,Latitude of natural origin                         ,1,8801,     0,3,degree
HC,1,5,2,Longitude of natural origin                        ,1,8802,   -15,3,degree
HC,1,5,2,Scale factor at natural origin                     ,1,8805,0.9996,4, unity
HC,1,5,2,False easting                                      ,1,8806,500000,1, metre
HC,1,5,2,False northing                                     ,1,8807,     0,1, metre
```

However it should be noted, unless the field width is specifically stated in the record field definition, this padding of records for readability is a recommendation and not an absolute requirement. Note that for recording in SEG-Y format, if the record (or record set) does not fit a multiple of 3200 byte blocks the last block should be padded with space (code 20_{16}).

D-1.3 Record Identifiers

The format defines that for most records the first comma-separated sections of each record contain the record identifying codes. The first section always contains two characters, which are used to identify the general record type. The first character identifies the type of record. Two common record types are defined across all formats, an “H” record indicates a header record and a “C” record indicates a comment record. (Other characters including “E”, “M”, “N”, “P”, “R”, “S”, “T”, “A” and “X” are used for data records.)

The second character indicates the data format:

2nd Character	Format Type
C	Common to all formats
1	Geophysical Position Data Exchange (P1/11)
2	Positioning Data Exchange (P2/11)
6	Seismic Bin Grid Data Exchange (P6/11)

Table 1 (D-1): Format Types

Thus “HC” is a header record common across P-formats (“Common Header”) and “R1” is a receiver position record from the P1/11 format.

All header records are identified by four comma-separated sections. (Data records are identified by two, three or four sections.) Where relevant, the remaining comma-separated sections contain numeric values which identify the record – thus for example record **HC,1,3,0** contains the coordinate reference system implicit identification.

D-1.4 Data Types used in the Format Definition

The following data types are used in the P1/11 format definition:

Name	Description	Conditions	Value
Single Items			
Integer	Integer Number		341234
Float	Floating Point Number		12.345678

Engineering	Engineering Format Floating Point Number		1.23456E+03
Text	Free Text	L J n: Specifies the text should be left justified to the minimum width specified	Hello World
Description	Record Description	A text field left justified to 50 characters	Project Name
Date	Date		YYYY:MM:DD
Time	Time		HH:MM:SS
Note: Time can be recorded to any number of decimal places, as defined by the data record			
Variant	Any of the above data types		
Lists (All of general format xx&xx&xx&xx)			
Integer List	List of Integer Numbers		12&34&56&78&9
Float List	List of Floats		1.23&4.56&6.78
Engineering List	List of Engineering		1.23456E3&7.89012E4&3.456E
Text List	List of Text		Hello&world

Table 2 (D-1): Format Data Types

For some fields the data type is given as “Variant”. This may take the form of any of the data types.

The codes used to define variant data stored within the data records are defined in Table 4 (D-1) below.

All individual text fields should contain only ASCII characters in the range 32 (Hex 0x20) to 126 (Hex 0x7E) and the following characters are additionally not to be used to ensure format rigidity:

Character	Description	ASCII Code	Usage in Format
,	Comma	44	Separates Fields
;	Semi Colon	59	Separates items in a Standard Record Extension Definition and Record Extension Field
:	Colon	58	Separates items in Date and Time fields
&	Ampersand	38	Separates items in a List

Table 3 (D-1): Reserved Characters

Where use of reserved characters is unavoidable, for example to refer to a parameter name exactly as used by its source, an escape character sequence can be used.

Any character can be expressed as a \u followed by the 4 digit hexadecimal value written in uppercase. For example:

```
,     is escaped with  \u002C
;     is escaped with  \u003B
:     is escaped with  \u003A
&     is escaped with  \u0026
```

An escape character which does not start with \u00 is interpreted as the start of an UTF-8 character sequence.

D-1.5 Record Data Types [DATATYPEREF]

The following codes are used within the format to define the data format of an item that can be of variant type:

Code	Name	Format	Example	Comments
General				
1	Integer	XX	23453	
2	Floating Point Number	XX.XX	12.345	
3	Engineering Format Floating Point Number	XX.XXE±NN	1.23456E+03	
4	Text	ABC	Hello World	
5	Boolean	X	1	1 if True, 0 if False
Time				
10	Relative Time	D:HH:MM:SS.SS	0:23:34:12.22	
11	Date and Time	YYYY:MM:DD:HH:MM:SS.	2010:04:20:23:34:12.	
12	Julian Day and Time	YYYY:JDD:HH:MM:SS.SS	2010:134:23:34:12.2	
Note: Time can be recorded to any number of decimal places, as defined by the data recorded				

Degree Representation

20	Degree Hemisphere	DDD.DDD H	34.442340 N	EPSG# 9116
21	Degree Minute	DDD MM.MMM	34 26.540400	EPSG# 9115
22	Degree Minute Hemisphere	DDD MM.MMM H	34 26.540400 N	EPSG#9118
23	Degree Minute Second	DDD MM SS.SSS	34 26 32.4240	EPSG#9107
24	Degree Minute Second Hemisphere	DDD MM SS.SSS H	34 26 32.4240 N	EPSG#9108
25	Hemisphere Degree	H DDD.DDDD	N 34.442340	EPSG#9117
26	Hemisphere Degree Minute	H DDD MM.MMMM	N 34 26.540400	EPSG#9119
27	Hemisphere Degree Minute Second	H DDD MM SS.SSSS	N 34 26 32.4240	EPSG#9120
28	Sexagesimal DM	DDD.MMMMMM	34.26540400	EPSG#9111
29	Sexagesimal DMS	DDD.MMSSSSSS	34.26324240	EPSG#9110
30	Sexagesimal DMS.S	DDDMMSS.SSSSS	342632.4240	EPSG#9121

Table 4 (D-1): DATATYPEREF Data Types

When recording a floating point number, the number shall be written as defined in an external source or normally to the relevant precision as defined by the precision inherent in the value recorded. It is acceptable to remove trailing decimal zeros in the bulk data.

The degree representation codes are only used when listing geodetic parameters, which should be quoted in the same format as originally provided from the source geodetic dataset. **EPSG unit code 9122 "degree (supplier to define representation)" should be regarded as decimal degrees within the 'P' formats.** All coordinates in degrees should be written as decimal degrees (EPSG unit code 9102, for example 34.4483444).

Unless a DATATYPEREF code is specifically listed for a variant data type, the DATATYPEREF code is referenced through the corresponding UNITREF code (see section D-1.8).

D-1.6 Common Header: File Identification Record

OGP: File Identification Record

Field	Description	Data Type	Reference Code	Comments
1	"OGP"	Text		
2	Contents Description	Text		e.g. "OGP P1"

3	Format Code	Integer List	FORMATREF	See Table 6 (D-1) below
4	Format Version Number	Float		Format version (this document) 1.1
5	File Issue Number	Integer		
6	Date File Written	Date		YYYY:MM:DD
7	Time File Written	Time		HH:MM:SS
8	Name of File	Text		
9	Prepared By	Text		

Note: the date and time of the file write is intended as a general reference. It should ideally be set to UTC, but can be different if this is not possible, in which case a comment record detailing the time reference used should follow this record.

Format Type Codes (FORMATREF)

Format Code	Format type
0	Common Header Only
1	P1/11
2	P2/11
6	P6/11

Table 6 (D-1): FORMATREF Format Type Codes

Example File Identification Record:

```
OGP,OGP P1,1,1.1,1,2010:02:12,21:43:01,SPEC201001.P111,OilFinder Ltd
```

Example File Identification Record (for Common Header only):

```
OGP,OGP P1 CRS,0,1.1,1,2010:02:12,21:43:01,SPEC201001.P111,OilFinder Ltd
```

For more information about the P1/11 record structure and the data types used, please refer to section 2 of the P1/11 standard document.

D-1.7 Common Header: Reference System Definitions

Three basic reference systems are defined in this part of the Common Header:

1) Unit reference systems (section D-1.8)
2) Time reference systems (not applicable for CRS definition in SEG-Y)
3) Coordinate reference systems including transformations between CRSs (section D-1.9)

The number of reference systems and transformations used in the file is provided in the following header record:

HC,1,0,0: Reference Systems Summary Information

Field	Description	Data Type	Comments
5	"Reference Systems Summary"	Description	
6	Number of Units of Measure defined	Integer	
7	Number of Time Reference Systems defined	Integer	(n/a for SEG-Y C
8	Number of Coordinate Reference Systems defined	Integer	
9	Number of Coordinate Transformations defined	Integer	

Example

```
HC,1,0,0,Reference Systems Summary                          ,5,1,4,2
```

D-1.8 Unit Reference Systems Definition

This section of the Common Header allows for the definition of all units of measure used within the P1/11 CRS definition, along with the data type used for this unit. For each unit of measure the conversion factors to convert that unit to the base unit for that measurement type shall be given. Additionally, the information source from which the unit information has been derived should be specified.

Each unit of measure is defined with a unique UNITREF code, which is then used in the remainder of the header to reference data recorded with that unit. The following UNITREF codes are reserved, user defined UNITREF codes should start from 5 onwards.

UNITREF	Units	Quantity Type	Format Code	Comments
1	Metres	Length	Floating Point	Base unit for length
2	Radians	Angle	Floating Point	Base unit for angles other than degree representations (including degree itself)
3	Degrees	Angle	Floating Point	Base unit for degree representations
4	Unity	Scale	Floating Point	Base unit for scale

Table 7 (D-1): Reserved UNITREF Codes

It is important to note that the unit of measure definition also defines the format code (see the DATATYPEREF Table 4 (D-1) earlier in this document) used to record the data, as well as the units of measure of that data. Thus you may have a “Degrees” unit of measure repeated twice with different UNITREF code, one formatted as decimal degrees, and the other formatted using a “Degree Minute Second Hemisphere” representation. In this case, both degrees units of measure will be defined relative to the base SI unit of Radians. The angular base unit is radians.

HC,1,1,0: Units of Measure Definition

Field	Description	Data Type	Reference Code	Comments
5	“Unit of Measure”	Description		
6	Unit Number	Integer	UNITREF	1 onwards (see above)
7	Unit Name	Text		
8	Quantity Type Name	Text		e.g. “length”
9	Format Reference	Integer	DATATYPEREF	See Table 4 (D-1)
10	Base Unit Number	Integer	UNITREF	Blank if this unit is the base unit
11	Conversion Factor A	Float		Blank if this unit is the base unit
12	Conversion Factor B	Float		Blank if this unit is the base unit
13	Conversion Factor C	Float		Blank if this unit is the base unit
14	Conversion Factor D	Float		Blank if this unit is the base unit
15	Description	Text		
16	EPSG Unit Code	Integer		Blank if not available
17	Source Description	Text		Defines the data source which provided details of this unit
18	Source Version Detail	Text		Defines the version of the data source which provided details of this unit
19	Source Unit Code	Variant		Defines the unit code used by the data source which provided details of this unit This item is written in the units used to define unit codes by the data source.

Note: To convert a unit X to the base unit Y

$Y = (A + BX) / (C + DX)$

Example Units of Measure Definition

```
HC,1,1,0,Unit of Measure… ,1, metre,length,2, , , , , ,SI base unit of
        length,9001,EPSG Dataset , 7.6, 9001
HC,1,1,0,Unit of Measure… ,2,radian, angle,2, , , , , ,SI angular measure
        unit,9101,EPSG Dataset , 7.6, 9101
HC,1,1,0,Unit of Measure… ,3,degree, angle,2,2, 0,3.141592654, 180,0,Measure of plane
        angle,9102,EPSG Dataset , 7.6, 9102
HC,1,1,0,Unit of Measure… ,4, unity, scale,2, , , , , ,For unitless entities,9201,EPSG
        Dataset , 7.6, 9201
```

For more information about data types used and the definition of units in the P1/11 record structure, please refer to sections 2 & 5.1 of the P1/11 standard document.

D-1.9 Coordinate Reference Systems Definition

The OGP ‘P’ formats Common Header allows any Coordinate Reference System (CRS) or coordinate transformation in use in the oil and gas industry to be defined. The format makes reference to the EPSG Geodetic Parameter Dataset (“EPSG Dataset”) during the definition of the CRS and coordinate transformation parameters. However, this should not preclude the full definition of all the coordinate reference system parameters in the SEG-Y Location data stanza; simply referencing the EPSG codes is not sufficient. To ensure that coordinates given in the SEG-Y file are unambiguous in their description of position, this format requires specification of geodetic parameters giving the full and complete definition of the coordinate reference systems in use during the survey, including transformations between different coordinate reference systems.

In general, a CRS or a coordinate transformation may be described in two ways:

- Implicit identification through citation of an EPSG code. The defining attributes and their values may then be obtained from the EPSG Dataset; or
- Explicit statement of all necessary defining attributes and their values.

In the P1/11 CRS format implicit identification alone is not acceptable. It is required by this format that the SEG-Y Location Data stanza contains the full defining parameters for all CRSs and any transformations used (“explicit definition”), and also includes implicit identification whenever the CRS or coordinate transformation data is in the EPSG Dataset.

To ensure that the format handles cases where the EPSG Dataset cannot be referenced in the definition of the geodetic parameters, the format defines internal codes for CRS Number (CRSREF) and Coordinate Transformation Number (COTRANSREF). If the EPSG Dataset is referenced then these internal codes are cross referenced to the EPSG code in the header. The internal codes are always the values used within the data headers.

In addition to the CRSs to which the coordinates in the file are referenced, the full set of survey geodetic information of earlier CRSs should be described in the Common Header to ensure that any transformation back to the earlier CRS or a common coordinate reference system (such as WGS 84) uses the correct parameters.

Latitude and longitude in the SEG-Y trace headers should be given in decimal degrees, but when parameters in transformation and conversion definitions they should be written in the same unit and to the same resolution as supplied by the information source. Thus EPSG unit code 9122 “degree (supplier to define representation)” should be regarded as decimal degrees within the ‘P’ formats.

The format follows the structure of the EPSG Geodetic Parameter Dataset and requires the use of the following parameter codes from that dataset:

- Coordinate Operation Method Codes for Map Projections and Transformations
- Coordinate Operation Parameter Codes for Map Projections and Transformations
- Coordinate Axis Codes

Any additional codes are provided for cross reference and need only be included if the geodetic parameters are directly extracted from an EPSG Dataset.

When writing explicit defining attributes and their values, if the application is referencing values from an EPSG-compliant database, the parameter names, values, signs and units must be exactly as given in that database.

In the EPSG Dataset, most coordinate transformations utilise the 2 dimensional variant of a coordinate reference system, whereas a GNSS (Global Navigation Satellite System) will provide positions in the 3 dimensional variant of the coordinate reference system. Thus, to ensure the EPSG structure is followed, it will be necessary to include both these coordinate reference systems and specify the correct 3D to 2D conversion. The table below defines the coordinate fields for each CRS type:

CRS Type	Coordinate Field 1	Coordinate Field 2	Coordinate Field 3
Projected[1]	Easting or northing[2]	Northing or easting[2]	(not used, leave blank)
Geographic 2D	Latitude	Longitude	(not used, leave blank)
Geographic 3D	Latitude	Longitude	Ellipsoidal height
Geocentric	Geocentric X	Geocentric Y	Geocentric Z
Vertical	(not used, leave blank)	(not used, leave blank)	Gravity-related height or depth[3]
Engineering 1D[4]	Distance along X axis	(not used, leave blank)	(not used, leave blank)
Engineering 2D[4,5]	Distance along X axis	Distance along Y axis	(not used, leave blank)
Engineering 3D[4]	Distance along X axis	Distance along Y axis	Distance along Z axis
Compound[6]	According to horizontal CRS	According to horizontal CRS	According to vertical CRS

Table 9 (D-1): Coordinate Reference System Types and associated Coordinate Field content

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 labelled X and Y; in such instances the first coordinate will be labelled X regardless of whether easting or northing and the second coordinate labelled Y.*

3. Whether vertical coordinates are heights (positive up) or depths (positive down) is given in the CRS definition.

4. 1D, 2D, and 3D engineering types are not explicitly split out in CRSTYPEREF (Table 10 (D-1)) but implicitly differentiated through the Coordinate System (CS) dimension instead (field 11 in HC,1,6,0).

5. Seismic bin grids are described through both an engineering 2D CRS and an associated affine transformation.

6. Compound CRS is a construct which allows coordinates from complementary horizontal 2D and vertical 1D CRSs to be linked together to form a single pseudo-3-dimensional tuple. For clarity, the horizontal CRS and vertical CRS are listed with all the relevant details, the compound CRS simply links them together into a single entity. The horizontal and vertical CRS details are not repeated in the compound CRS.

D-1.9.1 Coordinate Reference System Implicit Identification

HC,1,3,0: Coordinate Reference System Implicit Identification

Mandatory for all CRSs

Field	Description	Data Type	Reference Code	Comments
5	"CRS Number/EPSG Code/ Name/Source"	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG CRS Code	Integer		Blank if an EPSG-compliant database is not referenced
8	CRS Name	Text		
9	Version of EPSG-compliant database referenced	Text		Blank if an EPSG-compliant database is not referenced
10	Date of EPSG-compliant database referenced	Date		Blank if an EPSG-compliant database is not referenced
11	Source of EPSG-compliant database referenced	Text	e.g EPSG	Blank if an EPSG-compliant database is not referenced
12	Any Other Details	Text		Optional

Example Coordinate Reference System Implicit Identification

```
HC,1,3,0,CRS Number/EPSG Code/Name/Source…,1,      , WGS 84 / UTM zone 31N / EGM96,    ,           ,    ,
HC,1,3,0,CRS Number/EPSG Code/Name/Source…,2,32631,         WGS 84 / UTM zone 31N,7.6,2010:11:02,EPSG,Loaded
        from EPSG_v7_6.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source…,3, 4326,                       WGS 84,7.6,2010:11:02,EPSG,Loaded
        from EPSG_v7_6.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source…,4, 5773,            EGM96 Geoid Height,7.6,2010:11:02,EPSG,Loaded
        from EPSG_v7_6.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source…,5,     ,Block 29/10 (I=J+90°) bin grid,    ,           ,    ,
```

D-1.9.2 Coordinate Reference System Explicit Definition

HC,1,4,0: Coordinate Reference System Details (Explicit Definition)

Mandatory for all CRSs

Field	Description	Data Type	Reference Code	Comments
5	"CRS Number/EPSG Code/ Type/Name"	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG CRS Code	Integer		Blank if an EPSG-compliant database is not referenced
8	CRS Type Code	Integer	CRSTYPEREF	See Table 10 (D-1)
9	CRS Type	Text		As detailed in the CRSTYPEREF Table 10 (D-1)
10	CRS Name	Text		Use EPSG name if EPSG CRS code given

CRSTYPEREF: CRS Type Codes

Code	Name
1	projected
2	geographic 2D
3	geographic 3D
4	geocentric
5	vertical
6	engineering
7	compound

Table 10 (D-1): CRSTYPEREF Codes

Example

```
HC,1,4,0,CRS Number/EPSG Code/Type/Name…,1,32628,1,projected  ,WGS 84 / UTM zone 28N
HC,1,4,0,CRS Number/EPSG Code/Type/Name…,5,     ,6,engineering,Block 29/10 (I = J+90°)
      bin grid
```

HC,1,4,1: Compound CRS Horizontal CRS Identification

Mandatory when CRS type is compound. Shall not be given for any other CRS type. The horizontal CRS type shall be either Geographic 2D or Projected or Engineering. The horizontal CRS details shall be defined as a separate CRS entry.

Field	Description	Data Type	Reference Code	Comments
5	"Compound Horizontal CRS"	Description		
6	Compound CRS Number	Integer	CRSREF	
7	Horizontal CRS Number	Integer	CRSREF	
8	EPSG Horizontal CRS Code	Integer		Blank if an EPSG-compliant database is not referenced
9	Horizontal CRS Name	Text		

The Horizontal CRS is a Geographic 2D CRS, Engineering 2D CRS or a Projected CRS. Its full details shall be described within the file.

Example

