UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

MAGNETIC TAPE FORMAT FOR THE USGS OCEAN BOTTOM SEISMOMETER

by

G. K. Miller and J.E. Dodd[1]

Open-File Report 86 - 256

1. Woods Hole, Mass.

1986

MAGNETIC TAPE FORMAT
FOR THE USGS OCEAN BOTTOM SEISMOMETER

This report replaces the OBS tape format listed in Fredicks, J.J., Evenden, G.I., and Dodd, J.E., 1985, Shipboard playback for Ocean Bottom Instrument Package, Modification II: U.S. Geological Survey Open-File Report 85-111, 69 p.

The magnetic tape format used in the US Geological Survey ocean bottom seismometer (OBS) conforms to the Tape Interchange Package (TIP) format described in the Operator's Guide #OG - 100055 rev 0981 published by the Alloy Engineering Company, Inc., 12 Mercer Road, Natick, MA 01760. The OBS uses the Alloy IDXS-100 controller board as an interface to the tape recorder. Using Alloy's software simplified the process of reading and writing to tape.

A TIP record length of 8208 bytes was chosen as the most suitable transfer length for OBS data records. The TIP format also requires a specific 16-byte record header (file control block) at the beginning of each record. The mandated format is as follows:

Byte #	Description
0	Contains the user number 00 (hexadecimal).
1 - 8	File name in ASCII upper case with high bit = 0.
9 - 11	File type in ASCII upper case with high bit = 0.
12	Save set number 00 - 0F (hexadecimal).
13	Last block flag. This is true for the last block of a file only.
14	Source drive code 00 - 0F (hexadecimal) (Corresponds to drives A - P).
15	Number of 128-byte records in this tape block that contain data.

The USGS implementation of this format includes the following data:

Byte #	Description
0	Always contains the number 00H (hexadecimal).
1 - 10	Contains the 4-digit Series (S) and Experiment (E) numbers written as: SxxxxExxxx The first two records are exceptions to this. Record #1 contains all blanks (20 hexadecimal); Record #2 contains the label "GPHEADER" with the rest of the space filled with blanks (20H).
11	Always contains the number 20H (hexadecimal).
12	Always contains the number 00H (hexadecimal).
13	Last block (LB) flag. This is true (01 hexadecimal) for the last block of a file only.
14	Always contains the number 00H (hexadecimal).
15	Number of 128-byte records in this tape block that contain data. The maximum value = 40 (hexadecimal).

The OBS format requirements include:

1) All record lengths must be 8208 bytes.

2) An end-of-file mark (a record with 55H hexadecimal written instead of data) is written at the beginning of tracks 2, 3, and 4 of the tape. The first track is started with the beginning of tape (BOT) mark (a hole in the data tape used by the drive to determine the physical beginning of the tape).

3) Two end-of-file marks are written at the end of all data on the tape.

4) The first record on tape must be a test record. This record is generated by the OBS to test the tape drive and to ensure its operation before deployment. The test pattern written to tape is a sequence of hexadecimal values from 00H to FFH. This pattern is repeated until all 8208 bytes are filled.

5) The second record on tape must be the general purpose header. This record contains all of the experimental parameters entered into the OBS by the operator before deployment.

6) The last 256 bytes of the final record of each file contain

the parameters for all series and the time of the data event.

In standard TIP format, a file represents a series of data records, each of which begins with the 16-byte header followed by an end-of-file mark. In the OBS format, a file represents a data-gathering event which is called either a ´window´ or an ´event´. The number of records that make up this file is determined by the operator when entering parameters into the OBS. The OBS has a 32 kilobyte data buffer, so the maximum number of 8192-byte records (8208 bytes less the 16 bytes of TIP header) that can be written is four. The operating system of the OBS allows the operator to choose whether to use all of the buffer or only part of it. The current options are to write 1,2, or 4 records per file. These are called ´blocks´ in the OBS program. Each of these blocks contains the 16-byte header with the series and experiment numbers. The last record contains the 256-byte trailer that comprises the series information and the time of the data event. There are no end-of-file marks written at the end of each file. This was done to maximize the amount of data that could be written to tape.

