XTF Survey File Format

Revision 7

July 2004

Table of Contents

XTF SURVEY FILE FORMAT

The External Survey File (".XTF") is a structured binary file which stores well data in a hierarchical organization accessible via direct access methods. The XTF data file is created and accessed using the *libsurv* subroutine interface. Refer to the *libsurv* documentation for further information on using that interface.

The XTF file structure comprises two main sections:

- *File Header*
- *Data Types*

The File Header section accommodates relational information about the file, curves and specific data types used by the XTF access software to achieve fast data access. This section occupies the first eight physical records of the file.

The Data Types section holds data for curves, zones and other defined data types. Curve data include: curve header records containing curve attribute information, extended curve headers and sample data for various supported curve types. This Data Types section occupies the ninth and subsequent physical records.

The XTF format supports the following types of curve data storage:

- Curve Type 1: Conventional
- Curve Type 2: Waveform (1-Dimensional Array)
- Curve Type 3: Matrix
- Curve Type 4: Structured
- Curve Type 5: Composite

Note: The **Composite Curve** (Curve Type 5) has been designed as an improved replacement for the Structured Curve (DataType 4). New applications should use the Composite Curve rather that the Structured Curve. Support for the Structured Curve datatype is retained only for backwards compatibility of applications.

Other data types supported include:

- Symbolic Well Variables
- Named Zones and associated Zone Variables
- Service Headers (containing Log Run attributes)
- Zonelists (lists of zone depth pairs for intervals which match some specified criteria)
- Embedded ASCII files

In addition, applications may define and store application specific datatypes using the same general mechanism.

Refer to the *Data Types* section for descriptions of these different data storage types.

Figure 1 illustrates the general layout of the principal data sections within the XTF file.

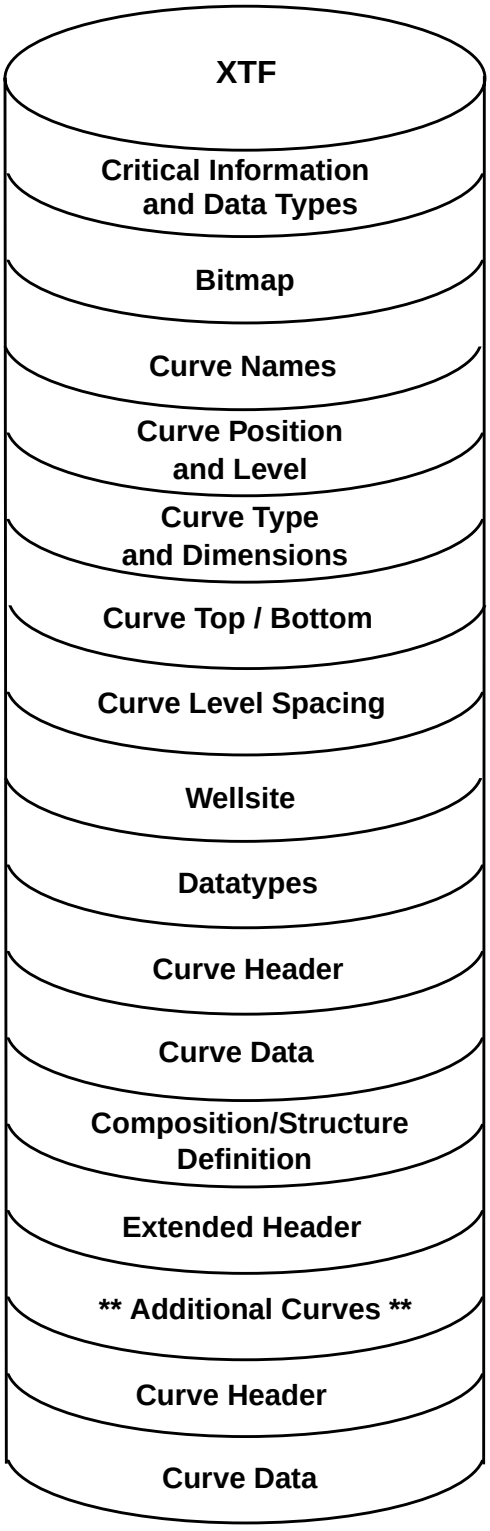

Figure 1

XTF Format Layout

FILE HEADER

The XTF File Header occupies the first eight physical records of the data file. It contains information used by the data access software to achieve rapid data access.

- *File Header Record 1: General Header*
- *File Header Record 2: Bitmap*
- *File Header Record 3: Curve Names*
- *File Header Record 4: Curve Data Addresses*
- *File Header Record 5: Curve Data Dimensions*
- *File Header Record 6: Curve Top and Bottom Indices*
- *File Header Record 7: Curve Level Spacings and Types*
- *File Header Record 8: Wellsite Attributes*

Table 1 summarizes the usage of the File Header records.