```
HC,1,4,1,Compound Horizontal CRS…,4,1,32628,WGS 84 / UTM zone 28N
```

HC,1,4,2: Compound CRS Vertical CRS Identification

Mandatory when CRS type is compound. Shall not be given for any other CRS type. The vertical CRS type shall be Vertical. The vertical CRS details shall be defined as a separate CRS entry.

Field	Description	Data Type	Reference Code	Comments
5	"Compound Vertical CRS"	Description		
6	Compound CRS Number	Integer	CRSREF	
7	Vertical CRS Number	Integer	CRSREF	
8	EPSG Vertical CRS Code	Integer		Blank if an EPSG-compliant database is not referenced
9	Vertical CRS Name	Text		

The vertical CRS full details shall be described within the file.

Example

```
HC,1,4,2,Compound Vertical CRS…,4,3,5715,MSL depth
```

HC,1,4,3: Base Geographic CRS Details

Mandatory when CRS type is projected. Shall not be given for any other CRS type.

Field	Description	Data Type	Reference Code	Comments
5	“Base Geographic CRS”	Description		
6	CRS Number	Integer	CRSREF	
7	Base Geographic CRS Number	Integer	CRSREF	
8	EPSG Base Geographic CRS Code	Integer		Blank if an EPSG-compliant database is not referenced
9	Base Geographic CRS Name	Text		

The base CRS full details shall be described within the file.

Example

```
HC,1,4,3,Base Geographic CRS…,1,2,4326,WGS 84
```

HC,1,4,4: Geodetic Datum Details

Mandatory when CRS type is geocentric, geographic 3D, geographic 2D or projected. Shall not be given when CRS type is vertical, engineering or compound.

Field	Description	Data Type	Reference Code	Comments
5	“Geodetic Datum”	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG Datum Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Datum name	Text		Use EPSG name if EPSG datum code given
9	Realization Epoch	Date		If known YYYY:MM:DD

Example

```
HC,1,4,4,Geodetic Datum…,1,6326,World Geodetic System 1984,1984:01:01
```

HC,1,4,5: Prime Meridian Details

Mandatory when both the CRS type is geocentric, geographic 3D, geographic 2D or projected, and the prime meridian name is not 'Greenwich' or the Greenwich longitude is not zero. Shall not be given when CRS type is vertical, engineering or compound.

Field	Description	Data Type	Reference Code	Comments
5	"Prime Meridian"	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG Prime Meridian Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Prime Meridian name	Text		
9	Greenwich Longitude	Variant		As defined by Unit Code
10	Unit Code	Integer	UNITREF	
11	Units of Measure Name	Text		

Example

```
HC,1,4,5,Prime Meridian…,1,8909,Ferro,-17.40,8,sexagesimal DMS
```

HC,1,4,6: Ellipsoid Details

Mandatory when CRS type is geocentric, geographic 3D, geographic 2D or projected. Shall not be given when CRS type is vertical, engineering or compound.

Field	Description	Data Type	Reference Code	Comments
5	"Ellipsoid"	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG Ellipsoid Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Ellipsoid Name	Text		Use EPSG name if EPSG ellipsoid code given
9	Semi-major axis (a)	Float		
10	Unit Code	Integer	UNITREF	
11	Units of Measure Name	Text		
12	Inverse flattening (1/f)	Float		

Example

```
HC,1,4,6,Ellipsoid…,1,7030,WGS 84,6378137,1,metre,298.257223563
```

HC,1,4,7: Vertical Datum Details

Mandatory when CRS type is vertical. Shall not be given for any other CRS type.

Field	Description	Data Type	Reference Code	Comments
5	"Vertical Datum"	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG Datum Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Datum Name	Text		Use EPSG name if EPSG datum code given

Example

```
HC,1,4,7,Vertical Datum…,3,5100,Mean Sea Level
```

HC,1,4,8: Engineering Datum Details

Mandatory when CRS type is engineering. Shall not be given for any other CRS type.

Field	Description	Data Type	Reference Code	Comments
5	"Engineering Datum"	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG Datum Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Datum Name	Text		Use EPSG name if EPSG datum code given

Example

```
HC,1,4,8,Engineering Datum…,3,9315,Seismic bin grid datum
```

HC,1,5,0: Map Projection Details

Mandatory when CRS type is projected. Shall not be given for any other CRS type.

Field	Description	Data Type	Reference Code	Comments
5	"Map Projection"	Description		
6	CRS Number	Integer	CRSREF	

7	EPSG Coordinate Operation Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Projection Name	Text		Use EPSG name if EPSG code given

Example

```
HC,1,5,0,Map Projection                                      ,1,16028,UTM zone 28N
```

HC,1,5,1: Projection Method Details

Mandatory when CRS type is projected. Shall not be given for any other CRS type.

Field	Description	Data Type	Reference Code	Comments
5	"Projection Method"	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG Coordinate Operation Method Code	Integer		Use EPSG Dataset method code
8	Coordinate Operation Method Name	Text		Use EPSG name
9	Number of Projection Parameters	Integer		As defined in EPSG method. The number of **HC,1,5,2** records listed for this map projection should equal this value

Example

```
HC,1,5,1,Projection Method…    ,1,9807,Transverse Mercator,5
```

HC,1,5,2: Projection Parameter Details

Mandatory when CRS type is projected. Shall not be given for any other CRS type. For each map projection definition the number of HC,1,5,2 records shall equal the number of projection parameters for that map projection's projection method.

Field	Description	Data Type	Reference Code	Comments
5	Parameter Name	Description		Use EPSG name
6	CRS Number	Integer	CRSREF	
7	EPSG Coordinate Operation Parameter Code	Integer		Use EPSG Dataset Parameter Code

8	Parameter Value	Variant		As defined by Unit Code
9	Unit Code	Integer	UNITREF	
10	Units of Measure Name	Text		

Example

```
HC,1,5,2,Latitude of natural origin                          ,1,8801      0,3,degree
HC,1,5,2,Longitude of natural origin                         ,1,8802,   -15,3,degree
HC,1,5,2,Scale factor at natural origin                      ,1,8805,0.9996,4, unity
HC,1,5,2,False easting                                       ,1,8806,500000,1, metre
HC,1,5,2,False northing                                      ,1,8807,      0,1, metre
```

HC,1,6,0: Coordinate System Details

Mandatory when CRS type is geocentric, geographic 3D, geographic 2D, projected, vertical or engineering. Shall not be given when CRS type is compound.

Field	Description	Data Type	Reference Code	Comments
5	“Coordinate System”	Description		
6	CRS Number	Integer	CRSREF	
7	EPSG Coordinate System Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Coordinate System Name	Text		
9	Coordinate System Type Reference	Integer	CSTYPEREF	See Table 11 (D-1)
10	Coordinate System Type Name	Text		As detailed in Table 11 (D-1)
11	Dimension	Integer		The number of **HC,1,6,1** records listed for this coordinate system should equal this value

It may be necessary to incorporate reserved characters to replicate the EPSG name, for example Ellipsoidal 2D CS. Axes: latitude, longitude. Orientations: north, east. UoM: degree would be represented (using escape characters for the reserved characters) as *Ellipsoidal 2D CS. Axes\u003A latitude\u002C longitude. Orientations\u003A north\u002C east. UoM\u003A degree*

CSTYPEREF: Coordinate System Type Reference

Code	Name	Used with CRS type(s)
1	Affine	engineering
2	Cartesian	geocentric, projected, engineering
3	Ellipsoidal	geographic 3D, geographic 2D
4	Polar	engineering
5	Vertical	vertical

Table 11 (D-1): CSTYPEREF Codes and constraints in relation to CRS type

Example

```
HC,1,6,0,Coordinate System…,1,4400,Cartesian 2D CS
         ,2,Cartesian,2
HC,1,6,0,Coordinate System…,5,1033,Bin grid CS (I = J+90°). Axes\u003A I\u002C
         J,2,Cartesian,2
```

HC,1,6,1: Coordinate Axis Details

Mandatory when CRS type is geocentric, geographic 3D, geographic 2D, projected, vertical or engineering. Shall not be given when CRS type is compound. For each CRS definition the number of HC,1,6,1 records shall equal the Dimension for that CRS's Coordinate System as given in the HC,1,6,0 record field 11.

Field	Description	Data Type	Reference Code	Comments
5	"Coordinate System Axis n"	Description		Where 'n' is the Coordinate Order
6	CRS Number	Integer	CRSREF	
7	Coordinate Order	Integer		
8	EPSG Coordinate Axis Code	Integer		Use EPSG Dataset Axis code[1]
9	Axis Name	Text		Use EPSG Axis Name
10	Axis Orientation	Text		
11	Axis Abbreviation	Text		Use EPSG abbreviation if EPSG axis code given
12	Unit Code	Integer	UNITREF	
13	Units of Measure Name	Text		

1. Not to be confused with the EPSG Axis Name Code

The Coordinate Order is a sequential number from 1 onwards where the maximum value n equals the coordinate system dimension. Thus for a 3D CRS there should be 3 records of type **HC,1,6,1** with Coordinate Order values of 1,2 and 3 respectively. Within data records, coordinates are ordered within tuples as described in Table 9 (D-1). For a 1D CRS there should be one record of type **HC,1,6,1**, always with Coordinate Order value of *1*; when that 1D CRS is of CRS type vertical the vertical coordinate will be in the *third* field of the coordinate tuple.

Example

```
HC,1,6,1,Coordinate System Axis 1…,1,1,   1,Easting   ,east           ,E, 1,metre
HC,1,6,1,Coordinate System Axis 2…,1,2,   2,Northing  ,north          ,N, 1,metre
HC,1,6,1,Coordinate System Axis 1…,5,1,1428,Bin grid I,J-axis plus 90°,I,20,bin
HC,1,6,1,Coordinate System Axis 2…,5,2,1429,Bin grid J,            20°,J,20,bin
```

D-1.9.3 Coordinate Transformation Implicit Identification

If during acquisition a coordinate transformation has been applied to derive the coordinates recorded in the file (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 be given through a Coordinate Transformation implicit identification and explicit definition. This is done using records HC,1,7,0 and HC,1,8,x below.

HC,1,7,0: Coordinate Transformation Implicit Identification

Mandatory for all coordinate transformations

Field	Description	Data Type	Reference Code	Comments
5	"Transformation Number/ EPSG Code/Name/Source"	Description		
6	Coordinate Transformation Number	Integer	COTRANSREF	
7	EPSG Coordinate Operation Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Transformation Name	Text		Use EPSG name if EPSG code given
9	Version of EPSG-compliant database referenced	Text		Blank if an EPSG-compliant database is not referenced
10	Date of EPSG-compliant database referenced	Date		Blank if an EPSG-compliant database is not referenced
11	Source of EPSG-compliant database referenced	Text	e.g EPSG	Blank if an EPSG-compliant database is not referenced
12	Any Other Details	Text		Optional

Example Coordinate Transformation Implicit Identification

```
HC,1,7,0,Transformation Number/EPSG Code/Name/Source       ,1, 1613,ED50 to WGS 84
        (24),7.4.1,2010:02:01,EPSG,Loaded from EPSG_v7_4_1.mdb
HC,1,7,0,Transformation Number/EPSG Code/Name/Source       ,2,15593,geog3D to
        geog2D,7.4.1,2010:02:01,EPSG,Loaded from EPSG_v7_4_1.mdb
HC,1,7,0,Transformation Number/EPSG Code/Name/Source       ,3,     ,Block 29/10 (I = J
        + 90°) bingrid transformation, , , ,
```

D-1.9.4 Coordinate Transformation Explicit Definition

HC,1,8,0: Coordinate Transformation Name

Mandatory for all Coordinate Transformations

Field	Description	Data Type	Reference Code	Comments
5	"Transformation Number/ EPSG Code/Name"	Description		
6	Coordinate Transformation Number	Integer	COTRANSREF	
7	EPSG Coordinate Operation Code	Integer		Blank if an EPSG-compliant database is not referenced
8	Transformation Name	Text		Use EPSG name if EPSG code given
9	Transformation Accuracy	Variant		Optional. In metres. Should be given when known

Example

```
HC,1,8,0,Transformation Number/EPSG Code/Name…,1,1998,ED50 to WGS 84 (36),1
HC,1,8,0,Transformation Number/EPSG Code/Name…,3,    ,Block 29/10 (I = J+90°) bin grid
        transformation,0
```

HC,1,8,1: Coordinate Transformation Details

Mandatory for all Coordinate Transformations

Field	Description	Data Type	Reference Code	Comments
5	"Source CRS/Target CRS/Version"	Description		
6	Coordinate Transformation Number	Integer	COTRANSREF	

7	Source CRS Number	Integer	CRSREF	
8	Source CRS EPSG Code	Integer		Blank if an EPSG-compliant database is not referenced
9	Source CRS Name	Text		
10	Target CRS Number	Integer	CRSREF	
11	Target CRS EPSG Code	Integer		Blank if an EPSG-compliant database is not referenced
12	Target CRS Name	Text		
13	Transformation Version	Text		Use EPSG version if EPSG code give

This record defines the forward direction of the transformation by identifying the coordinate reference systems containing the Source and Target datums of the transformation. If referencing an EPSG-compliant database the forward direction is normally implicit in the transformation name. In the case of reversible transformations the forward direction defined here determines the signs of the transformation parameters listed in HC,1,8,4 records, regardless of the direction in which the transformation is actually applied. In the case of transformations between seismic bin grid and map grid, EPSG has defined the forward formula from map grid to bin grid.

Examples

```
HC,1,8,1,Source CRS/Target CRS/Version…,1,2, 4230,ED50                 ,3,4326,WGS
        84,EPSG-Ger Nsea
HC,1,8,1,Source CRS/Target CRS/Version…,3,2,32631,WGS 84 / UTM zone 31N,5,    ,Block
        29/10 (I = J + 90°) bin grid,
```

HC,1,8,2: Coordinate Transformation Method Details

Mandatory for all Coordinate Transformations

Field	Description	Data Type	Reference Code	Comments
5	“Transformation Method”	Description		
6	Coordinate Transformation Number	Integer	COTRANSREF	
7	Coordinate Operation Method Code	Integer		Use EPSG Dataset method code
8	Coordinate Operation Method Name	Text		Use EPSG name

9	Operation Reversible Flag	Integer		0= operation is not reversible 1= operation is reversible
10	Number of Parameters	Integer		As defined in EPSG method. The number of **HC,1,8,3** or **HC,1,8,4** records listed for this transformation should equal this value

Example

```
HC,1,8,2,Transformation Method…,1,9606,Position Vector transformation (geog2D domain),1, 7
HC,1,8,2,Transformation Method…,3,9666,P6 (I = J+90°) seismic bin grid transformation,1,10
```

HC,1,8,3: Transformation Parameter File Details

Mandatory if transformation method requires a parameter file

Field	Description	Data Type	Reference Code	Comments
5	Parameter File Name	Description		
6	Coordinate Transformation Number	Integer	COTRANSREF	
7	Coordinate Operation Parameter Code	Integer		Use EPSG Dataset Parameter Code
8	Parameter File Name	Text		
9	Operation Parameter Sign Reversal	Integer		Mandatory if operation method is reversible (**HC,1,8,2** record field 9 = 1), not required if operation method is not reversible. 0 = operation parameter sign is not reversed for reverse transformation 1 = operation parameter sign is reversed for reverse transformation

Example

```
HC,1,8,3,Latitude difference file                        ,1,8657,conus.las,1
HC,1,8,3,Longitude difference file                       ,1,8658,conus.los,1
```

HC,1,8,4: Transformation Parameter Details

Mandatory if transformation method requires a set of parameters

Field	Description	Data Type	Reference Code	Comments
5	Parameter Name	Description		Use EPSG name
6	Coordinate Transformation Number	Integer	COTRANSREF	
7	Coordinate Operation Parameter Code	Integer		Use EPSG Dataset Parameter Code
8	Parameter Value	Variant		As defined by Unit Code
9	Unit Code	Integer	UNITREF	
10	Units of Measure Name	Text		
11	Operation Parameter Sign Reversal	Integer		Mandatory if operation method is reversible (**HC,1,8,2** record field 9 = 1), not required if operation method is not reversible. 0 = operation parameter sign is not reversed for reverse transformation 1 = operation parameter sign is reversed for reverse transformation

Example

```
HC,1,8,4,X-axis translation…,1,8605,-157.89, 1,           metre,1
HC,1,8,4,Y-axis translation…,1,8606, -17.16, 1,           metre,1
HC,1,8,4,Z-axis translation…,1,8607, -78.41, 1,           metre,1
HC,1,8,4,X-axis rotation…   ,1,8608,  2.118, 9,      arc-second,1
HC,1,8,4,Y-axis rotation…   ,1,8609,  2.697, 9,      arc-second,1
HC,1,8,4,Z-axis rotation…   ,1,8610, -1.434, 9,      arc-second,1
HC,1,8,4,Scale difference…  ,1,8611,  -5.38,10,parts per million,1
```

The following example represents the parameters for a P6 (I = J + 90°) bin grid transformation (Coordi nate Operation Method 9666), source and target CRS as defined in record HC,1,8,1:

```
HC,1,8,4,Bin grid origin I…                 ,3,8733,      1.0,20,bin   ,0
HC,1,8,4,Bin grid origin J…                 ,3,8734,      1.0,20,bin   ,0
HC,1,8,4,Bin grid origin Easting…           ,3,8735, 456781.0, 1,metre ,0
HC,1,8,4,Bin grid origin Northing…          ,3,8736,5836723.0, 1,metre ,0
HC,1,8,4,Scale factor of bin grid…          ,3,8737,  0.99984, 4,unity ,0
HC,1,8,4,Bin width on I-axis…               ,3,8738,     25.0, 1,metre ,0
HC,1,8,4,Bin width on J-axis…               ,3,8739,     12.5, 1,metre ,0
HC,1,8,4,Map grid bearing of bin grid J-axis…,3,8740,     20.0, 3,degree,0
HC,1,8,4,Bin node increment on I-axis…      ,3,8741,      1.0,20,bin   ,0
HC,1,8,4,Bin node increment on J-axis…      ,3,8742,      1.0,20,bin   ,0
```

D-1.9.5 Example Point Conversion

HC,1,9,0: Example Point Conversion

Recommended

Field	Description	Data Type	Reference Code	Comments
5	“Example Point Conversion”	Description		
6	Point Number	Integer		
7	Point Name	Text		
8	CRS Number	Integer	CRSREF	
9	Coordinate 1	Variant		Format as defined for CRS
10	Coordinate 2	Variant		Format as defined for CRS
11	Coordinate 3	Variant		Format as defined for CRS

Fields 8 through 11 can be repeated as required, or the record repeated. For each point, the coordinates should be listed in at least two CRSs.

This record allows the coordinates for one or more test points to be listed referenced to different CRSs. This is to allow the configured coordinate reference systems to be checked. The point is identified by the “Point Number” which is repeated for each CRS in which the position of the point is shown.

Example Point Conversions

```
HC,1,9,0,Example Point Conversion… ,1,STN 1,1,674092.03,9716717.23,,2,-
        2.561968694,133.565880528,
HC,1,9,0,Example Point Conversion… ,1,STN 1,1,674092.03,9716717.23,,5,1528.00,3757.00,
```

D-1.10 Comment Records

Comment records should be inserted as close as possible to the data items to which they refer. They may be inserted into the header or the data section.

CC,1,0,0: Additional Information

Field	Description	Data Type	Reference Code	Comments
5	Comment	Text		

Example

```
CC,1,0,0,DATA ACQUIRED IN WGS84 GEOGRAPHIC 3D CRS
CC,1,0,0,CONVERTED IN REAL-TIME TO LOCAL PROJECTED CRS
```

D-1.11 Example CRS Definition

The following is an example of a location data Extended Textual Header stanza for a CRS definition in P1/11 format. A SEG-Y file can only contain coordinates in one CRS[20]. This may be a compound CRS linking horizontal 2D and vertical 1D CRS components (and their sub-components) into a single entity, which the CRS definition requires to be individually defined, in accordance with the IOGP coordinate model. See Table 9 (D-1).

CRSREF 1 is the compound CRS to which XYZ coordinates in the SEG-Y file refer. Its components are a horizontal (projected) CRS (WGS 84 / UTM zone 30N) and a vertical CRS (MSL height). These have been assigned to CRSREF 3 & 4 resp. The horizontal CRS is in turn the projection of a geodetic CRS (WGS 84) which has been assigned to CRSREF 5.

CRSREF 2 contains the Seismic Datum to which heights above Seismic Datum (stored in specific byte addresses in the standard and extended trace headers) refer. The relationship between Seismic Datum and the vertical datum is defined here.

Note that several lines are too long for the page and wrap around to next line. This is an artifact of fitting the data into the document and must not be done when storing the CRS definition in SEG-Y.