EXAMPLE: It is decided to write four records (or ´blocks´) for each file. The number of records will be recorded in the general purpose header block in the ´buffer-size´ field. Then, four physical records will be recorded into the data buffer. Note that the last block flag in the 16-byte header must be set in the fourth record header. Finally the file will be written to tape in following way:

```
       Record 1         Record 2           Record 3          Record 4
  _______________  _________________  _________________  _________________
 | :             || :               || :               || :          :    |
 |_:_____________||_:_______________||_:_______________||_:__________:____|
 ^                ^                  ^                  ^                 ^
 16-byte          16-byte            16-byte            16-byte           |
 header           header             header             header            |
                                                        LB set            |
                                                                  256-byte
                                                                  trailer
```

FORMAT OF THE GENERAL PURPOSE HEADER

The general purpose header is one 8208-byte record that is the second record (following the test record) written to tape. This second record is divided into two sections. The first section contains the cruise information, the second section contains the 256-byte trailer that lists the parameters for each series of experiments and the time of the actual data event.

The cruise information contains miscellaneous information entered by the operator before deployment. This information has no impact on the operation of the OBS, but the information is useful in identifying the data tape. This section can fill from byte 16 (the 16-byte header starts at byte 0) up to byte 7950 of the record. Fifteen different lines of information must be entered in this section, and each line can contain up to 80 bytes. The lines are terminated with a carriage return and line feed (CRLF). Labels for each line of header information are automatically written into the header by the OBS. If no entry is made, a carriage return and line feed will follow the label. All data is stored as ASCII. The entry lines are as follows:

DEPLOYMENT # (xxxxxxxxxxxxxxxxCRLF)
INSTRUMENT # (xxxxxxxxxxxxxxxxCRLF)
CHIEF SCIENTIST (xxxxxxxxxxxxxCRLF)
CRUISE # (xxxxxxxxxxxxxxxxxxxxCRLF)
SPHERE # (xxxxxxxxxxxxxxxxxxxxCRLF)
LATITUDE (xxxxxxxxxxxxxxxxxxxxCRLF)
LONGITUDE (xxxxxxxxxxxxxxxxxxxCRLF)
FRONT END GAIN CRLF
CHANNEL 1 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)
CHANNEL 2 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)
CHANNEL 3 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)
CHANNEL 4 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)
FRONT END DAMPING CRLF
CHANNEL 1 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)
CHANNEL 2 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)
CHANNEL 3 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)
CHANNEL 4 (xxxxxxxxxxxxxxxxxxxxxxxCRLF)

(xxxCRLF) = operator's entry up to 80 bytes

Since the end point of this section depends on the length of each entry, a hexadecimal value of 00 is written at the end of the section. The remaining space of this section is filled with zeros.

The second section is the 256-byte trailer, which is also included at the end of each data file, and which contains the experimental parameters for each series of the deployment. These parameters affect the operation of the OBS. This information starts at byte 7952 and ends at byte 8168. This is immediately followed by the time of the current data-acquisition sample and its series and experiment number. In the general purpose header, the clock is not read, so no time data are recorded. The following is a breakdown of the 256-byte trailer:

BYTE #	DATA TYPE	DESCRIPTION
Series 1 Information		
7952	hexadecimal	Base address of the pertinent A-D port (used to determine the base channel) 18H = Channel 1 1AH = Channel 2 1CH = Channel 3 1EH = Channel 4
7953	hexadecimal	Numbers of data channels times 2
7954	ASCII	Series type (74H for timer and 65H for event)
7955	packed BCD	LSB of total number of experiments
7956	packed BCD	MSB of total number of experiments
7957	packed BCD	Year of start time
7958	packed BCD	Month of start time
7959	packed BCD	Day of start time
7960	packed BCD	Hour of start time
7961	packed BCD	Minute of start time
7962	packed BCD	Year of stop time
7963	packed BCD	Month of stop time
7964	packed BCD	Day of stop time
7965	packed BCD	Hour of stop time
7966	packed BCD	Minute of stop time
7967	packed BCD	Number of data blocks (records) per file.
7968	hexadecimal	MSB of number of post event samples - for event experiments only
7969	hexadecimal	LSB of number of post event samples - for event experiments only
7970	hexadecimal	High order address for the start of the data buffer in memory
7971	hexadecimal	MSB of maximum number of samples (bytes) - based on buffer size and sample rate
7972	hexadecimal	LSB of maximum number of samples (bytes) - based on buffer size and sample rate
7973	packed BCD	Window offset (in seconds) - for window (Timer mode) only
7974	packed BCD	Window period (in minutes) - for window (Timer mode) only
7975	hexadecimal	Sample rate code - value to load into the A-D board 02H = 1 ms 06H = 2 ms 01H = 4 ms 05H = 8 ms
7976	hexadecimal	Short term average and threshold code - value to load into the analog board (event mode only), upper 4 bits = STA, lower 4 bits = Threshold 11H = STA of .05s & THRES of 6dB 22H = STA of .10s & THRES of 12dB 44H = STA of .25s & THRES of 18dB 88H = STA of .50s & THRES of 24dB

BYTE #	DATA TYPE	DESCRIPTION
Series 2 Information		
7977 - 8001		as above
Series 3 Information		
8002 - 8026		as above
Series 4 Information		
8027 - 8051		as above
Series 5 Information		
8052 - 8076		as above
Series 6 Information		
8077 - 8101		as above
Series 7 Information		
8102 - 8127		as above
Series 8 Information		
8128 - 8152		as above
Data Event Information		
8170	hexadecimal	Pointer to next series parameters (added to location 7952H hexidecimal)
8171	packed BCD	LSB of current series number
8172	packed BCD	MSB of current series number
8173	packed BCD	LSB of current experiment number
8174	packed BCD	MSB of current experiment number
8175	packed BCD	Tenths of seconds - current time (not rounded)
8176	packed BCD	Unit seconds - current time
8177	packed BCD	Tens of seconds - current time
8178	packed BCD	Unit minutes - current time
8179	packed BCD	Tens of minutes - current time
8180	packed BCD	Unit hours - current time
8181	packed BCD	Tens of hours - current time
8182	packed BCD	Unit days - current time
8183	packed BCD	Tens of days- current time
8184	packed BCD	Day of week - current time (always 00H)
8185	packed BCD	Unit months - current time
8186	packed BCD	Tens of months - current time
8187	packed BCD	Years - current time
8188	packed BCD	Thousands of seconds - current time
8189	packed BCD	Tenths and hundredths of seconds - current time
8190	hexadecimal	Number of 128-byte records written

EXAMPLE: Suppose that a data event was recorded at December 25, 1986 at 12:35 and 47.289 seconds in event mode, series 2, experiment number 1764. The data event section of the trailer would result as follows:

BYTE #	HEXADECIMAL VALUES
8170	32 02 00 64 17 02 07 04 05 03 02 01 05 02 00 02
8186	01 86 90 28 3E 00 00 00 00 00 00 00 00 00 00 00

DATA FILE FORMAT

The basic layout of a data file was described on page 3. The format of the 16-byte header was described on page 2. The data comprise 2 bytes per data word. The data are given as a simple serial stream that enters channel 1 LSB, channel 1 MSB, channel 2 LSB, channel 2 MSB, channel 3 LSB, channel 3 MSB, channel 4 LSB, and channel 4 MSB over and over again until the data buffer is filled. The number of channels and the base channel used (both located in the 256-byte trailer) must be known before the locations of each data word can be found in each record. The base channel is always written first.

Example: Suppose series 1, experiment 1, records data on base channel 2 and that the total number of channels is three (i.e., record channels 2,3,4). The data record will be written as follows:

```
                                                        01 in last
BYTE #     HEXADECIMAL VALUES                           record
                                                        |
 0000      00 53 30 30 30 31 45 30 30 30 31 20 00 00 00 40
 0016      87 9D 45 C3 02 9A 65 9D 67 C3 34 99 90 9D 12 C3
           ^     ^     ^     ^
           LSB   LSB   LSB   LSB ... etc.
           CH2   CH3   CH4   CH2
```

The data word is made up of two 8-bit bytes. The first byte is the least significant 8 bits of the analog-to-digital (A-D) conversion. The second byte is divided into two 4-bit values. The upper four bits equal the gain code from the gain-ranging circuit, and the lower four bits record the most significant bits of the A-D conversion. A voltage value can be derived out of the data word by first considering the 12-bit value from the A-D. The maximum number of binary steps possible with a 12-bit A-D is 4096, which is determined by adding 1 to the maximun data word (FFFH, hexidecimal or 4095, decimal). This is referenced to a 10 volt span (+/- 5 volts) in the OBS. The step value for the A-D output can be calculated by dividing the voltage span by the maximum number of steps. In this example,

$$10 \text{ volts}/ 4096 \text{ steps} = 0.0024414 \text{ volts/step}$$

The voltage value is obtained by multiplying this value times the 12-bit data word (decimal value). This result must be corrected for the gain-ranging to arrive at the voltage level produced by the preamplifier. The 4-bit gain code is a compression of a 16-bit gain word. To convert the 4-bit gain code to the 16-bit gain word use the following equation:

$$\text{gain word} = 2^{\text{gain code}} + 1$$

This result is then divided into the voltage output of the A-D. To obtain the voltage output of the sensor, the last result must be divided by the value entered in the general purpose header as the preamp gain.

EXAMPLE: Suppose a value of 9D87 (hex) was recorded in an OBS that had a preamp gain of 466. First, calculate the voltage output of the A-D. The 12-bit word equals D87H (hexadecimal) or 3463 (decimal). The voltage value is calculated by

$$3463 \text{ bits} * .0024414 \text{ volts/bit} = 8.45457 \text{ volts}$$

To correct for the gain-range, calculate the gain word by

$$\text{gain word} = 2^{9(\text{hexadecimal})} + 1 \text{ or } = 2^{9(\text{decimal})} + 1$$

$$\text{gain word} = 512 + 1$$

the voltage is then corrected for the gain-ranging by

$$\text{voltage} = 8.45457 \text{ volts} / 513 = 0.0164806 \text{ volts}$$

to obtain the final corrected voltage divide by the preamp gain (466)

$$\text{sensor voltage} = 0.0164806 \text{ volts} / 466 = 0.0000353 \text{ volts}$$

or

$$\text{sensor voltage} = 35.3 \text{ microvolts}$$

The 256-byte trailer will appear in the last record of the file (the record with the last-block flag set in the 16-byte header). This trailer is identical with the description starting on page 4. This trailer will affect the amount of data recorded for each data event, since the length of the data is calculated on full 8 kilobyte blocks. To determine the exact length of the data for each channel, 256 bytes worth of data samples must be subtracted from the total data buffer.

EXAMPLE: Suppose the OBS recorded 4 channels of data using 4 records (blocks) of data per file. The sample rate used in recording is 8 milliseconds. The total data buffer in the OBS used in this instance is 32768 bytes. Since 4 channels of data are recorded, the data buffer for each channel is 8192 bytes. Each data word is made up two bytes, so there are 4096 samples of data per channel recorded per file. At a sampling rate of 8 milliseconds, 125 samples are recorded per second. Divide this into the total number of samples per channel (4096), and the total time of each file is 32.768 seconds. The loss of the last 256 bytes corresponds to a loss of 32 samples per channel or 0.256 seconds of data. This means that the data buffer is actually 32.512 seconds long.