```
((OGP:P1/11 CRS))
OGP,OGP P1/11,0,1.1,1,2016:06:29,16:55:54,SEG-Y example.p111,IOGP Geomatics
HC,1,0,0,Reference Systems Summary                                    ,4,,5,1
HC,1,1,0,Unit of Measure                                              ,1,            metre,
       length,2, , ,                 ,       , , SI base unit of length,9001,EPSG
       Dataset,7.9,9001
HC,1,1,0,Unit of Measure                                              ,2,           radian,
       angle,2, , ,                 ,       , ,SI angular measure unit,9101,EPSG Dataset,7.9,9101
HC,1,1,0,Unit of Measure                                              ,3,           degree,
       angle,2,2,0,3.141592653589793,    180,0, Measure of plane angle,9102,EPSG Dataset,7.9,9102
HC,1,1,0,Unit of Measure                                              ,4,            unity,
       scale,2, , ,                 ,       , ,  For unitless entities,9201,EPSG Dataset,7.9,9201
CC,1,0,0,TRACE HEADER XYZ COORDINATES IN THIS SEG-Y FILE REFER TO CRSREF NUMBER 1
HC,1,3,0,CRS Number/EPSG Code/Name/Source                             ,1,     ,WGS 84 / UTM zone 30N / MSL
       height,,,,TRACE HEADER XYZ COORDINATES REFER TO THIS CRS
CC,1,0,0,SEISMIC DATUM ELEVATIONS IN THIS SEG-Y FILE REFER TO CRSREF NUMBER 2
HC,1,3,0,CRS Number/EPSG Code/Name/Source                             ,2,     ,Seismic Datum height
       ,,,,SEISMIC DATUM ELEVATIONS REFER TO THIS CRS
HC,1,3,0,CRS Number/EPSG Code/Name/Source                             ,3,32630,WGS 84 / UTM zone
       30N,8.2,2013:05:14,EPSG,Loaded from EPSG_v8_2.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source                             ,4, 5714,MSL height
       ,8.6,2014:08:10,EPSG,Loaded from EPSG_v8_6.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source                             ,5, 4326,WGS 84
       ,8.2,2013:05:14,EPSG,Loaded from EPSG_v8_2.mdb
HC,1,4,0,CRS Number/EPSG Code/Type/Name                               ,1,,7,compound,WGS 84 / UTM zone 30N /
       MSL height
HC,1,4,1,Compound Horizontal CRS                                      ,1,3,32630,WGS 84 / UTM zone 30N
HC,1,4,2,Compound Vertical CRS                                        ,1,4, 5714,MSL height
HC,1,4,0,CRS Number/EPSG Code/Type/Name                               ,2,,5,vertical,Seismic Datum height
HC,1,4,7,Vertical Datum                                               ,2,,Seismic Datum
HC,1,6,0,Coordinate System                                            ,2,6499,Vertical CS. Axis\u003A height
       (H). Orientation\u003A up. UoM\u003A m.,5,Vertical,1
HC,1,6,1,Coordinate System Axis 1                                     ,2,1,114,Gravity-related
       height,up,H,1,metre
CC,1,0,0,HORIZONTAL CRS COMPONENT OF CRSREF 1
```

[20] The vertical CRS based on Seismic Datum can also be accommodated because there are specific bytes assigned to heights referred to this datum surface.

```
HC,1,4,0,CRS Number/EPSG Code/Type/Name                   ,3,32630,1,projected,WGS 84 / UTM zone
        30N
HC,1,4,3,Base Geographic CRS                              ,3,5,4326,WGS 84
HC,1,4,4,Geodetic Datum                                   ,3,6326,World Geodetic System
        1984,1984:01:01
HC,1,4,5,Prime Meridian                                   ,3,8901,Greenwich,0,3,degree
HC,1,4,6,Ellipsoid                                        ,3,7030,WGS
        84,6378137,1,metre,298.257223563
HC,1,5,0,Map Projection                                   ,3,16030,UTM zone 30N
HC,1,5,1,Projection Method                                ,3,9807,Transverse Mercator,5
HC,1,5,2,Latitude of natural origin                       ,3,8801,     0,3,degree
HC,1,5,2,Longitude of natural origin                      ,3,8802,    -3,3,degree
HC,1,5,2,Scale factor at natural origin                   ,3,8805,0.9996,4, unity
HC,1,5,2,False easting                                    ,3,8806,500000,1, metre
HC,1,5,2,False northing                                   ,3,8807,     0,1, metre
HC,1,6,0,Coordinate System                                ,3,4400,Cartesian 2D CS. Axes\u003A
        easting\u002C northing (E\u002CN). Orientations\u003A east\u002C north. UoM\u003A
        m.,2,Cartesian,2
HC,1,6,1,Coordinate System Axis 1                         ,3,1,1, Easting, east,E,1,metre
HC,1,6,1,Coordinate System Axis 2                         ,3,2,2,Northing,north,N,1,metre
CC,1,0,0,VERTICAL CRS COMPONENT OF CRSREF 1
HC,1,4,0,CRS Number/EPSG Code/Type/Name                   ,4,5714,5,vertical,MSL height
HC,1,4,7,Vertical Datum                                   ,4,5100,Mean Sea Level
HC,1,6,0,Coordinate System                                ,4,6499,Vertical CS. Axis\u003A height
        (H). Orientation\u003A up. UoM\u003A m.,5,Vertical,1
HC,1,6,1,Coordinate System Axis 1                         ,4,1,114,Gravity-related
        height,up,H,1,metre
CC,1,0,0,BASE GEOGRAPHIC CRS OF CRSREF 3
HC,1,4,0,CRS Number/EPSG Code/Type/Name                   ,5,4326,2,geographic 2D,WGS 84
HC,1,4,4,Geodetic Datum                                   ,5,6326,World Geodetic System
        1984,1984:01:01
HC,1,4,5,Prime Meridian                                   ,5,8901,Greenwich,0,3,degree
HC,1,4,6,Ellipsoid                                        ,5,7030,WGS
        84,6378137,1,metre,298.257223563
HC,1,6,0,Coordinate System                                ,5,6422,Ellipsoidal 2D CS. Axes\u003A
        latitude\u002C longitude. Orientations\u003A north\u002C east. UoM\u003A
        degree,3,Ellipsoidal,2
HC,1,6,1,Coordinate System Axis 1                         ,5,1,106, Geodetic latitude,north,
        Lat,3,degree
HC,1,6,1,Coordinate System Axis 2                         ,5,2,107,Geodetic longitude,
        east,Long,3,degree
CC,1,0,0,THE RELATIONSHIP BETWEEN SEISMIC DATUM AND MSL IS DEFINED THROUGH A VERTICAL OFFSET
        TRANSFORMATION
HC,1,7,0,Transformation Number/EPSG Code/Name/Source      ,1,,Mean Sea Level to Seismic
        Datum,,,,RELATES SEISMIC DATUM TO MSL
HC,1,8,0,Transformation Number/EPSG Code/Name             ,1,,Mean Sea Level to Seismic Datum,
HC,1,8,1,Source CRS/Target CRS/Version                    ,1,4,5715,MSL height,2,,Seismic Datum
        height,
HC,1,8,2,Transformation Method                            ,1,9616,Vertical Offset,1,1
HC,1,8,4,Vertical Offset                                  ,1,8603,-100.0,1,metre,1
HC,1,9,0,Example Point Conversion                         ,1,STN 1,2,,,0.0,4,,,-100.0
```

D-1.12 Data Geographic Extent using OGP P1/11:

Any type of survey perimeter or geographical extent can be defined in P1/11 with a header record and P1 data records that contain the coordinates of the perimeter. The header record that identifies the perimeter type and geodetic reference for the coordinate records is the following:

P1 Header: M1 Survey Perimeter Position Definition
H1,5,0,0: M1 Survey Perimeter Definition

Field	Description	Data Type	Reference Code	Comments
5	“Survey Perimeter Definition”	Description		
6	Perimeter Number	Integer	PERIMREF	1 onwards
7	Name	Text		
8	CRS A Number	Integer	CRSREF	
9	CRS B Number	Integer	CRSREF	
10	Perimeter Type	Integer		1 = Data Extent 2 = Total Coverage 3 = Full Fold Coverage 4 = Null Full Fold Coverage 5 = Null Coverage 6 = Merged Survey Outline 7 onwards = User Defined
11	Perimeter Type Description	Text		
12	Number of Record Extension Fields Recorded Per Position Record	Integer		
13	Record Extension Field Definition	Record Extension Field text string		Optional Standard Record Extension Definition - see section 2.7 of P1/11 format description document

Field 13 is repeated as required.

Refer to Table 17 in the P1/11 format description document for format-defined record extension identifiers (as input to Field 13),

Refer to Section 2.7 in the P1/11 format description document for explanation of the record extension construction.

Example Survey Perimeter Definition

```
H1,5,0,0,Survey Perimeter Definition… ,1,Full Fold Boundary,1,5,3,Full Fold Coverage,0,
```

The list of perimeter types in field 10 can be extended with user-defined examples, such as minimum bounding rectangles in terms of lat/long, projected grid or bin grid coordinates, in which the CRSREF values in fields 8 & 9 define the coordinate reference.

The data record containing the coordinates of the perimeter has the following construction:

P1 Data Records: M1 Survey Perimeter Position Record
M1: Survey Perimeter Positions

Field	Description	Data Type	Reference Code	Comments
1	Record Identifier	Text		M1
2	Record Version	Integer		0
3	Perimeter Number	Integer	PERIMREF	
4	Point Group Number	Integer		1 onwards
5	Point Number	Integer		1 onwards
6	Segment Computation Method to next point	Integer		1 = grid 2 = geodesic (~great circle) 3 = loxodrome / rhumb line 4 = parallel of latitude arc 5 = meridional arc
7	CRS A Coordinate 1	Variant		Format for CRS A as listed in H1,5,0,0 and as defined in HC,1,6,1
8	CRS A Coordinate 2	Variant		Format for CRS A as listed in H1,5,0,0 and as defined in HC,1,6,1
9	CRS A Coordinate 3	Variant		Format for CRS A as listed in H1,5,0,0 and as defined in HC,1,6,1 Blank if n/a
10	CRS B Coordinate 1	Variant		Format for CRS B as listed in H1,5,0,0 and as defined in HC,1,6,1
11	CRS B Coordinate 2	Variant		Format for CRS B as listed in H1,5,0,0 and as defined in HC,1,6,1
12	CRS B Coordinate 3	Variant		Format for CRS B as listed in H1,5,0,0 and as defined in HC,1,6,1 Blank if n/a
13	Record Extension Fields	Additional Field List		The number of items must equal that given in the H1,5,0,0 record

Fields 5 onwards can be repeated as required.

The record can be repeated as required, with vertices in sequential order around the perimeter. The coordinates of the first vertex should be repeated at the end of the list as the (n+1)th vertex. No segment computation method should be given with the coordinates for the (n+1)th vertex.

The position tuple in CRS B is mandatory for the first set of positions in each record, it is optional in the second and subsequent sets of positions, but optional fields must retain their field delimiters.

The point group number is available to allow multiple areas to be defined as part of the same perimeter – thus the first discrete area is given a group number of 1, the second area is given a group number of 2 etc.

The Segment Computation Method defines the line computation from one position to the next. If the perimeter is given in the coordinates of 2 CRSs (eg map grid and bin grid), the segment computation method (and the resulting connection) refers to the grid defined by CRS A.

The following is an example of a location data Extended Textual Header stanza for a data geographic extent definition in OGP P1/11 format. This must be preceded by the OGP P1/11 CRS definition to identify the geodetic reference of the perimeter coordinates.

```
((OGP:P1/11 CRS))
.
.
((OGP:P1/11 Data Geographic Extent))
H1,5,0,0,Survey Perimeter Definition… ,1,Block boundary,1,5,7,Block Boundary,0,
M1,0,1,1,1,1,391194.94,4092809.86,,54.2344345434,-9.2344345434,,
M1,0,1,1,2,1,392747.34,4093232.60,,54.2655123423,-9.2435354534,,
M1,0,1,1,3,1,393576.45,4094267.73,,54.2834225677,-9.2578834354,,
M1,0,1,1,4,1,391243.56,4095786.14,,54.2535353553,-9.2367002431,,
M1,0,1,1,1, ,391194.94,4092809.86,,54.2344345434,-9.2344345434,,
```

D-2. Seismic Bin Grid Definition: International Association of Oil and Gas Producers P6/11

The OGP P6/11 Seismic bin grid data exchange format is preferred for bin grid definition and survey coverage perimeter. It must start with an OGP File Identification record, followed by the CRS definition section as described in Appendix D-1, in order to define the bin grid geometry, survey perimeter or bin node coordinates, depending on the purpose of the data.

The bin grid definition blocks contain text like this:

```
((OGP:P1/11 CRS))
P1-format CRS definition records (lines of text)
.
((OGP:P6/11 Coverage Perimeter))
P6-format perimeter position records (lines of text)
.
((OGP:P6/11 Bin Node Coordinates))
P6-format bin node position records (lines of text)
     :
((SEG:EndText))
```

There are header records specific to P6/11 necessary to give further definition to bin grid position records, as shown below.

D-2.1 P6 Header: File Content Definitions

H6,0,0,0: File Contents Description

Field	Description	Data Type	Reference Code	Comments
5	"File Contents Description"	Text, LJ50		
6	Description	Text		
7	Any Other Details	Text		Optional

H6,0,1,0: File Processing Details

Field	Description	Data Type	Reference Code	Comments
5	"Processing Details"	Text, LJ50		
6	Details	Text		

Record can be repeated as required

H6,0,2,0: File Contents Attribute

Field	Description	Data Type	Reference Code	Comments
5	Attribute Name	Description		See Table 12 (D-2)
6	Unique Attribute Number	Integer	ATTREF	See Table 12 (D-2)
7	Attribute Value	Variant		If fields 8 and 9 are blank, the Attribute Value is assumed to be of text data type
8	Attribute Units	Integer	UNITREF	If not listed, the attribute value is assumed to be of text format
9	Attribute Unit Name	Text		If not listed, the attribute value is assumed to be of text format

Record can be repeated as required

File Contents Attribute Reference Numbers [ATTREF]

ATTREF Code	Description	Comments
1	Receiver Groups Per Shot	
2	Original File	Used when the file is converted from an original output file
100 onwards	User Defined	User to provide Attribute Name in H6,0,2,0

Table 12 (D-2): File Contents Attribute Reference Numbers

Example P6 header records

```
H6,0,0,0,File Contents Description...,Processed bin grid geometry and data extent
         perimeter,Checked in-house by Survey Dept
H6,0,1,0,Processing Details...       ,Bin grid produced by seismic processing system
H6,0,2,0,Original File...            ,2,Quad_36N_Acqn_bingrid.p611,,
```

D-2.2 Bin Node Position Definition using OGP P6/11

The term bin node is used instead of bin centre and refers to the locations where the bin grid lines intersect. Coordinates of three check nodes are required as a minimum to permit numerical verification of the bin grid definition parameters. These can be recorded as B6 bin node position records as this section describes, but may also be recorded as Example Point Conversions (in HC 1,9,0 records) to distinguish them from normal bin nodes. Two of these points should be on the J axis and the third point should be remote from the J axis within the area of coverage. Any other bin node can also be stored in the same record type.

P6 Header: Position Record Type Definitions

H6,1,0,0: Bin Node Position Record Type Definitions

Field	Description	Data Type	Reference Code	Comments
5	"Bin Node Position Record Definition"	Description		
6	Record Type Number	Integer	P6TYPEREF	
7	CRS A Number	Integer	CRSREF	
8	CRS B Number	Integer	CRSREF	
9	Number of Record Extension Fields Recorded Per Record Integer	Integer		
10	Record Extension Field Definition	Record Extension Field text string		Optional Standard Record Extension Definition - see section 2.7 of P1/11 format description document

Field 10 is repeated as required.

Refer to Table 13 (D-2) for format-defined record extension identifiers (as input to Field 10).

Refer to Section 2.7 in the P6/11 format description document for an explanation of the record extension construction.

Each Position Record provides storage for the position referenced to two CRSs. It is expected that generally CRS A will be the bin grid CRS or a compound CRS encompassing the bin grid CRS, whereas CRS B will be the projected CRS or a compound CRS encompassing the projected CRS. However, this is not an absolute rule and the CRSs used can be defined in the logical way for the position data recorded.

There is also a facility to store attributes with each bin position record, such as fold, water depth etc, using the position record extension construction:

Bin Node Position Record Extension Field Identifiers

Extension Identifier	Description	Additional Parameter
1	Water Depth	Vertical CRS Reference (CRSREF)
2	Vertical CRS Difference	The From (source) and To (target) Vertical CRS References (CRSREF), separated by an ampersand. Unit is in source CRS
3	Point Depth	Vertical CRS Reference (CRSREF)
4	Fold	
100 onwards	User Defined	

Table 13 (D-2): Position Record Extension Field Data Identifiers

Position Record Definition Example

```
H6,1,0,0,Bin Node Position Record Definition… ,1,3,1,1,1;4;Water Depth;1
```

Having defined the bin node positions in the H6 header, the coordinates themselves are contained in B6 bin node position records, as follows:

P6 Data Records: B6 Bin Node Position Record

B6: Bin Node Position Record

Field	Description	Data Type	Reference Code	Comments
1	Record Identifier	Text		B6
2	Record Version	Integer		0
3	Record Type Number	Integer	P6TYPEREF	
4	CRS A Coordinate 1	Variant		Format as defined for CRS A listed in H6,1,0,0
5	CRS A Coordinate 2	Variant		Format as defined for CRS A listed in H6,1,0,0
6	CRS A Coordinate 3	Variant		Format as defined for CRS A listed in H6,1,0,0
7	CRS B Coordinate 1	Variant		Format as defined for CRS B listed in H6,1,0,0
8	CRS B Coordinate 2	Variant		Format as defined for CRS B listed in H6,1,0,0

9	CRS B Coordinate 3	Variant	Format as defined for CRS B listed in H6,1,0,0
10	Record Extension Fields	Additional Field List	The number of items must equal that given in the H6,1,0,0 record

Fields 4 onwards can be repeated as required.

Record can be repeated as required.

The position tuple in CRS B is mandatory for the first set of positions in each record. It is optional in the second and subsequent sets of positions listed in each record.

The following is an example of a location data stanza for bin node coordinates in P6/11 format. This must be preceded by the OGP P1/11 CRS definition to identify the geodetic reference of the bin node coordinates.

```
((OGP:P1/11 CRS))
.
.
((OGP:P6/11 Bin Node Coordinates))
H6,1,0,0,Bin Node Position Record Definition… ,1,3,1,1,1;4;Water Depth;1
B6,0,1,4000.00,8040.00,,372653.00,6202462.04,,100.4
B6,0,1,4000.00,6240.00,,370705.12,6157491.61,,110.7
B6,0,1,1700.00,6240.00,,313242.90,6159980.57,, 89.8
```

D-2.3 Coverage Perimeter Definition using OGP P6/11:

The survey perimeter facility in OGP P6/11 is designed to describe the limits of a 3D dataset. Any meaningful perimeter can be defined, such as total coverage, full-fold coverage etc, including islands of null full-fold or of no data within the full-fold areas. A perimeter is defined with a header record and P6 data records that contain the coordinates of the perimeter. These are similar to the equivalent records for data geographic extent in OGP P1/11. The header record that identifies the perimeter type and geodetic reference for the coordinate records is the following:

P6 Header: M6 Survey Perimeter Position Definition

H6,2,0,0: M6 Survey Perimeter Definition

Field	Description	Data Type	Reference Code	Comments
5	"Survey Perimeter Definition"	Description		
6	Perimeter Number	Integer	PERIMREF	1 onwards
7	Name	Text		
8	CRS A Number	Integer	CRSREF	
9	CRS B Number	Integer	CRSREF	

10	Perimeter Type	Integer	1 = Data Extent 2 = Total Coverage 3 = Full Fold Coverage 4 = Null Full Fold Coverage 5 = Null Coverage 6 = Merged Survey Outline 7 onwards = User Defined
11	Perimeter Type Description	Text	
12	Number of Record Extension Fields Recorded Per Position Record	Integer	
13	Record Extension Field Definition	Record Extension Field text string	Optional Standard Record Extension Definition - see section 2.7 of P1/11 format description document

Field 13 is repeated as required.

Refer to Table 13 (D-2) for format-defined record extension identifiers (as input to Field 13).

Example Survey Perimeter Definition

```
H6,2,0,0,Survey Perimeter Definition… ,1,Full Fold Boundary,2,1,3,Full Fold Coverage,0,
```

Having defined the perimeter in the H6 header, the coordinates themselves are contained in M6 survey perimeter position records, as follows:

P6 Data Records: M6 Survey Perimeter Position Record

M6: Survey Perimeter Positions

Field	Description	Data Type	Reference Code	Comments
1	Record Identifier	Text		M6
2	Record Version	Integer		0
3	Perimeter Number	Integer	PERIMREF	
4	Point Group Number	Integer		1 onwards
5	Point Number	Integer		1 onwards
6	Segment Computation Method to next point	Integer		1 = grid 2 = geodesic (~great circle) 3 = loxodrome / rhumb line 4 = parallel of latitude arc 5 = meridional arc

7	CRS A Coordinate 1	Variant	Format for CRS A as listed in H6,2,0,0 and as defined in HC,1,6,1
8	CRS A Coordinate 2	Variant	Format for CRS A as listed in H6,2,0,0 and as defined in HC,1,6,1
9	CRS A Coordinate 3	Variant	Format for CRS A as listed in H6,2,0,0 and as defined in HC,1,6,1
10	CRS B Coordinate 1	Variant	Format for CRS B as listed in H6,2,0,0 and as defined in HC,1,6,1
11	CRS B Coordinate 2	Variant	Format for CRS B as listed in H6,2,0,0 and as defined in HC,1,6,1
12	CRS B Coordinate 3	Variant	Format for CRS B as listed in H6,2,0,0 and as defined in HC,1,6,1
13	Record Extension Fields	Additional Field List	The number of items must equal that given in the H6,2,0,0 record

Fields 5 onwards can be repeated as required.

The record can be repeated as required, with nodes in sequential order around the perimeter. The coordinates of the first node should be repeated at the end of the list as the (n+1)th node. No segment computation method should be given with the coordinates for the (n+1)th node.

The position tuple in CRS B is mandatory for the first set of positions in each record, it is optional in the second and subsequent sets of positions, but optional fields must retain their field delimiters.

The point group number is available to allow multiple areas to be defined as part of the same perimeter – thus the first discrete area is given a group number of 1, the second area is given a group number of 2 etc.

The Segment Computation Method defines the line computation from one position to the next. If the perimeter is given in the coordinates of 2 CRSs (eg map grid and bin grid), the segment computation method (and the resulting connection) refers to the grid defined by CRS A.

The following is an example of a location data stanza for a coverage perimeter in P6/11 format. This must be preceded by the OGP P1/11 CRS definition to identify the geodetic reference of the perimeter coordinates.

```
((OGP:P1/11 CRS))
.
.
((OGP:P6/11 Coverage Perimeter))
H6,2,0,0,Survey Perimeter Definition… ,1,Full Fold Boundary,2,1,3,Full Fold
       Coverage,0,
M6,0,1,1,1,1,502500.0,6201000.0,,1700.0,8040.0,,
M6,0,1,1,2,1,560000.0,6201000.0,,4000.0,8040.0,,
M6,0,1,1,3,1,560000.0,6156000.0,,4000.0,6240.0,,
M6,0,1,1,4,1,502500.0,6156000.0,,1700.0,6240.0,,
M6,0,1,1,1, ,502500.0,6201000.0,,1700.0,8040.0,,
```

The following is an example of a location data stanza for a bin grid. It is headed by the P1/11 CRS definition to identify the geodetic reference for the survey, and contains a perimeter definition (total coverage) and some example bin nodes. Note this is not a complete file but cut back to essential detail.

```
((OGP:P1/11 CRS))
OGP,OGP P6,6,1.1,1,2016:01:04,22:22:58,Quad_36N_Acqn_bingrid.p611,SeisCo
HC,1,0,0,Reference Systems Summary                          ,5,,4,2
HC,1,1,0,Unit of Measure
         ,1,metre,length,2,,,,,,metre,9001,EPSG,8.3,9001
HC,1,1,0,Unit of Measure                                    ,2,radian,angle,2,,,,,,SI angular
         measure unit,9101,EPSG,8.3,9101
HC,1,1,0,Unit of Measure
         ,3,degree,angle,2,2,0,3.14159265358979,180,0,Measure of plane angle,9102,EPSG,8.3,9102
HC,1,1,0,Unit of Measure                                    ,4,unity,scale,2,,,,,,For unitless
         entities,9201,EPSG,8.3,9201
HC,1,1,0,Unit of Measure
         ,5,bin,scale,2,4,0,1,1,0,bin,1024,EPSG,8.3,1024
CC,1,0,0,COORDINATE REFERENCE SYSTEMS IMPLICIT IDENTIFICATION
HC,1,3,0,CRS Number/EPSG Code/Name/Source                   ,1,4326,WGS
         84,8.3,2013:11:29,EPSG,Loaded from EPSG_v8_3.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source
         ,2,4230,ED50,8.3,2013:11:29,EPSG,Loaded from EPSG_v8_3.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source                   ,3,23090,ED50 / TM 0
         N,8.9.2,2016:04:06,EPSG,Loaded from EPSG_v8_9_2.mdb
HC,1,3,0,CRS Number/EPSG Code/Name/Source                   ,4,,Quad 36N (I=J+90°) bin grid,,,,Bin
         grid used for acquisition
CC,1,0,0,COORDINATE REFERENCE SYSTEMS EXPLICIT DEFINITION
.
.
HC,1,4,0,CRS Number/EPSG Code/Type/Name                     ,4,,6,engineering, Quad 36N (I=J+90°)
         bin grid
HC,1,4,8,Engineering Datum                                  ,4,9315,Seismic bin grid datum
HC,1,6,0,Coordinate System                                  ,4,1033,Bin grid CS (I = J+90°).
         Axes\u003A I\u002C J.,2,Cartesian,2
HC,1,6,1,Coordinate System Axis 1                           ,4,1,1428,Bin grid I, J-axis plus
         90°,I,5,bin
HC,1,6,1,Coordinate System Axis 2                           ,4,2,1429,Bin grid J, 0° (see
         associated operation COTRANSREF 2),J,5,bin
CC,1,0,0,COORDINATE TRANSFORMATIONS IMPLICIT IDENTIFICATION
HC,1,7,0,Transformation Number/EPSG Code/Name/Source        ,1,1311,ED50 to WGS 84
         (18),8.3,2013:11:29,EPSG,Loaded from EPSG_v8_3.mdb
HC,1,7,0,Transformation Number/EPSG Code/Name/Source        ,2,,ED50 / TM 0 N to Quad 36N
         (I=J+90°) bin grid,,,,Used to project bin grid
CC,1,0,0,COORDINATE TRANSFORMATIONS EXPLICIT DEFINITION
.
.
HC,1,8,0,Transformation Number/EPSG Code/Name               ,2,,ED50 / TM 0 N to Quad 36N
         (I=J+90°) bin grid,
HC,1,8,1,Source CRS/Target CRS/Version                      ,2,3,23090,ED50 / TM 0 N,4,, Quad 36N
         (I=J+90°) bin grid,
HC,1,8,2,Transformation Method                              ,2,9666,P6 (I = J+90°) seismic bin
         grid transformation,1,10
HC,1,8,4,Bin grid origin I                                  ,2,8733,0.0,5,bin,0
HC,1,8,4,Bin grid origin J                                  ,2,8734,0.0,5,bin,0
HC,1,8,4,Bin grid origin Easting                            ,2,8735,500000.00,1,metre,0
HC,1,8,4,Bin grid origin Northing                           ,2,8736,6206216.61,1,metre,0
```

```
HC,1,8,4,Scale factor of bin grid                    ,2,8737,1.0,4,unity,0
HC,1,8,4,Bin width on I-axis                         ,2,8738,25.0,1,metre,0
HC,1,8,4,Bin width on J-axis                         ,2,8739,50.0,1,metre,0
HC,1,8,4,Map grid bearing of bin grid J-axis         ,2,8740,0.0,3,degree,0
HC,1,8,4,Bin node increment on I-axis                ,2,8741,1.0,5,bin,0
HC,1,8,4,Bin node increment on J-axis                ,2,8742,1.0,5,bin,0
CC,1,0,0,EXAMPLE POINT CONVERSION
HC,1,9,0,Example Point Conversion                    ,1,Well 36/1-
         1,3,516765.93,6302799.68,,4,1630,5001,
```

((OGP:P6/11 Coverage Perimeter))

```
CC,1,0,0,FILE CONTENT DEFINITIONS
H6,0,0,0,File Contents Description                   ,Acquisition bin grid geometry and
         survey perimeter,Checked by Survey Rep
H6,0,1,0,Processing Details                          ,Bin grid produced by CircumNav system
H6,0,2,0,Original File                               ,2,Quad_36N_Preplot_bingrid.p611,,
H6,0,2,0,I-Axis Description                          ,100,In-line,,
H6,0,2,0,J-Axis Description                          ,101,Cross-line,,
CC,1,0,0,SURVEY PERIMETER DEFINITIONS
H6,2,0,0,Survey Perimeter Definition                 ,1,Quad 36N Total Coverage,3,4,2,Total
         Coverage,0,
M6,0,1,1,1,1,500000.00,6187666.57,,   0.0,   0.0,,
M6,0,1,1,2,1,500000.00,6206216.61,,   0.0, 371.0,,
M6,0,1,1,3,1,512474.17,6206234.66,,498.97,371.36,,
M6,0,1,1,4,1,512527.80,6187684.66,,501.11,  0.36,,
M6,0,1,1,1, ,500000.00,6187666.57,,   0.0,   0.0,,
```

((OGP:P6/11 Bin Node Coordinates))

```
H6,1,0,0,Bin Node Position Record Definition… ,1,4,3,2,1;4;Water Depth;1,4;;Fold;4
B6,0,1,17000.00,4308.00,,421552.71,6838594.02,,101.5&48
B6,0,1,17000.00,4309.00,,421577.71,6838593.72,,100.6&47
B6,0,1,17000.00,4310.00,,421602.71,6838593.41,,100.1&47
```

D-2.4 Bin Grid Coordinate Reference Systems

The coordinates used for bin grids are:

- The map grid coordinates, which provide a geodetic reference frame
- The bin grid coordinates, which provide a relative reference frame.

These reference frames are related by an affine transformation.

The map grid coordinates require the definition of the projected CRS including the geodetic datum, the reference ellipsoid, the coordinate system and the map projection. See the P6/11 User Guide for further details.

The bin grid description should comprise of:

- For bin grids where the I-axis is 90 degrees clockwise from the J-axis (when viewed from above the plane of the two bin grid axes):
  - An engineering CRS (similar to code 5818 in the EPSG Dataset which is an example), plus
  - An affine transformation using EPSG operation method 9666. The EPSG Dataset contains an example in Coordinate Transformation 6918.

When adding a bin grid to an EPSG-compliant database, users should give the engineering CRS and the affine transformation their own unique codes, avoiding the reserved EPSG code range of 0-32767. Where this is not the case, the CRS and transformation can be defined without EPSG codes.

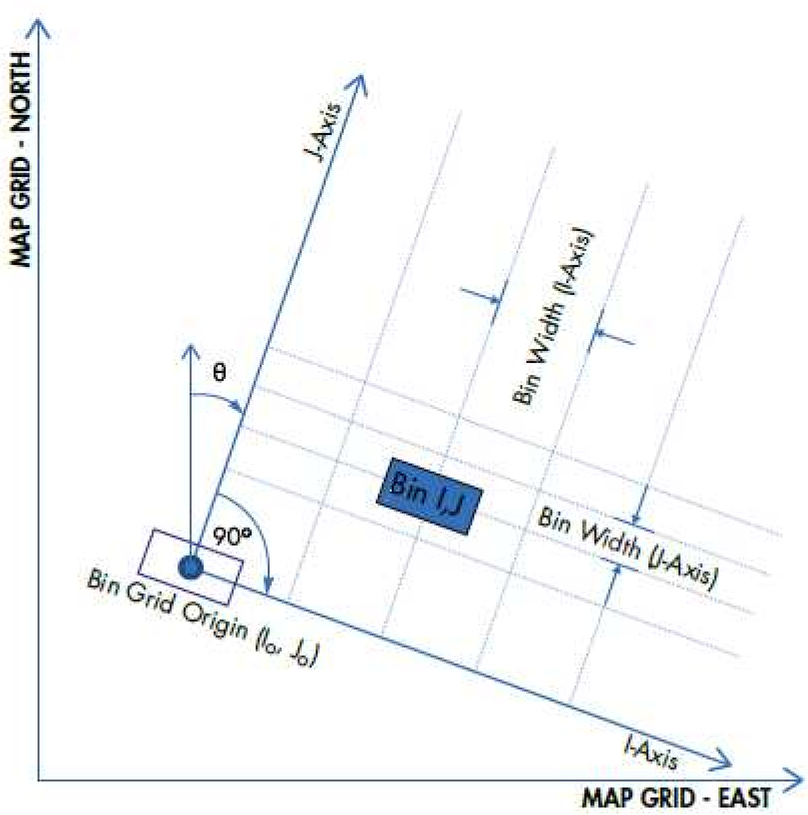

Figure 1 (D-2): I-axis is 90 degrees clockwise from the J-axis when viewed from above the plane of the two bin grid axes.

- For bin grids where the I-axis is 90 degrees counter-clockwise from the J-axis (when viewed from above the plane of the two bin grid axes):
    - An engineering CRS (similar to code 5859 in the EPSG Dataset which is an example), plus
    - An affine transformation using EPSG operation method 1049. The EPSG Dataset contains an example in Coordinate Transformation 6919.

When adding a bin grid to an EPSG-compliant database, users should give the engineering CRS and the affine transformation their own unique codes, avoiding the reserved EPSG code range of 0-32767. Where this is not the case, the CRS and transformation can be defined without EPSG codes.

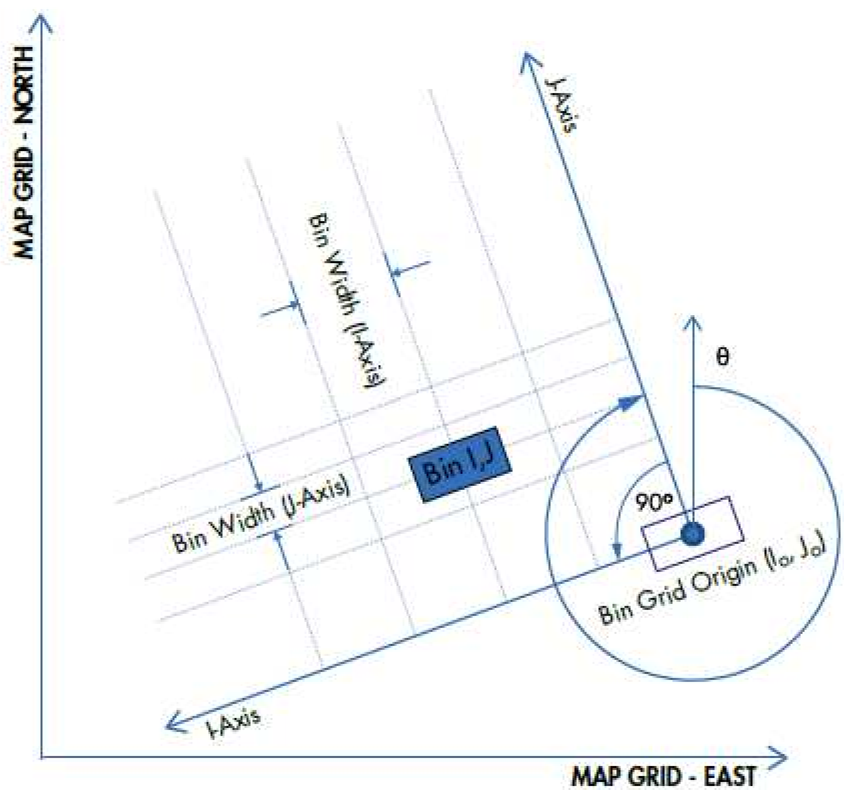

Figure 2 (D-2): I-axis is 90 degrees counter-clockwise from the J-axis when viewed from above the plane of the two bin grid axes.

Note: θ is the Bin Grid Orientation in Figure 1 (D-2) & 2 (D-2)

Note that for either type of bin grid, the text in the axis direction fields in the EPSG example CRSs, should be replaced with the actual values in header record [HC,1,6,1] - Axis Orientation, field 10 in the P6/11 file. The direction of the J-axis should be the equivalent value as that for the transformation parameter “Map grid bearing of bin grid J-axis”, (field 8 of HC,1,8,4). The direction of the I-axis should be this value ± 90.

The P6/11 User Guide, due for issue in 2016, will have example records. For details of the affine transformation methods, see EPSG Guidance Note 7-2.

D-3. Coordinate Reference System Definition (Deprecated)

The older, deprecated, stanzas are prefixed with

((SEG: Location Data ver 3.0)),
((SEG: Location Data EPSG Reference ver 3.0)),
((SEG: Location Data Coordinate Transformation ver 3.0)), or
((SEG: Location Data Coordinate Transformation EPSG Reference ver 3.0)).

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-3.3. Examples are given in D-3.4.

Some keywords in both the Location Data stanza (D-3.1 table 5) and the Location Data Coordinate Transformation stanza (D-3.3 table 7) 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.

Degree Representation

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.

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. Any value given in sexagesimal DMS must be clearly labeled as such.

Sexagesimal DMS must be used *only* for geodetic parameter values in Extended Textual Header Location Data stanzas. For latitude and longitude coordinates given in General Header records, signed decimal degrees must be used.

D-3.1 Stanza for Location Data

The Location Data stanza identifies the Coordinate Reference System (CRS) to which the coordinates for the source, group or CDP location given in Standard Trace Header bytes 71–88 and 181–188 and optional SEG-Y Trace Header Extension 1 bytes 97–128 are referenced. It also identifies the system to which the elevations and depths given in trace header bytes 41–70 and optional SEG-Y Trace Header Extension 1 bytes 33–96 are referenced. Without this identification these coordinates are ambiguous. This information may be given implicitly[21] through reference to the EPSG Geodetic Parameter Dataset (Table 7) or explicitly[22] (Table 8).

Whenever the Coordinate Reference System used is included in the EPSG dataset both the implicit identification 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.

[21] Implicit identification 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.

[22] 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.

Table 7 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[23] =	Text	80 character limit	The release number for the EPSG Geodetic Parameter Dataset.

[23] 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 8 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.			

Stanza Header and Keyword	Format	Resolution / Limits	Comment
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 <= • $_G$ <= +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”.
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	

Stanza Header and Keyword	**Format**	**Resolution / Limits**	**Comment**
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	

Stanza Header and Keyword	**Format**	**Resolution / Limits**	**Comment**
Projection parameter n value =	Real Number	IEEE double precision. See EPSG dataset for examples	
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.

Stanza Header and Keyword	**Format**	**Resolution / Limits**	**Comment**
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 ratio = 0.3048; if unit = unity, conversion ratio = 1.0.
CS axis 2 unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
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”.

Stanza Header and Keyword	Format	Resolution / Limits	Comment
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-3.2 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**

D-3.3 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[24] through reference to the EPSG Geodetic Parameter Dataset (Table 9) or described explicitly[25] (Table 10).

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).

[24] Implicit identification 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.

[25] 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.

Table 9 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[26] =	Text	80 character limit	The release number for the EPSG Geodetic Parameter Dataset.

[26] 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 10 Stanza for explicit identification 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:			

Stanza Header and Keyword	Format	Resolution / Limits	Comment
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:			
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.

Stanza Header and Keyword	Format	Resolution / Limits	Comment
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 = 0.3048; if unit = parts per million, conversion ratio = 0.000001.
Transformation parameter n unit name =	Text	80 character limit, but must be from EPSG Unit of Measure table Name field.	
Transformation parameter n unit conversion ratio =	Real Number	IEEE double precision.	

Stanza Header and Keyword	**Format**	**Resolution / Limits**	**Comment**
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.	

Stanza Header and Keyword	**Format**	**Resolution / Limits**	**Comment**
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
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-3.4 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**

D-4. Bin Grid Definition: Deprecated

D-4.1 Stanza for Bin Grid Definition

The Bin Grid Definition stanza defines a bin grid including its relationship to a projected coordinate reference system (map grid). The projected coordinate reference system must be defined in a Location Data stanza (see section D-3). The content of this Bin Grid Definition stanza follows the provisions of the UKOOA P6/98 v3.0 format. As with the location stanzas, the OGP P6/11 format is recommended when backward compatibility is not an issue. See Appendix D-2 for an explanation of the preferred format.

The bin grid is the relative coordinate framework which defines a matrix of evenly spaced points referred to as the bin nodes. The term bin node is used instead of the term bin center and refers to the locations where the bin grid lines intersect.

The bin grid is defined by a pair of orthogonal axes designated the I and the J axes. In the right-handed case the I axis is rotated 90 degrees clockwise from the J axis. The order of specifying bin grid coordinates will be the I value followed by the J value (I , J). The choice of I, J axes is made to avoid any confusion between bin grid (I,J) and map grid (E,N) coordinates. Axes may be labeled by users as they wish within their own software, including such terms as In-line and Cross-line, Row and Column, x and y, Line and Trace. There is no industry accepted common terminology for axis labeling and terms such as In-line and Cross-line are used in contradictory ways by different users. For the purpose of data exchange through SEG-Y the only reference is to the I and J axes.

Coordinates of three check nodes are required to permit numerical verification of the bin grid definition parameters. Two of these points should be on the J axis and the third point should be remote from the J axis within the area of coverage.

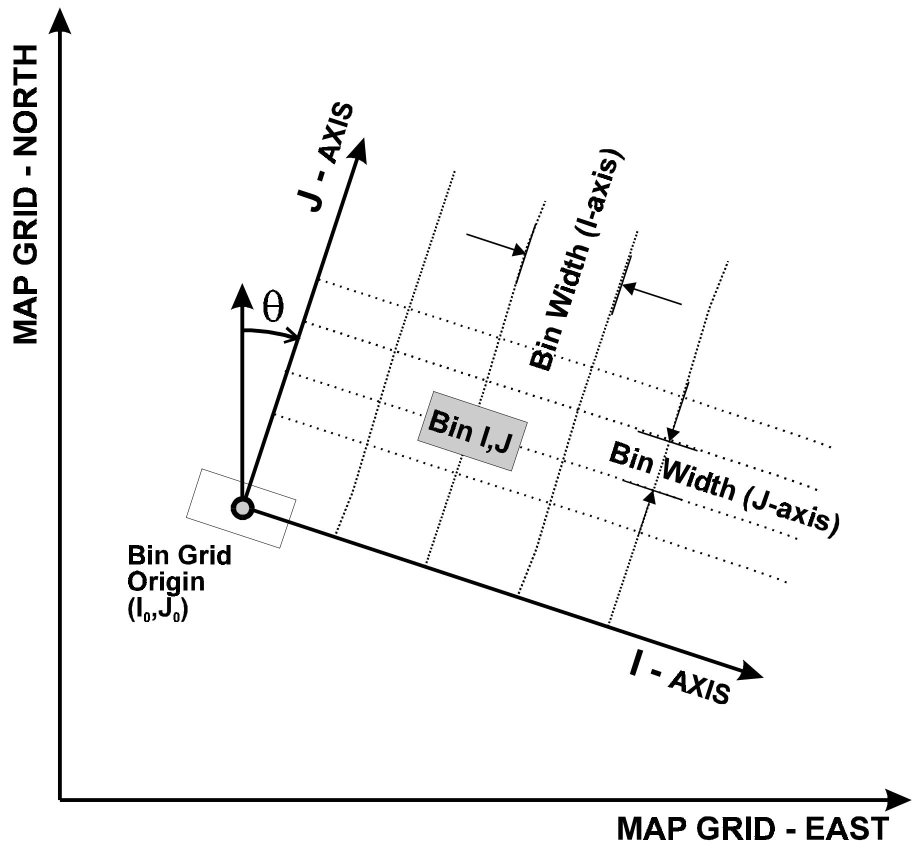

Figure 3 Bin grid definition (Right-handed case)

Table 11 Stanza for Bin Grid Definition

Stanza Header and Keyword	Format	Comment
((SEG: Bin Grid Definition ver 1.0))	Text	Stanza name
Bin grid name =	Text	Text description of the defined bin grid.
Alternate I-Axis description	Text	The description used in the acquisition documentation to describe the I-Axis orientation (i.e. cross-line, X-Axis)
Alternate J-Axis description	Text	The description used in the acquisition documentation to describe the J-Axis orientation (i.e. in-line, Y-Axis)
Bin grid origin I coordinate =	Real Number	Bin grid I coordinate at the bin grid origin.
Bin grid origin J coordinate =	Real Number	Bin grid J coordinate at the bin grid origin. The positive J axis is orientated 90 degrees counter clockwise from the positive I axis.
Bin grid origin Easting =	Real Number	Map grid Easting coordinate at the bin grid origin.
Bin grid origin Northing =	Real Number	Map grid Northing coordinate at the bin grid origin.

Stanza Header and Keyword	Format	Comment
Scale factor of bin grid =	Real Number	Map grid scale factor at any bin node within the bin grid, preferably the center of the area of coverage. This is NOT the same as the scale factor at the projection origin. *If the survey has been acquired on the map grid, then the node interval is a map grid interval and the Scale Factor of the Bin Grid is unity.*
Scale factor node I coordinate =	Real Number	Bin grid I coordinate of the bin node at which the scale factor (above) has been defined. Not required if scale factor of bin grid is unity.
Scale factor node J coordinate =	Real Number	Bin grid J coordinate of the bin node at which the scale factor (above) has been defined. Not required if scale factor of bin grid is unity.
Nominal bin width on I axis =	Real Number	Nominal separation of bin nodes in the I-axis direction. Units are those of the projected coordinate reference system (map grid).
Nominal bin width on J axis =	Real Number	Nominal separation of bin nodes in the J-axis direction. Units are those of the projected coordinate reference system (map grid).
Grid bearing of bin grid J axis =	Real Number	Bearing of the positive direction of the bin grid J-axis defined as a map grid bearing, measured clockwise from map grid north.
Grid bearing unit name =	Text	The name of the angle unit for the bin grid bearing.
Bin node increment on I axis =	Real Number	Increment value between adjacent bin grid nodes in the I-axis direction.
Bin node increment on J axis =	Real Number	Increment value between adjacent bin grid nodes in the J-axis direction.
First check node I coordinate =	Real Number	
First check node J coordinate =	Real Number	
First check node Easting =	Real Number	
First check node Northing =	Real Number	
Second check node I coordinate =	Real Number	
Second check node J coordinate =	Real Number	

Stanza Header and Keyword	Format	Comment
Second check node Easting =	Real Number	
Second check node Northing =	Real Number	
Third check node I coordinate =	Real Number	
Third check node J coordinate =	Real Number	
Third check node Easting =	Real Number	
Third check node Northing =	Real Number	

D-4.2 Example for Bin Grid Definition

((SEG: Bin Grid Definition ver 1.0))	
Bin grid name =	**Marine X final migrated volume**
Alternate I-Axis description	**Cross-line**
Alternate J-Axis description	**In-line**
Bin grid origin I coordinate =	**1.0**
Bin grid origin J coordinate =	**1.0**
Bin grid origin Easting =	**456781.0**
Bin grid origin Northing =	**5836723.0**
Scale factor of bin grid =	**0.99984**
Scale factor node I coordinate =	**1.0**
Scale factor node J coordinate =	**1.0**
Nominal bin width on I axis =	**25.0**
Nominal bin width on J axis =	**12.5**
Grid bearing of bin grid J axis =	**20**
Grid bearing unit name =	**degree**
Bin node increment on I axis =	**1**
Bin node increment on J axis =	**1**
First check node I coordinate =	**334.0000**
First check node J coordinate =	**235.0000**
First check node Easting =	**465602.94**
First check node Northing =	**5836624.30**
Second check node I coordinate =	**1352.0000**
Second check node J coordinate =	**955.0000**
Second check node Easting =	**492591.98**
Second check node Northing =	**5836377.16**
Third check node I coordinate =	**605.0000**
Third check node J coordinate =	**955.0000**
Third check node Easting =	**475046.03**
Third check node Northing =	**5842763.36**

D-5. Data Geographic Extent & Coverage Perimeter (deprecated)

The content of the Data Geographic Extent and Coverage Perimeter stanzas follow the provisions of the UKOOA P6/98 format or the newer OGP P6/11. As with the location stanzas, the relevant OGP format (P1/11 or P6/11) is recommended when backward compatibility is not an issue. See Appendices D-1 and D-2 for an explanation of the preferred format.

The Data Geographic Extent stanza describes the geographical extent of data in bin grid, projected (map grid) and/or geographical (latitude/longitude) coordinates. The Coverage Perimeter stanza describes the perimeter of a 3-D data set. The geographical and projected (map grid) coordinate reference systems must be defined in the Location Data stanza (see Appendix D-3). The bin grid must be defined in a Bin Grid Definition stanza (see Appendix D-4).

The Coverage Perimeter stanza allows for the description of the following coverages:

- The total coverage of all data within the data set through the coordinates of a series of points describing the perimeter of the total coverage.
- Full-fold coverage through the coordinates of a series of points describing the outer perimeter of the full-fold coverage.
- Islands within the full-fold coverage with less than full-fold through the coordinates of a series of points describing the outer perimeter of the null full-fold coverage.
- Islands within the total coverage within which there is no coverage through the coordinates of a series of points describing the outer perimeter of the null fold coverage.

Figure 4 describes these concepts by showing the various data extents and coverage perimeters for a seismic survey encompassing a platform undershoot.

For processed data sets (near-trace cubes, migrated volumes, etc.), the fold will be affected by various processing steps (trace summation, offset rejection, migration, etc.). These processed data sets can be represented by either a Total Coverage Perimeter or a Full-fold Perimeter. The type of processed data set should be stated using the coverage extent comment keyword.

Wherever a detailed coverage perimeter is known for a data set, the perimeter should be included in the exchange file. Bin grid and/or map grid coordinates may be given for each node of each perimeter. The data set extent can then be easily derived from the detailed perimeter. However, given the practical importance of the data set extent (e.g. used for loading of data onto workstations), the extent should also be defined explicitly in bin grid, map grid and/or latitude and longitude terms through a Data Geographic Extent stanza.

The Data Geographic Extent provides the user with a simple representation of the area covered by the survey for mapping and data management purposes, rather than a precise representation of the fold of coverage of a binning system or process.

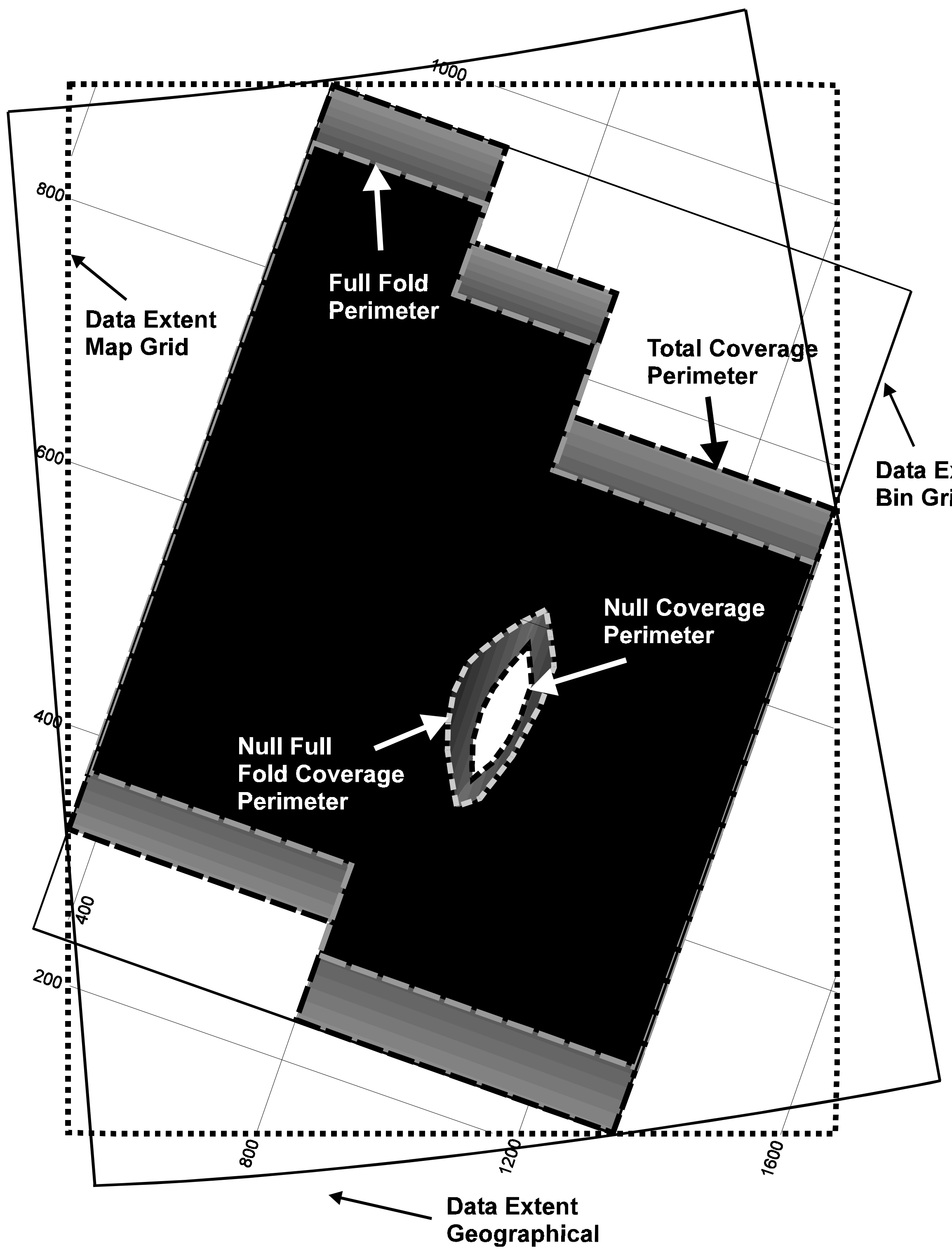

Figure 4 Various data extents and coverage perimeters for a seismic survey

D-5.1 Stanza for Data Geographic Extent

The Data Geographic Extent stanza describes the geographical extent of data in bin grid, map grid and/or geographical (latitude/longitude) coordinates. The coordinate reference system for the geographical coordinates and derived projected (map grid) coordinates must be given in the Location Data stanza (See Appendix D-3). Note that where backward compatibility is not an issue this method is replaced by the data geographic extent facility in the OGP P1/11 format (see section D-1.12).

Table 12 Stanza for Data Geographic Extent

Stanza Header and Keyword	Format	Comment
((SEG: Data Geographic Extent ver 1.0))	Text	Stanza name
Minimum Easting =	Real Number	Map grid Easting of the bin node with the minimum value of map grid Easting.
Maximum Easting =	Real Number	Map grid Easting of the bin node with the maximum value of map grid Easting.
Minimum Northing =	Real Number	Map grid Northing of the bin node with the minimum value of map grid Northing.
Maximum Northing =	Real Number	Map grid Northing of the bin node with the maximum value of map grid Northing.
Minimum Latitude =	Real Number	Latitude of the bin node with the minimum latitude value.
Maximum Latitude =	Real Number	Latitude of the bin node with the maximum latitude value.
Minimum Longitude =	Real Number	Longitude of the bin node with the minimum longitude value.
Maximum Longitude =	Real Number	Longitude of the bin node with the maximum longitude value.
Geographical coordinate unit name =	Text	The name of the units in which the maximum and minimum latitude and longitude are given.
Minimum I coordinate =	Real Number	Bin grid I coordinate of the bin node with the minimum value of I coordinate.
Maximum I coordinate =	Real Number	Bin grid I coordinate of the bin node with the maximum value of I coordinate.
Minimum J coordinate =	Real Number	Bin grid J coordinate of the bin node with the minimum value of J coordinate.
Maximum J coordinate =	Real Number	Bin grid J coordinate of the bin node with the maximum value of J coordinate.

Stanza Header and Keyword	Format	Comment
Data Extent comment =	Text	May be repeated as necessary.

D-5.2 Example for Data Geographic Extent

((SEG: Data Geographic Extent ver 1.0))	
Minimum Easting =	**465966.28**
Maximum Easting =	**491792.63**
Minimum Northing =	**5827921.28**
Maximum Northing =	**5845080.18**
Minimum Latitude =	**52.4516782**
Maximum Latitude =	**52.3604359**
Minimum Longitude =	**2.3209385**
Maximum Longitude =	**2.5243181**
Geographical coordinate unit name =	**DDD.MMSSsss**
Minimum I coordinate =	**334.0000**
Maximum I coordinate =	**1352.0000**
Minimum J coordinate =	**235.0000**
Maximum J coordinate =	**955.0000**
Data Extent comment =	**UKOOA P6/98 example. Note: because the example given here is taken from UKOOA the CRS referred to (WGS 84 / UTM zone 31N) is not consistent with the example shown in section D-3.1. However in usage the CRS identified in a Location Data stanza would be expected to apply to the Data Geographic Extent stanza.**

D-5.3 Stanza for Coverage Perimeter

The Coverage Perimeter stanza describes the perimeter of a 3-D data set in bin grid and/or map grid coordinates. When map grid coordinates are given the coordinate reference system for the projected (map grid) coordinates must be given in the Location Data stanza (See Appendix D-3). Note that where backward compatibility is not an issue this method is replaced by the coverage perimeter facility in the OGP P6/11 format (see Appendix D-2).

Table 13 Stanza for Coverage Perimeter

Stanza Header and Keyword	Format	Comment
((SEG: Coverage Perimeter ver 1.0))	Text	Stanza name
Coverage type =	From enumerated list: total full-fold null full-fold null fold	See section D-3 preamble for description of enumerated coverage types.

Stanza Header and Keyword	Format	Comment
Perimeter coordinate type =	From enumerated list: I,J E,N I,J,E,N	The coverage perimeter may be described by bin grid and/or map grid coordinates. The provision of both bin grid and map grid is encouraged.
Perimeter node number =	Integer	The number of nodes describing the perimeter. For an n-sided perimeter the perimeter node number should be n.
Perimeter node coordinates =	2 or 4 comma-separated real numbers	I, J and/or Easting, Northing coordinates of the bin node. Repeat the first point at the end of the list: for an n-sided perimeter the perimeter node coordinate record count should be n+1.
Coverage Perimeter comment =	Text	May be repeated as necessary.

D-5.4 Example Stanza for Coverage Perimeter

This example is based on Figure 4

```
((SEG: Coverage Perimeter ver 1.0))
Coverage type =                    total
Perimeter coordinate type =        I,J,E,N
Perimeter node number =            10
Perimeter node coordinates =       334.0000,955.0000,468680.63,5845080.18
Perimeter node coordinates =       654.0000,955.0000,476196.97,5842344.46
Perimeter node coordinates =       654.0000,875.0000,475855.00,5841404.91
Perimeter node coordinates =       900.0000,875.0000,481633.18,5839301.83
Perimeter node coordinates =       900.0000,768.0000,481175.81,5838045.19
Perimeter node coordinates =       1352.0000,768.0000,491792.63,5834180.98
Perimeter node coordinates =       1352.0000,235.0000,489514.29,5827921.28
Perimeter node coordinates =       802.0000,235.0000,476595.58,5832623.30
Perimeter node coordinates =       802.0000,320.0000,476958.92,5833621.57
Perimeter node coordinates =       334.0000,320.0000,465966.28,5837622.56
Perimeter node coordinates =       334.0000,955.0000,468680.63,5845080.18

((SEG: Coverage Perimeter ver 1.0))
Coverage type =                    full-fold
Perimeter coordinate type =        I,J
Perimeter node number =            10
Perimeter node coordinates =       334.0000,908.0000
Perimeter node coordinates =       654.0000,908.0000
Perimeter node coordinates =       654.0000,833.0000
Perimeter node coordinates =       900.0000,833.0000
```

Perimeter node coordinates = 900.0000,721.0000
Perimeter node coordinates = 1352.0000,721.0000
Perimeter node coordinates = 1352.0000,289.0000
Perimeter node coordinates = 802.0000,289.0000
Perimeter node coordinates = 802.0000,368.0000
Perimeter node coordinates = 334.0000,368.0000
Perimeter node coordinates = 334.0000,908.0000
Coverage Perimeter comment = 48 fold data
((SEG: Coverage Perimeter ver 1.0))
Coverage type = null full-fold
Perimeter coordinate type = E,N
Perimeter node number = 9
Perimeter node coordinates = 482101.92,5835620.00
Perimeter node coordinates = 482874.75,5834820.00
Perimeter node coordinates = 482067.29,5834063.19
Perimeter node coordinates = 481388.11,5833804.99
Perimeter node coordinates = 480572.36,5833902.39
Perimeter node coordinates = 479705.57,5834736.58
Perimeter node coordinates = 479274.40,5835452.12
Perimeter node coordinates = 479633.25,5835707.21
Perimeter node coordinates = 480739.50,5835823.27
Perimeter node coordinates = 482101.92,5835620.00

((SEG: Coverage Perimeter ver 1.0))
Coverage type = null fold
Perimeter coordinate type = I,J,E,N
Perimeter node number = 8
Perimeter node coordinates = 958.0000,579.0000,481730.25,5835329.67
Perimeter node coordinates = 978.0000,552.0000,482084.61,5834841.59
Perimeter node coordinates = 980.0000,512.0000,481960.60,5834354.72
Perimeter node coordinates = 958.0000,481.0000,481311.34,5834178.73
Perimeter node coordinates = 946.0000,468.0000,480973.91,5834128.64
Perimeter node coordinates = 900.0000,498.0000,480021.67,5834874.23
Perimeter node coordinates = 920.0000,522.0000,480594.03,5834985.11
Perimeter node coordinates = 958.0000,582.0000,481743.07,5835364.90
Perimeter node coordinates = 958.0000,579.0000,481730.25,5835329.67

D-6. Data Sample Measurement Unit

D-6.1 Stanza for Data Sample Measurement Unit

The Data Sample Measurement Unit stanza provides a means of defining a measurement unit other than the measurement units defined in SEG-Y Standard Trace Header bytes 203–204.

Table 14 Stanza for Data Sample Measurement Unit

Stanza and Keyword	Format	Comment
((SEG: Data Sample Measurement Unit ver 1.0))	Text	Stanza name
Data Sample Measurement Unit =	Text	A textual description of the measurement unit used for the data samples (i.e. millivolts, meters)
Volt conversion =	Real Number	The multiplicative constant that converts the Data Sample Measurement Unit to Volts.

D-6.2 Example stanza for Data Sample Measurement Unit

```
((SEG: Data Sample Measurement Unit ver 1.0))
Data Sample Measurement Unit =          Millivolts
Volt conversion =                       0.001
```

D-7. Processing History

The Processing History stanza provides a means to track the processing history of the seismic data traces.

D-7.1 Stanza for Processing History

Table 15 Stanza for Processing History

Stanza and Keyword	Format	Comment
((SEG: Processing History ver 1.0))	Text	Stanza name
The following six entries are repeated as needed to define all processing steps applied to the data traces.		
Processing Company =	Text	
Processing Software =	Text	
Input Data Set =	Text	Data set name or data set id of the data traces being processed

Stanza and Keyword	Format	Comment
Processing Date =	Text	Date in YYYYMMDD-HHMMSS format
Process Applied =	Text	Name of the algorithm or program being applied to the data traces
Process Parameters =	Text	

D-7.2 Example stanza for Processing History

((SEG: Processing History ver 1.0))

Processing Company =	**Expert Processing Inc.**
Processing Software =	**Omega**
Input Data Set =	**$ADBigDo_FieldSeq463**
Processing Date =	**20010519-231643**
Processing Applied =	**SEG-D edit**
Process Parameters =	**MP factor applied**
Processing Company =	**Expert Processing Inc.**
Processing Software =	**Omega**
Input Data Set =	**$ADBigDo_FieldSeq463**
Processing Date =	**20010519-231643**
Processing Applied =	**Trace select/sort**
Process Parameters =	**Data traces, Common Rcv Sort**
Processing Company =	**Expert Processing Inc.**
Processing Software =	**Omega**
Input Data Set =	**$ADBigDo_FieldSeq463_Edit**
Processing Date =	**20010520-115959**
Processing Applied =	**Predictive deconvolution**
Process Parameters =	**Surface consistent, 130 ms, 3 windows**

In this example, a field data set was edited and sorted into common receiver order as an initial process. In a second step, the edited data was deconvolved using a surface consistent deconvolution operator.

D-8. Source Type/Orientation

D-8.1 Stanza for Source Type/Orientation

The Source Type stanza allows the source types used during the data acquisition to be uniquely identified. The source type identifier is used in Trace Header bytes 217–218. This stanza is used when the predefined source types in the Trace Header bytes 217–218 do not adequately identify the sources used for acquisition or an expanded description is desired. When a source type is capable of generating energy in multiple orientations, a Source Type/Orientation stanza should be defined for each orientation.

Table 16 Stanza for Source Type/Orientation

Stanza and Keyword	Format	Comment
((SEG: Source Type/Orientation ver 1.0))	Text	Stanza name
Source description =	Text	A textual description of the source.
Source description (continued 1) =	Text	A textual description of the source.
Source description (continued 2) =	Text	A textual description of the source.
Source type identifier =	Negative Integer	The negative integer that will be used in SEG-Y Standard Trace Header bytes 217–218 to identify this source.

D-8.2 Example stanza for Source Type/Orientation

((SEG: Source Type/Orientation ver 1.0))	
Source Description =	**Inclined Impactor**
Source description (continued) =	**80-45 -45 incident angle**
Source description (continued) =	
Source type identifier =	**-6**
((SEG: Source Type/Orientation ver 1.0))	
Source Description =	**Inclined Impactor**
Source description (continued) =	**100-135-135 incident angle**
Source description (continued) =	
Source type identifier =	**-7**
((SEG: Source Type/Orientation ver 1.0))	
Source Description =	**Mini-shallow water air gun**
Source description (continued) =	**182 ci at 10,000 psi**
Source description (continued) =	
Source type identifier =	**-8**

D-9. Source Measurement Unit

D-9.1 Stanza for Source Measurement Unit

The Source Measurement Unit stanza provides a means of defining a measurement unit other than the measurement units defined in the Trace Header bytes 231-232.

Table 17 Stanza for Source Measurement Unit

Stanza and Keyword	Format	Comment
((SEG: Source Measurement Unit ver 1.0))	Text	Stanza name
Source Measurement Unit =	Text	A textual description of the measurement unit used for the source measurement (i.e. joules, millivolts, meters, vibrators, kilograms of dynamite, etc.)

Stanza and Keyword	Format	Comment
Joule conversion =	Real Number	The multiplicative constant that converts the Source Measurement Unit to Joules. Specify the value of zero if the Source Measurement Unit cannot be converted to joules

D-9.2 Example stanza for Source Measurement Unit

((SEG: Source Measurement Unit ver 1.0))
Source Measurement Unit = **Vibrators * sweep length in seconds**
Joule conversion = **0.0**

D-10. Stanza for Trace Header Mapping

The Trace Header Mapping stanza provides definitions for user-defined trace 240-byte header extensions and overrides of SEG-defined trace header standard mappings.

Table 18 Stanza for Trace Header Mapping

Stanza and Keyword	Format	Comment
((SEG: Trace Header Mapping ver 1.0))	Text	Stanza name
Header name =	Text 8	The name that appears in the last 8 bytes of the trace header being mapped. SEG00000 represents the initial Standard SEG-Y Trace Header. It is suggested that these stanzas occur in the same sequence as the trace headers being mapped.
Delete field name =	Text	The name of an existing trace header field definition to be deleted. This optional keyword is provided to allow removal or replacement of a predefined trace header field. It is suggested that all “Delete field name” keywords appear immediately after the “Header name” entry.
Field name 1 =	Text	The name assigned to a new trace header field. If omitted, a name is autogenerated from the “Header name” and the field’s byte location, e.g. USER0001.21 for the field starting in byte 21 of header USER0001.

Stanza and Keyword	Format	Comment
Byte position 1 =	Integer	The starting byte position of new trace header field 1. This must be in the range 1 to 240 and is required.
Value format code 1 =	Integer	The format code as listed for bytes 3225-3226 of the 400-byte Binary File Header. Use code 0 for ASCII or UTF-8 text. This keyword is required.
Number of values 1 =	Integer	The number of values provided in new trace header 1. If omitted, 1 is assumed. Byte position 1 plus the Number of values times the length of the Value format must not exceed 241.
Field description 1 =	Text	Description of the trace header field.
…	…	…
Field name N =	Text	The name assigned to a new trace header field. If omitted, a name is autogenerated from the "Header name" and the field's byte location, e.g. USER0001.211 for the field in byte 211 of header USER0001.
Byte position N =	Integer	The starting byte position of new trace header field N. This must be in the range 1 to 240 and is required.
Value format code N =	Integer	The format code as listed for bytes 3225-3226 of the 400-byte Binary File Header. Use code 0 for ASCII or UTF-8 text. This keyword is required.
Number of values N =	Integer	The number of values provided in new trace header N. If omitted, 1 is assumed. Byte position 1 plus the Number of values times the length of the Value format must not exceed 241.
Field description N =	Text	Description of the trace header field.

D-10.1 Example Stanzas for Trace Header Mapping

Stanzas for the two currently predefined SEG-Y trace headers are defined below. The field names have been chosen, where feasible, to match those used in the widely used open source SU seismic processing package from the Colorado School of Mines Center for Wave Phenomena. Fields may also be referred to as SEG00000.x and SEG00001.x respectively, irrespective of these predefined names.

((SEG: Trace Header Mapping ver 1.0))

Header name =	**SEG00000**
Field name 1 =	**tracl**
Byte position 1 =	**1**
Value format code 1 =	**2**
Field description 1 =	**Trace sequence number within line**
Field name 2 =	**tracr**
Byte position 2 =	**5**
Value format code 2 =	**2**
Field description 2 =	**Trace sequence number within SEG-Y file**
Field name 3 =	**fldr**
Byte position 3 =	**9**
Value format code 3 =	**2**
Field description 3 =	**Original field record number**
Field name 4 =	**tracf**
Byte position 4 =	**13**
Value format code 4 =	**2**
Field description 4 =	**Trace number within the original field record**
Field name 5 =	**ep**
Byte position 5 =	**17**
Value format code 5 =	**2**
Field description 5 =	**Energy source point number**
Field name 6 =	**cdp**
Byte position 6 =	**21**
Value format code 6 =	**2**
Field description 6 =	**Ensemble number**
Field name 7 =	**cdpt**
Byte position 7 =	**25**
Value format code 7 =	**2**
Field description 7 =	**Trace number within the ensemble**
Field name 8 =	**trid**
Byte position 8 =	**29**
Value format code 8 =	**3**
Field description 8 =	**Trace identification code**
Field name 9 =	**nvs**
Byte position 9 =	**31**
Value format code 9 =	**3**
Field description 9 =	**Number of vertically summed traces yielding this trace**
Field name 10 =	**nhs**
Byte position 10 =	**33**
Value format 10 =	**3**
Field description 10 =	**Number of horizontally stacked traces yielding this trace**
Field name 11 =	**duse**
Byte position 11 =	**35**
Value format 11 =	**3**
Field description 11 =	**Data use**

Field name 12 = **offset**
Byte position 12 = **37**
Value format 12 = **2**
Field description 12 = **Distance from center of the source point to the center of the receiver group**
Field name 13 = **gelev**
Byte position 13 = **41**
Value format 13 = **2**
Field description 13 = **Receiver group elevation**
Field name 14 = **selev**
Byte position 14 = **45**
Value format 14 = **2**
Field description 14 = **Surface elevation at source**
Field name 15 = **sdepth**
Byte position 15 = **49**
Value format 15 = **2**
Field description 15 = **Source depth below surface**
Field name 16 = **gdel**
Byte position 16 = **53**
Value format 16 = **2**
Field description 16 = **Seismic Datum elevation at receiver group**
Field name 17 = **sdel**
Byte position 17 = **57**
Value format 17 = **2**
Field description 17 = **Seismic Datum elevation at source**
Field name 18 = **swdep**
Byte position 18 = **61**
Value format 18 = **2**
Field description 18 = **Water column height at source location**
Field name 19 = **gwdep**
Byte position 19 = **65**
Value format 19 = **2**
Field description 19 = **Water column height at receiver group location**
Field name 20 = **scalel**
Byte position 20 = **69**
Value format 20 = **3**
Field description 20 = **Scalar to be applied to all elevations and depths in fields 13-19 and any extension in SEG-Y Trace Header SEG00001.**
Field name 21 = **scalco**
Byte position 21 = **71**
Value format 21 = **3**
Field description 21 = **Scalar to be applied to all coordinates specified in fields 22-25, 72-73, and any extension in SEG-Y Trace Header SEG00001.**
Field name 22 = **sx**
Byte position 22 = **73**
Value format 22 = **2**
Field description 22 = **Source coordinate - X**
Field name 23 = **sy**
Byte position 23 = **77**
Value format 23 = **2**
Field description 23 = **Source coordinate - Y**
Field name 24 = **gx**
Byte position 24 = **81**
Value format 24 = **2**
Field description 24 = **Group coordinate - X**

Field name 25 =	**gy**
Byte position 25 =	**85**
Value format 25 =	**2**
Field description 25 =	**Group coordinate - Y**
Field name 26 =	**counit**
Byte position 26 =	**89**
Value format 26 =	**3**
Field description 26 =	**Coordinate units**
Field name 27 =	**wevel**
Byte position 27 =	**91**
Value format 27 =	**3**
Field description 27 =	**Weathering velocity**
Field name 28 =	**swevel**
Byte position 28 =	**93**
Value format 28 =	**3**
Field description 28 =	**Subweathering velocity**
Field name 29 =	**sut**
Byte position 29 =	**95**
Value format 29 =	**3**
Field description 29 =	**Uphole time at source**
Field name 30 =	**gut**
Byte position 30 =	**97**
Value format 30 =	**3**
Field description 30 =	**Uphole time at group**
Field name 31 =	**sstat**
Byte position 31 =	**99**
Value format 31 =	**3**
Field description 31 =	**Source static correction**
Field name 32 =	**gstat**
Byte position 32 =	**101**
Value format 32 =	**3**
Field description 32 =	**Group static correction**
Field name 33 =	**tstat**
Byte position 33 =	**103**
Value format 33 =	**3**
Field description 33 =	**Total static applied**
Field name 34 =	**laga**
Byte position 34 =	**105**
Value format 34 =	**3**
Field description 34 =	**Lag time A**
Field name 35 =	**lagb**
Byte position 35 =	**107**
Value format 35 =	**3**
Field deswcription 35 =	**Lag time B**
Field name 36 =	**delrt**
Byte position 36 =	**109**
Value format 36 =	**3**
Field description 36 =	**Delay recording time**
Field name 37 =	**muts**
Byte position 37 =	**111**
Value format 37 =	**3**
Field description 37 =	**Mute time start**

Field name 38 =	**mute**
Byte position 38 =	**113**
Value format 38 =	**3**
Field description 38 =	**Mute time end**
Field name 39 =	**ns**
Byte position 39 =	**115**
Value format 39 =	**11**
Field description 39 =	**Number of samples in this trace**
Field name 40 =	**dt**
Byte position 40 =	**117**
Value format 40 =	**11**
Field description 40 =	**Sample interval for this trace**
Field name 41 =	**gain**
Byte position 41 =	**119**
Value format 41 =	**3**
Field description 41 =	**Gain type of field instruments**
Field name 42 =	**igc**
Byte position 42 =	**121**
Value format 42 =	**3**
Field description 42 =	**Instrument gain constant**
Field name 43 =	**igi**
Byte position 43 =	**123**
Value format 43 =	**3**
Field description 43 =	**Instrument early or initial gain**
Field name 44 =	**corr**
Byte position 44 =	**125**
Value format 44 =	**3**
Field description 44 =	**Correlated**
Field name 45 =	**sfs**
Byte position 45 =	**127**
Value format 45 =	**3**
Field description 45 =	**Sweep frequency at start**
Field name 46 =	**sfe**
Byte position 46 =	**129**
Value format 46 =	**3**
Field description 46 =	**Sweep frequency at end**
Field name 47 =	**slen**
Byte position 47 =	**131**
Value format 47 =	**3**
Field description 47 =	**Sweep length**
Field name 48 =	**styp**
Byte position 48 =	**133**
Value format 48 =	**3**
Field description 48 =	**Sweep type**
Field name 49 =	**stas**
Byte position 49 =	**135**
Value format 49 =	**3**
Field description 49 =	**Sweep trace taper length at start**
Field name 50 =	**stae**
Byte position 50 =	**137**
Value format 50 =	**3**
Field description 50 =	**Sweep trace taper length at end**

Field name 51 =	**tatyp**
Byte position 51 =	**139**
Value format 51 =	**3**
Field description 51 =	**Taper type**
Field name 52 =	**afilf**
Byte position 52 =	**141**
Value format 52 =	**3**
Field description 52 =	**Alias filter frequency, if used**
Field name 53 =	**afils**
Byte position 53 =	**143**
Value format 53 =	**3**
Field description 53 =	**Alias filter slope**
Field name 54 =	**nofilf**
Byte position 54 =	**145**
Value format 54 =	**3**
Field description 54 =	**Notch filter frequency, if used**
Field name 55 =	**nofils**
Byte position 55 =	**147**
Value format 55 =	**3**
Field description 55 =	**Notch filter slope**
Field name 56 =	**lcf**
Byte position 56 =	**149**
Value format 56 =	**3**
Field description 56 =	**Low-cut frequency, if used**
Field name 57 =	**hcf**
Byte position 57 =	**151**
Value format 57 =	**3**
Field description 57 =	**High-cut frequency, if used**
Field name 58 =	**lcs**
Byte position 58 =	**153**
Value format 58 =	**3**
Field description 58 =	**Low-cut slope**
Field name 59 =	**hcs**
Byte position 59 =	**155**
Value format 59 =	**3**
Field description 59 =	**High-cut slope**
Field name 60 =	**year**
Byte position 60 =	**157**
Value format 60 =	**3**
Field description 60 =	**Year data recorded**
Field name 61 =	**day**
Byte position 61 =	**159**
Value format 61 =	**3**
Field description 61 =	**Day of year**
Field name 62 =	**hour**
Byte position 62 =	**161**
Value format 62 =	**3**
Field description 62 =	**Hour of day**
Field name 63 =	**minute**
Byte position 63 =	**163**
Value format 63 =	**3**
Field description 63 =	**Minute of hour**

Field name 64 =	**sec**
Byte position 64 =	**165**
Value format 64 =	**3**
Field description 64 =	**Second of minute**
Field name 65 =	**timbas**
Byte position 65 =	**167**
Value format 65 =	**3**
Field description 65 =	**Time basis code**
Field name 66 =	**trwf**
Byte position 66 =	**169**
Value format 66 =	**3**
Field description 66 =	**Trace weighting factor**
Field name 67 =	**grnors**
Byte position 67 =	**171**
Value format 67 =	**3**
Field description 67 =	**Geophone group number of roll switch position one**
Field name 68 =	**grnofr**
Byte position 68 =	**173**
Value format 68 =	**3**
Field description 68 =	**Geophone group number of trace number one within original field record**
Field name 69 =	**grnlof**
Byte position 69 =	**175**
Value format 69 =	**3**
Field description 69 =	**Geophone group number of last trace within original field record**
Field name 70 =	**gaps**
Byte position 70 =	**177**
Value format 70 =	**3**
Field description 70 =	**Gap size**
Field name 71 =	**otrav**
Byte position 71 =	**179**
Value format 71 =	**3**
Field description 71 =	**Over travel associated with taper at beginning or end of line**
Field name 72 =	**cdpx**
Byte position 72 =	**181**
Value format 72 =	**2**
Field description 72 =	**X coordinate of ensemble (CDP) position of this trace**
Field name 73 =	**cdpy**
Byte position 73 =	**185**
Value format 73 =	**2**
Field desciption 73 =	**Y coordinate of ensemble (CDP) position of this trace**
Field name 74 =	**iline**
Byte position 74 =	**189**
Value format 74 =	**2**
Field description 74 =	**3-D poststack data in-line number**
Field name 75 =	**xline**
Byte position 75 =	**193**
Value format 75 =	**2**
Field description 75 =	**3-D poststack data cross-line number**
Field name 76 =	**sp**
Byte position 76 =	**197**
Value format 76 =	**2**
Field description 76 =	**Shotpoint number**

Field name 77 =	**spscal**
Byte position 77 =	**201**
Value format 77 =	**3**
Field description 77 =	**Scalar to be applied to shotpoint number in field 76**
Field name 78 =	**tvmu**
Byte position 78 =	**203**
Value format 78 =	**3**
Field description 78 =	**Time value measurement unit**
Field name 79 =	**trdman**
Byte position 79 =	**205**
Value format 79 =	**2**
Field description 79 =	**Transduction constant mantissa**
Field name 80 =	**trdexp**
Byte position 80 =	**209**
Value format 80 =	**3**
Field description 80 =	**Transduction constant exponent**
Field name 81 =	**trdun**
Byte position 81 =	**211**
Value format 81 =	**3**
Field description 81 =	**Transduction Units**
Field name 82 =	**dti**
Byte position 82 =	**213**
Value format 82 =	**3**
Field description 82 =	**Device/Trace Identifier**
Field name 83 =	**timscal**
Byte position 83 =	**215**
Value format 83 =	**3**
Field description 83 =	**Scalar to be applied to times in fields 29-38 and any extension in SEG-Y Trace Header SEG00001**
Field name 84 =	**stypor**
Byte position 84 =	**217**
Value format 84 =	**3**
Field description 84 =	**Source Type/Orientation**
Field name 85 =	**sedir**
Byte position 85 =	**219**
Value format 85 =	**3**
Number of values 85 =	**3**
Field description 85 =	**Source Energy Direction with respect to the source orientation**
Field name 86 =	**smman**
Byte position 86 =	**225**
Value format 86 =	**2**
Field description 86 =	**Source Measurement mantissa**
Field name 87 =	**smexp**
Byte position 87 =	**229**
Value format 87 =	**3**
Field description 87 =	**Source Measurement exponent**
Field name 88 =	**smun**
Byte position 88 =	**231**
Value format 88 =	**3**
Field description 88 =	**Source Measurement Unit**

((SEG: Trace Header Mapping ver 1.0))

Header name =	**SEG00001**
Field name 1 =	**etracl**
Byte position 1 =	**1**
Value format 1 =	**12**
Field description 1 =	**Extended trace sequence number within line**
Field name 2 =	**etracr**
Byte position 2 =	**9**
Value format 2 =	**12**
Field description 2 =	**Extended trace sequence number within SEG-Y file**
Field name 3 =	**efldr**
Byte position 3 =	**17**
Value format 3 =	**9**
Field description 3 =	**Extended original field record number**
Field name 4 =	**ecdp**
Byte position 4 =	**25**
Value format 4 =	**9**
Field description 4 =	**Extended ensemble number**
Field name 5 =	**egelev**
Byte position 5 =	**33**
Value format 5 =	**6**
Field description 5 =	**Extended elevation of receiver group**
Field name 6 =	**gdepth**
Byte position 6 =	**41**
Value format 6 =	**6**
Field description 6 =	**Receiver group depth below surface**
Field name 7 =	**eselev**
Byte position 7 =	**49**
Value format 7 =	**6**
Field description 7 =	**Extended surface elevation at source**
Field name 8 =	**esdepth**
Byte position 8 =	**57**
Value format 8 =	**6**
Field description 8 =	**Extended source depth below surface**
Field name 9 =	**egdel**
Byte position 9 =	**65**
Value format 9 =	**6**
Field description 9 =	**Extended Seismic Datum elevation at receiver group**
Field name 10 =	**esdel**
Byte position 10 =	**73**
Value format 10 =	**6**
Field description 10 =	**Extended Seismic Datum elevation at source**
Field name 11 =	**eswdep**
Byte position 11 =	**81**
Value format 11 =	**6**
Field description 11 =	**Extended water column height at source location**
Field name 12 =	**egwdep**
Byte position 12 =	**89**
Value format 12 =	**6**
Field description 12 =	**Extended water column height at receiver group location**
Field name 13 =	**esx**
Byte position 13 =	**97**
Value format 13 =	**6**
Field description 13 =	**Extended source coordinate — X**

Field name 14 =	**esy**
Byte position 14 =	**105**
Value format 14 =	**6**
Field description 14 =	**Extended source coordinate — Y**
Field name 15 =	**egx**
Byte position 15 =	**113**
Value format 15 =	**6**
Field description 15 =	**Extended group coordinate — X**
Field name 16 =	**egy**
Byte position 16 =	**121**
Value format 16 =	**6**
Field description 16 =	**Extended group coordinate — Y**
Field name 17 =	**eoffset**
Byte position 17 =	**129**
Value format 17 =	**6**
Field description 17 =	**Extended distance center of source to center of receiver group**
Field name 18 =	**ens**
Byte position 18 =	**137**
Value format 18 =	**10**
Field description 18 =	**Extended number of samples in this trace**
Field name 19 =	**secfrac**
Byte position 19 =	**141**
Value format 19 =	**2**
Field description 19 =	**Nanoseconds to add to Second of minute**
Field name 20 =	**edt**
Byte position 20 =	**145**
Value format 20 =	**6**
Field description 20 =	**Extended sample interval in microseconds**
Field name 21 =	**cable**
Byte position 21 =	**153**
Value format 21 =	**2**
Field description 21 =	**Cable number for multicable acquisition**
Field name 22 =	**nthe**
Byte position 22 =	**157**
Value format 22 =	**11**
Field description 22 =	**Number of trace header extensions**
Field name 23 =	**lasttr**
Byte position 23 =	**159**
Value format 23 =	**11**
Field description 23 =	**Last trace flag**
Field name 24 =	**ecdpx**
Byte position 24 =	**161**
Value format 24 =	**6**
Field description 24 =	**extended X coordinate of ensemble (CDP) position of this trace**
Field name 25 =	**ecdpy**
Byte position 25 =	**169**
Value format 25 =	**6**
Field description 25 =	**extended Y coordinate of ensemble (CDP) position of this trace**

D-11. User Data stanza

The User Data stanza **((SEG:UserData))** contains a set of user-defined blocks separated by a description block identifying the producer, contents, size and data format of the data block following it.

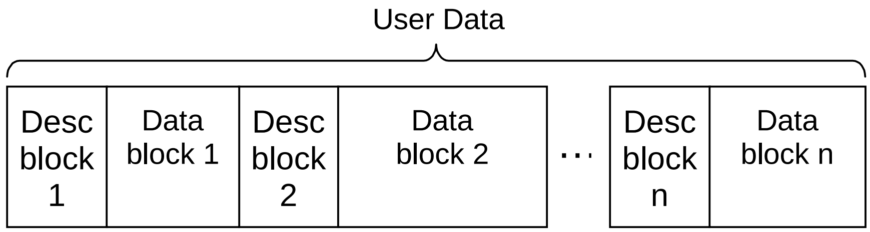

Figure 5 Structure of a User Data stanza containing n data blocks

For efficiency we recommended the User Data stanza in the Extended Textual Header be used for information that is required for interpreting the data traces. Large data blocks like backup of positioning data, databases etc. should be stored in the User Data trailer stanza. This will simplify decoding and interpretation of, and speed up access to, the data.

The description block is a well-formed XML document with a strictly defined format providing information about the following data block. It must be written in ASCII text (1 byte characters), and contain English text only. This simplifies decoding of the description block on all platforms and in any locale for all readers of the SEG-Y format.

The data blocks can be any size—there is no padding between blocks.

The description block follows directly after the last data byte of the previous block.

The last data block is padded with binary 0's to fill the last 3200 byte record if necessary.

The data block following the description block can be of any format and have any contents in any language (binary, text, Unicode, big-endian, little-endian etc.)

The format is designed to make it easy to append and insert new blocks, and delete existing blocks. This allows systems to add information to the SEG-Y record as processing stages refines and updates the data.

Examples of data that can be stored in the User Data stanza are information about survey, contract, processing system, edits, trace data description, parameters and deliverables, processing notes - decisions and result evaluation, etc.

The User Data stanza also allows bundling of any meta data with the seismic traces/cubes (e.g. database files, velocity models, processing scripts, observer logs). This can be used to automate and simplify the workflows, and improve consistency, correctness, robustness and repeatability of the seismic processing and interpretation.

The trailer can also be used as a backup of information like positioning files (P1, P6, SPS etc.) or equipment/sensor information (from e.g. SEG D).

D-11.1 Description block

The description block is a well-formed XML 1.0 compatible[27] document that describes the following data block. It must be written in ASCII (1 byte characters), and contain only English text. This is done to simplify decoding of the description block on all platforms (no endian or encoding issues) and simplify parsing in any language by all readers of the SEG-Y format.

The XML structure is also simplified to allow decoding by software without implementing a full XML reader.

The XML structure is single level only, and contains only tags, attributes are not allowed. Also comments and other special tags are not allowed (<?> <!> etc.).

The XML structure may contain line change (ASCII CR($0D_{16}$) or CRLF($0D0A_{16}$)) and indentation (ASCII SPACE(20_{16}) and TAB(09_{16})) to improve human readability, though this is not required.

Tags are written in lower case. All values are case insensitive.

The following tags are supported in the description block, other tags are not allowed:

Tag	Description	Required
segydescblock	Top level tag. If the User header or Data trailer contains errors readers can search for this tag to find the start of next block.	X
version	Version number of description block format, this table describes format version **1**	X
blocksize	Size of data block in bytes.	X
dataformat	Short string describing the format of the data in the data block Starts with *text-* for textual data, and *bin-* for binary data. Text formats can either contain encoding information in the datablock, or specify it explicitly in the dataformat tag (ex: text-utf-8). Html, xml and xhtml are examples of text formats that contain encoding information in the data itself. Encoding follows the html standard for naming (us-ascii, utf-8, ISO-8859-1, iso-8859-5, x-euc-jp etc.) If the byte order/endian is required (the data format does not imply byte order, or specify it in the data format itself) *be-* and *le-* is used to indicate byte order. Some examples below may list multiple similar type dataformats in one table cell for efficiency reasons, but one block can only have one dataformat. Examples (this is just a set of examples, not a complete list):	X

[27] See http://www.w3.org/TR/2008/REC-xml-20081126/ for a suitable XML 1.0 reference.

dataformat	**Description**
text-us-ascii	ASCII text
text-xml	XML text containing definition of encoding (in e.g. <?xml> tag)
text-be-utf-8	UTF 8 big endian unicode text
text-ISO-8859-1	ISO-8859-1 (latin1) Western European text
text-p1/11	P1/11 file
text-html	HTML text containing definition of encoding (in e.g. <meta> tag)
text-other	Other unspecified formatted text. Only use if no other option is available. Readers need to use datalabel, creator and dataformat tags to determine how to decode.

Binary formats will typically use a file format as dataformat code.

Examples (this is just a set of examples, not a complete list):

dataformat	**Description**

bin-doc
bin-docx
bin-xlsx
bin-ppt
bin-ods
bin-odt
bin-odp
bin-pdf

Data block contains a Microsoft Word/Excel/Powerpoint, an Open Office word processor/spreadsheet/presentation or a pdf document.

The content of the document is specified in datalabel/description tags. This may be any document like survey description, contract information, job specifications, observer log, processing notes, result evaluations, information to downstream processing and interpretation, etc.

Note: These are human readable formats only, and the use of computer readable formats is recommended when possible.

bin-hdf5

HDF 5 formatted binary data.

bin-zip
bin-tar

Data block contains multiple files compressed into a single zip or tar.

Useful for backup, but the use of individual blocks for each file is recommended if the archive contains multiple data types.

Refer to datalabel and description tags for more information about actual contents.

bin-jar

Java archive containing executable Java byte code

bin-python

Python script code

bin-iso

CD or DVD ISO-9660 image

bin-mpeg4

MPEG4 encoded video file

bin-mp3

MP3 encoded audio file

bin-sqlite

SQLite database

bin-segd
bin-segy

Data block contains a SEG D or SEG-Y file.

text-myformat
bin-myformat

Format of data block is a format called “myformat”. The format is described in another block.

The labelname tag for the format description block should be TextMyformatDesc or BinMyformatDesc respectively.

The format description is typically a Word/xml/html/text document or similar.

Description tag text should also state which data block (by labelname) contains the format description.

bin-le-other

Unspecified binary little endian format. Typically used for a proprietary binary formats. Readers need to use datalabel, creator and dataformat to determine how to decode. It is recommended to use the “text-myformat/bin-myformat” method described above if possible, and include the format description in another block.

datalabel	The label describes the type of data in a short string. The field can consist of any alphanumeric characters (a-zA-Z0-9). The datalabel is defined by the recorder. Example: SURVEYINFO, VELMODEL. The datalabel must be unique for each creator, i.e. the combination creator and datalabel must identify one and only one type of data block. Data labels starting with the letters *SEG* is reserved for usage by the Society of Exploration Geophysicists.	X
description	A textual description of the contents of the data block. Designed for human consumption and can contain more details about creator and data (who, when, what, why), comments, URL references to more information, etc.	
creationtime	Timestamp for creation of the data block. May be set to the recording time for the SEG-Y record if unknown. Cannot be empty or dummy timestamp. Timestamp format is UTC time (zero GMT time offset) according to ISO 8601 (24 hour clock): Format: **YYYY-MM-DD hh:mm:ssZ** Example: **2015-09-01 22:04:00Z**	X
creator	Name of creator of the data block, usually the name of the system making the data block, for example “NavProc”.	X
dataversion	Version number of the contents of the data block, user defined, for example “1.1”, “2002.b2”. If omitted it implies dataversion “1”, or the data block itself contains a version number.	
compression	If the data block is compressed this specifies the compression method. Proprietary compression methods are allowed, but must be specified by name. Uncompressing by the specified method is necessary to get to the format specified in the dataformat tag. Omit the compression tag or set the value to **none** if data block is not compressed. Examples: zip, bzip2, gzip, compress, snappy	

The combination dataformat, datalabel, dataversion and creator is used to determine how the data should be decoded and used.

D-11.2 Data block

The data block consists of a number of bytes of data as described in the description block. The format of the data block is completely user defined. The data block starts at the first byte after `</segydescblock>`.

D-11.3 Examples

Description block example:

```
<segydescblock>
  <version>1</version>
  <blocksize>16340</blocksize>
  <dataformat>text-xml</dataformat>
  <datalabel>MSPJobParams</datalabel>
  <description>Full set of job parameters in XML format as described in
My Seismic Processing v3.14 User manual Appendix F. Available for
download at
http://www.myseismicproc.com/sw/3.14/downloads.php</description>
  <creationtime>2015-10-30 19:58:17Z</creationtime>
  <creator>My Seismic Processing</creator>
  <dataversion>14</dataversion>
  <compression>none</compression>
</segydescblock>
```

Full User header / Data trailer example consisting of two blocks

Text blocks only as binary data is difficult to write in a document. Padding to fill 3200 byte block is also not shown.

```
((SEG:User Data ver 1.0))
<segydescblock>
  <version>1</version>
  <blocksize>155</blocksize>
  <dataformat>text-xml</dataformat>
  <datalabel>SurvPars</datalabel>
  <description>Short example of survey parameters in xml.</description>
  <creationtime>2015-10-03 00:00:00Z</creationtime>
  <creator>DataProc</creator>
</segydescblock><?xml version=”1.0” encoding=”UTF-8”?>
<dataproc>
<surveyparams version=”3.0”>
  <area>□□</area>
  <contractid>C320012/2015</contractid>
</surveyparams><segydescblock>
  <version>1</version>
  <blocksize>240</blocksize>
  <dataformat>text-iso-8859-1</dataformat>
  <datalabel>ObsLog</datalabel>
  <description>Short textual observer log example</description>
  <creationtime>2015-07-04 09:00:00Z</creationtime>
  <creator>MyAcqSystem</creator>
  <dataversion>7</dataversion>
</segydescblock>Sequence 001:
11:38    Start of line SK15P1001
```

```
12:03    Shots 2001-3200 - Seismic interference from vessel “Støyende
Ørn” NW of survey area
13:18    Missed shots 2155-2157
15:12    End of line SK15P1001
Sequence 002:
17:45    Start of line SK15P1312
```

Example of a user defined block format with description

```
((SEG:User Data ver 1.0))
<segydescblock>
  <version>1</version>
  <blocksize>810221411</blocksize>
  <dataformat>bin-mspprj3</dataformat>
  <datalabel>MSPProjectFile</datalabel>
  <description>My Seismic Processing v3.14 project database for survey
SK1501NS. File format described in block labelled
BinMSPPRJ3Desc.</description>
  <creationtime>2015-10-30 19:58:17Z</creationtime>
  <creator>My Seismic Processing</creator>
  <dataversion>3.14</dataversion>
  <compression>none</compression>
</segydescblock>
…………
<segydescblock>
  <version>1</version>
  <blocksize>154340</blocksize>
  <dataformat>bin-docx</dataformat>
  <datalabel>BinMSPPRJ3Desc</datalabel>
  <description>My Seismic Processing v3.14 project database format
description.</description>
  <creationtime>2015-10-30 19:58:17Z</creationtime>
  <creator>My Seismic Processing</creator>
</segydescblock>
…………
```

P1 file backup using tar and gzip

```
((SEG:User Data ver 1.0))
<segydescblock>
  <version>1</version>
  <blocksize>27320014096</blocksize>
  <dataformat>bin-tar</dataformat>
  <datalabel>MSPPosBackup</datalabel>
  <description>Tar and gzipped file containing all final P1/90 or P1/11
files used for processing this survey</description>
  <creationtime>2015-10-30 19:58:17Z</creationtime>
  <creator>My Seismic Processing</creator>
  <dataversion>1</dataversion>
  <compression>gzip</compression>
</segydescblock>
```

Determine how to decode data block

This is done by examining dataformat, compression, datalabel, dataversion and creator tags in the description block.

The description tag should also provide help information on how to decode

1. Description contains information about how to decode

```
<segydescblock>
  <version>1</version>
  <blocksize>16340</blocksize>
  <dataformat>text-xml</dataformat>
  <datalabel>MSPJobParams</datalabel>
  <description>Full set of job parameters in XML format as described in
My Seismic Processing v3.14 User manual Appendix F. Available for
download at
http://www.myseismicproc.com/sw/3.14/downloads.php</description>
  <creationtime>2015-10-30 19:58:17Z</creationtime>
  <creator>My Seismic Processing</creator>
  <dataversion>14</dataversion>
  <compression>none</compression>
</segydescblock>
```

In this example the data block consists of uncompressed XML text. The contents are job parameters for the My Seismic Processing system. The format of the parameter structure is version 14 as defined by the manufacturer of the MSP system.

If this is already a known format to the reader, it can easily be automatically decoded.

If not the, the description contains information on how to download the format description. The reader may either implement decoding in software, or present the parameter text to the user and do manual interpretation of them.

2. No description text

```
<segydescblock>
  <version>1</version>
  <blocksize>16340</blocksize>
  <dataformat>text-xml</dataformat>
  <datalabel>MSPJobParams</datalabel>
  <creationtime>2015-10-30 19:58:17Z</creationtime>
  <creator>My Seismic Processing</creator>
  <dataversion>14</dataversion>
  <compression>none</compression>
</segydescblock>
```

In this example there are no description showing how to interpret the job parameter format. The reader will then have to try to find the format description somewhere (e.g. by contacting the manufacturers of the My Seismic Processing system), or display the text to the user who can then try to interpret the parameters manually.

For binary formats the result of lack of format description is basically that unknown formats must be ignored by the SEG-Y reader.

It is therefore highly recommended to include information about format decoding in any user defined block.

Appendix E. Data Word Format

This appendix details the coding of values stored in SEG-Y trace samples. To convert these values to the units give by trace header bytes 203–204, multiply by 2–N where N is given by the Trace weighting factor in trace header bytes 169–170. Trace header bytes 205–212 are further used to convert to alternative physical units. If all of bytes 203–212 ore zero, the units after trace weighting should be assumed to be volts unless external information says otherwise.

Code 1 — 4-byte hexadecimal exponent data (i.e. IBM single precision floating point)

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. The exponent has been biased by 64 such that it represents $16^{(CCCCCCC-64)}$ where CCCCCCC can assume values from 0 to 127.

Q_{1-24} = 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 $-1 + 2^{-24}$). If this fraction is zero, the sign and exponent must also be zero (i.e., the entire word is zero).

Value = S.QQQQ,QQQQ,QQQQ,QQQQ,QQQQ,QQQQ x $16^{(CCCCCCC-64)}$

Code 2 — 4-byte, two's complement integer

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

Value = $(I_{31}*2^{31} + I_{30}*2^{30} + \ldots + I_1*2^1 + I_0*2^0 + 2^{31}) \bmod 2^{32} - 2^{31}$

Code 3 — 2-byte, two's complement integer

Bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Byte 1	I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8
Byte 2	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Value = $(I_{15}*2^{15} + I_{14}*2^{14} + \ldots + I_1*2^1 + I_0*2^0 + 2^{15}) \bmod 2^{16} - 2^{15}$

Code 4 — 32-bit fixed point with gain values (*Obsolete*)

Bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Byte 1	0	0	0	0	0	0	0	0
Byte 2	G_7	G_6	G_5	G_4	G_3	G_2	G_1	G_0
Byte 3	S	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

Value = $S\,(I_{14}*2^{14} + I_{13}*2^{13} + \ldots + I_1*2^1 + I_0*2^0) * 2^{-GGGGGGGG}$

Code 5 — 4-byte, IEEE Floating Point

The IEEE format is fully documented in the IEEE standard, "ANSI/IEEE Std 754 - l985", available from the IEEE. The IEEE 32-bit floating point 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 = 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.

Code 6 — 8-byte, IEEE Floating Point

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_{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 $e = 2047$ & $f \neq 0$. .$v = \text{NaN}$ Not-a-Number (see Note 1)
if $e = 2047$ & $f = 0$. . $v = (-1)^s \times \infty$ Overflow
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.

Code 7 — 3-byte, two's complement integer

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}

Value = $(I_{23}*2^{23} + I_{22}*2^{22} + \ldots + I_1*2^1 + I_0*2^0 + 2^{23}) \bmod 2^{24} - 2^{23}$

Code 8 — 1-byte, two's complement integer

Bit	0	1	2	3	4	5	6	7
Byte 1	I_{7}	I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	I_{0}

Value = $(I_7*2^7 + I_6*2^6 + \ldots + I_1*2^1 + I_0*2^0 + 2^7) \bmod 2^8 - 2^7$

Code 9 — 8-byte, two's complement integer

Bit	0	1	2	3	4	5	6	7
Byte 1	I_{63}	I_{62}	I_{61}	I_{60}	I_{59}	I_{58}	I_{57}	I_{56}
Byte 2	I_{55}	I_{54}	I_{53}	I_{52}	I_{51}	I_{50}	I_{49}	I_{48}
Byte 3	I_{47}	I_{46}	I_{45}	I_{44}	I_{43}	I_{42}	I_{41}	I_{40}
Byte 4	I_{39}	I_{38}	I_{37}	I_{36}	I_{35}	I_{34}	I_{33}	I_{32}
Byte 5	I_{31}	I_{30}	I_{29}	I_{28}	I_{27}	I_{26}	I_{25}	I_{24}
Byte 6	I_{23}	I_{22}	I_{21}	I_{20}	I_{19}	I_{18}	I_{17}	I_{16}
Byte 7	I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_{9}	I_{8}
Byte 8	I_{7}	I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	I_{0}

Value = $(I_{63}*2^{63} + I_{62}*2^{62} + \ldots + I_1*2^1 + I_0*2^0 + 2^{63}) \bmod 2^{64} - 2^{63}$

Code 10 — 4-byte, unsigned integer

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

Value = $I_{31}*2^{31} + I_{30}*2^{30} + \ldots + I_1*2^1 + I_0*2^0$

Code 11 — 2-byte, unsigned integer

Bit	0	1	2	3	4	5	6	7
Byte 1	I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8
Byte 2	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Value = $I_{15}*2^{15} + I_{14}*2^{14} + \ldots + I_1*2^1 + I_0*2^0$

Code 12 — 8-byte, unsigned integer

Bit	0	1	2	3	4	5	6	7
Byte 1	I_{63}	I_{62}	I_{61}	I_{60}	I_{59}	I_{58}	I_{57}	I_{56}
Byte 2	I_{55}	I_{54}	I_{53}	I_{52}	I_{51}	I_{50}	I_{49}	I_{48}
Byte 3	I_{47}	I_{46}	I_{45}	I_{44}	I_{43}	I_{42}	I_{41}	I_{40}
Byte 4	I_{39}	I_{38}	I_{37}	I_{36}	I_{35}	I_{34}	I_{33}	I_{32}
Byte 5	I_{31}	I_{30}	I_{29}	I_{28}	I_{27}	I_{26}	I_{25}	I_{24}
Byte 6	I_{23}	I_{22}	I_{21}	I_{20}	I_{19}	I_{18}	I_{17}	I_{16}
Byte 7	I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8
Byte 8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Value = $I_{63}*2^{63} + I_{62}*2^{62} + \ldots + I_1*2^1 + I_0*2^0$

Code 15 — 3-byte, unsigned integer

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

Value = $I_{23}*2^{23} + I_{22}*2^{22} + \ldots + I_1*2^1 + I_0*2^0$

Code 16 — 1-byte, unsigned integer

Bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Byte 1	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

Value = $I_7*2^7 + I_6*2^6 + \ldots + I_1*2^1 + I_0*2^0$

Appendix F. EBCDIC and ASCII Codes

Table 19 IBM 3270 Char Set Ref Ch 10, GA27-2837-9, April 1987

Char	EBCDIC (hex)	ASCII (hex)	Description
NU	x00	x00	Null (nul)
SH	x01	x01	Start of heading (soh)
SX	x02	x02	Start of text (stx)
EX	x03	x03	End of text (etx)
ST	x04	x9C	String terminator (st)
HT	x05	x09	Character tabulation (ht)
SA	x06	x86	Start of selected area (ssa)
DT	x07	7F	Delete (del)
EG	x08	x97	End of guarded area (epa)
RI	x09	x8D	Reverse line feed (ri)
S2	x0A	x8E	Single-shift two (ss2)
VT	x0B	x0B	Line tabulation (vt)
FF	x0C	x0C	Form feed (ff)
CR	x0D	x0D	Carriage return (cr)
SO	x0E	x0E	Shift out (so)
SI	x0F	x0F	Shift in (si)
DL	x10	x10	Datalink escape (dle)
D1	x11	x11	Device control one (dc1)
D2	x12	x12	Device control two (dc2)
D3	x13	x13	Device control three (dc3)
OC	x14	x9D	Operating system command (osc)
NL	x15	x85	Next line (nel)
BS	x16	x08	Backspace (bs)
ES	x17	x87	End of selected area (esa)
CN	x18	x18	Cancel (can)
EM	x19	x19	End of medium (em)
P2	x1A	x92	Private use two (pu2)

Char	EBCDIC (hex)	ASCII (hex)	Description
S3	x1B	x8F	Single-shift three (ss3)
FS	x1C	x1C	File separator (is4)
GS	x1D	x1D	Group separator (is3)
RS	x1E	x1E	Record separator (is2)
US	x1F	x1F	Unit separator (is1)
PA	x20	x80	Padding character (pad)
HO	x21	x81	High octet preset (hop)
BH	x22	x82	Break permitted here (bph)
NH	x23	x83	No break here (nbh)
IN	x24	x84	Index (ind)
LF	x25	x0A	Line feed (lf)
EB	x26	x17	End of transmission block (etb)
EC	x27	x1B	Escape (esc)
HS	x28	x88	Character tabulation set (hts)
HJ	x29	x89	Character tabulation with justification (htj)
VS	x2A	x8A	Line tabulation set (vts)
PD	x2B	x8B	Partial line forward (pld)
PU	x2C	x8C	Partial line backward (plu)
EQ	x2D	x05	Enquiry (enq)
AK	x2E	x06	Acknowledge (ack)
BL	x2F	x07	Bell (bel)
DC	x30	x90	Device control string (dcs)
P1	x31	x91	Private use one (pu1)
SY	x32	x16	Synchronous idle (syn)
TS	x33	x93	Set transmit state (sts)
CC	x34	x94	Cancel character (cch)
MW	x35	x95	Message waiting (mw)
SG	x36	x96	Start of guarded area (spa)
ET	x37	x04	End of transmission (eot)
SS	x38	x98	Start of string (sos)
GC	x39	x99	Single graphic character

Char	EBCDIC (hex)	ASCII (hex)	Description
			introducer (sgci)
SC	x3A	x9A	Single character introducer (sci)
CI	x3B	x9B	Control sequence introducer (csi)
D4	x3C	x14	Device control four (dc4)
NK	x3D	x15	Negative acknowledge (nak)
PM	x3E	x9E	Privacy message (pm)
SB	x3F	x1A	Substitute (sub)
SP	x40	x20	Space, Blank
¢	x4A	xA2	Cent sign
.	x4B	x2E	Full stop, Period
<	x4C	x3C	Less-than sign
(	x4D	x28	Left parenthesis
+	x4E	x2B	Plus sign
|	x4F	x7C	Vertical line, Logical OR
&	x50	x26	Ampersand
!	x5A	x21	Exclamation mark
$	x5B	x24	Dollar sign
*	x5C	x2A	Asterisk
)	x5D	x29	Right parenthesis
;	x5E	x3B	Semicolon
¬	x5F	xAC	Not sign
-	x60	x2D	Hyphen, Minus
/	x61	x2F	Solidus, Forward slash
BB	x6A	xA6	Broken bar
,	x6B	x2C	Comma
%	x6C	x25	Percent sign
_	x6D	x5F	Low line, Underline, Underscore
>	x6E	x3E	Greater-than sign
?	x6F	x3F	Question mark
`	x79	x60	Grave accent
:	x7A	x3A	Colon
#	x7B	x23	Number sign, Pound sign, hash mark
@	x7C	x40	Commercial at
'	x7D	x27	Apostrophe

Char	EBCDIC (hex)	ASCII (hex)	Description
=	x7E	x3D	Equals sign
"	x7F	x22	Quotation mark
a	x81	x61	Latin small letter A
b	x82	x62	Latin small letter B
c	x83	x63	Latin small letter C
d	x84	x64	Latin small letter D
e	x85	x65	Latin small letter E
f	x86	x66	Latin small letter F
g	x87	x67	Latin small letter G
h	x88	x68	Latin small letter H
i	x89	x69	Latin small letter I
j	x91	x6A	Latin small letter J
k	x92	x6B	Latin small letter K
l	x93	x6C	Latin small letter L
m	x94	x6D	Latin small letter M
n	x95	x6E	Latin small letter N
o	x96	x6F	Latin small letter O
p	x97	x70	Latin small letter P
q	x98	x71	Latin small letter Q
r	x99	x72	Latin small letter R
~	xA1	x7E	Tilde
s	xA2	x73	Latin small letter S
t	xA3	x74	Latin small letter T
u	xA4	x75	Latin small letter U
v	xA5	x76	Latin small letter V
w	xA6	x77	Latin small letter W
x	xA7	x78	Latin small letter X
y	xA8	x79	Latin small letter Y
z	xA9	x7A	Latin small letter Z
{	xC0	x7B	Left curly bracket
A	xC1	x41	Latin capital letter A
B	xC2	x42	Latin capital letter B
C	xC3	x43	Latin capital letter C
D	xC4	x44	Latin capital letter D
E	xC5	x45	Latin capital letter E
F	xC6	x46	Latin capital letter F
G	xC7	x47	Latin capital letter G

Char	EBCDIC (hex)	ASCII (hex)	Description
H	xC8	x48	Latin capital letter H
I	xC9	x49	Latin capital letter I
}	xD0	x7D	Right curly bracket
J	xD1	x4A	Latin capital letter J
K	xD2	x4B	Latin capital letter K
L	xD3	x4C	Latin capital letter L
M	xD4	x4D	Latin capital letter M
N	xD5	x4E	Latin capital letter N
O	xD6	x4F	Latin capital letter O
P	xD7	x50	Latin capital letter P
Q	xD8	x51	Latin capital letter Q
R	xD9	x52	Latin capital letter R
\	xE0	x5C	Reverse solidus, Back slash
S	xE2	x53	Latin capital letter S
T	xE3	x54	Latin capital letter T
U	xE4	x55	Latin capital letter U
V	xE5	x56	Latin capital letter V
W	xE6	x57	Latin capital letter W
X	xE7	x58	Latin capital letter X
Y	xE8	x59	Latin capital letter Y
Z	xE9	x5A	Latin capital letter Z
0	xF0	x30	Digit zero
1	xF1	x31	Digit one
2	xF2	x32	Digit two
3	xF3	x33	Digit three
4	xF4	x34	Digit four
5	xF5	x35	Digit five
6	xF6	x36	Digit six
7	xF7	x37	Digit seven
8	xF8	x38	Digit eight
9	xF9	x39	Digit nine
AC	xFF	x9F	Application program command (apc)
NUL	x00	x00	Nul
SOH	x01	x01	Start of heading (soh)
STX	x02	x02	Start of text (stx)

Char	EBCDIC (hex)	ASCII (hex)	Description
ETX	x03	x03	End of text (etx)
EOT	x37	x04	End of transmission (eot)
ENQ	x2D	x05	Enquiry (enq)
ACK	x2E	x06	Acknowledge (ack)
alert	x2F	x07	Bell (bel)
BEL	x2F	x07	Bell (bel)
backspace	x16	x08	Backspace (bs)
tab	x05	x09	Character tabulation (ht)
newline	x25	x0A	Line feed (lf)
vertical-tab	x0B	x0B	Line tabulation (vt)
form-feed	x0C	x0C	Form feed (ff)
carriage-return	x0D	x0D	Carriage return (cr)
DLE	x10	x10	Datalink escape (dle)
DC1	x11	x11	Device control one (dc1)
DC2	x12	x12	Device control two (dc2)
DC3	x13	x13	Device control three (dc3)
DC4	x3C	x14	Device control four (dc4)
NAK	x3D	x15	Negative acknowledge (nak)
SYN	x32	x16	Synchronous idle (syn)
ETB	x26	x17	End of transmission block (etb)
CAN	x18	x18	Cancel (can)
SUB	x3F	x1A	Substitute (sub)
ESC	x27	x1B	Escape (esc)
IS4	x1C	x1C	File separator (is4)
IS3	x1D	x1D	Group separator (is3)
intro	x1D	x1D	Group separator (is3)
IS2	x1E	x1E	Record separator (is2)
IS1	x1F	x1F	Unit separator (is1)
DEL	x07	x7F	Delete (del)
space	x40	x20	Space
!	x5A	x21	Exclamation mark
"	x7F	x22	Quotation mark
#	x7B	x23	Number sign
$	x5B	x24	Dollar sign

Char	EBCDIC (hex)	ASCII (hex)	Description
%	x6C	x25	Percent sign
&	x50	x26	Ampersand
'	x7D	x27	Apostrophe
(	x4D	x28	Left parenthesis
)	x5D	x29	Right parenthesis
*	x5C	x2A	Asterisk
+	x4E	x2B	Plus sign
,	x6B	x2C	Comma
-	x60	x2D	Hyphen, Minus
.	x4B	x2E	Full stop, Period
/	x61	x2F	Solidus, Slash
0	xF0	x30	Digit Zero
1	xF1	x31	Digit one
2	xF2	x32	Digit two
3	xF3	x33	Digit three
4	xF4	x34	Digit four
5	xF5	x35	Digit five
6	xF6	x36	Digit six
7	xF7	x37	Digit seven
8	xF8	x38	Digit eight
9	xF9	x39	Digit nine
:	x7A	x3A	Colon
;	x5E	x3B	Semicolon
<	x4C	x3C	Less-than sign
=	x7E	x3D	Equals sign
>	x6E	x3E	Greater-than sign
?	x6F	x3F	Question mark
@	x7C	x40	Commercial at
[	X00	x5B	Left square bracket
\	xE0	x5C	Reverse solidus, Backslash
]	x00	x5D	Right square bracket
^	x00	x5E	Circumflex, Caret
_	x6D	x5F	Low line, Underscore
`	x79	x60	Grave accent
{	xC0	x07B	Left curly bracket
|	x4F	x7C	Vertical line
}	xD0	x7D	Right curly bracket
~	xA1	x7E	Tilde

Appendix G. References

Barry, K. M., Cavers, D. A. and Kneale, C. W., 1975, Report on recommended standards for digital tape formats: Geophysics, 40**,** no. 02, 344-352[28].

Booth, Algan, Duke, Guyton, Norris, Stainsby, Theriot, Wildgoose and Wilhelmsen, 1997, SEG Rode Format Record Oriented Data Encapsulation: Geophysics, 61, no. 05, 1545-1558.

Geomatics Committee, 2015, OGP P1/11 Geophysical position data exchange format, International Association of Oil & Gas Producers, Report 483-1, www.iogp.org.

Geomatics Committee, 2012, OGP P6/11 Seismic bin grid data exchange format, International Association of Oil & Gas Producers, Report 483-6, www.iogp.org.

Norris, M. W. and Faichney, A.K. (Eds.), 2002, SEG-Y rev1 Data Exchange format: SEG, Tulsa, OK.

Northwood, E. J., Wisinger, R. C., and Bradley, J. J., 1967, Recommended standards for digital tape formats: Geophysics, v32, p. 1073-1084.

SEG Comm. Field Tape Std., 1994, Digital field tape standards - SEG-D, revision 1 (special report): Geophysics, 59, no. 04, 668-684.

SEG Technical Standards Committee, 2011, SEG-D Rev. 3.0 update, SEG, Tulsa, OK, www.seg.org.

Wood, Behn, Borresen, Guyton, Miles, O'Neill, Oishi and Scales, 1998, SEG-D,

[28] SEG-Y rev 0 format

Rev 2 Field Tape Standards, Geophysics,
62, no. 03, 1004-1031 .