Table 1: XTF Record Structure

Record #	Type	Description
1	General Header	Critical header data such as file name, system code, revision level, and datatype names.
2	Bitmap	Contains information about disk allocation. Additional bitmap blocks can be intermixed with the curve data.
3	Curve Names	Contains up to 512 eight character curve names.
4	Curve Addresses	Relative start address and number of levels in each curve.
5	Curve Dimensions	Dimensionality of curve data for each curve.
6	Curve Top / Bottom	Curve top and bottom depth or time values.
7	Curve Level Spacings	Curve level spacing and type of data.
8	Wellsite Attributes	Wellsite attributes such as location and reference datum.
9	Data Record	Data for any of the supported datatypes.
.	.	.
.	.	.
EOF	Data Record	Data (limited only by the size of the disk drive).

File Header Record 1: General Header

The first record of a survey file is subdivided into four quadrants of predefined information. The first quadrant of record one contains pertinent information about the well data file in its entirety.

Table 2: General Well Data File Information Record (I4BUFH) - First Quadrant

4 Byte Elements	Bytes	Type	Parameter	Description
001-008	0001-0032	Int*2	INITS	Location of file header records, particularly the current bitmap in bytes 4-8
009-190	0033-0760	Char	Unused	N / A
191	0761-0764	Char*4	CHTYPE	"*.xtf*" or other extension type
192-211	0765-0844	Char*80	CHNAME	File name, including extension
212-213	0845-0852	Char*8	CHUNIT	Depth units mnemonics for this file (lowercase)
214-215	0853-0860	Char*8	CH8VER	AIF acquisition revision number
216-237	0861-0948	Int*2	Unused	N / A
238	0949	Byte	NUMSYS	System code: 1 - PC 2 - Perkin Elmer 3 - Vax 4 - IBM mainframe 5 - Unix -- HP, Sun, SGI
	0950	Byte	QC	Used to indicate target system has a format conversion in progress.
	0951	Byte	Major Rev	SURVLIB major revision number
	0952	Byte	Minor Rev	SURVLIB minor revision number
239-249	0953-0996	Int*2	Unused	N / A
250	0997-1000	Int*2	ISNUMCV	Number of curves in the survey file
251	1001-1004	Int*2	ISMAXCV	Maximum number of curves that can be held in the survey file (currently 512)
252-1024	1005-1024	Int*4	Unused	N / A

The second quadrant of record one contains information about additional datatypes other than curve data. These datatypes include well site information, zone lists and named zones. A maximum of 64 datatypes can be defined.

Table 3: General Well Data File Information Record (I4BUFH) - Second Quadrant

4 Byte Elements	Bytes	Type	Parameter	Description
257-258	1025-1032	Char*8	CHARDT(1)	Name of first datatype (XWD0001)
259	1033-1036	Int*4	I4REC1(1)	From record range
260	1037-1040	Int*4	I4REC2(1)	To record range
261-262	1041-1048	Char*8	CHARDT(2)	Name of second datatype (XWD0002)
.	.	.	.	.
512	2045-2048	Int*4	I4REC2(*n*)	To record range

The third quadrant of record one contains floating point values for the well.

Table 4: General Well Data File Information Record (I4BUFH) - Third Quadrant

4 Byte Elements	Bytes	Type	Parameter	Description
513	2049-2052	Real*4	SURVTOP	Default top of survey
514	2053-2056	Real*4	SURVBOT	Default bottom of survey
515	2057-2060	Real*4	SURVRLEV	Default level spacing
516-768	2061-3072	Real*4	Unused	N / A

The fourth quadrant of record one is mostly unused.

Table 5: General Well Data File Information Record (I4BUFH) - Fourth Quadrant

4 Byte Elements	Bytes	Type	Parameter	Description
0769-1023	3073-4092	Unused	Unused	Unused
1024	4093-4096	Int*4	I4BYTES	Size of this file in 4096 byte blocks

File Header Record 2: Bitmap

The second header record contains information on disk storage. It is used to locate unused or deleted areas within the file, and is link-listed to other records when it becomes full, thereby allowing only the physical disk size as a limitation to file size.

Table 6: Bitmap Record (I4BUFB)

4 Byte Elements	Bytes	Type	Parameter	Description
1(lh)	0001-0002	Bits	N / A	First bitmap reflects eight used header records
1(rh) - 1018	0003-4072	Bits	N / A	Marked with a “1” (bit “on”) when a data block (4096-bytes) is allocated
1019	4073-4076	Int*4	I4FULL	Bitmap full (1) flag
1020	4077-4080	Int*4	I4FRSTRC	First record in this bitmap
1021	4081-4084	Int*4	I4LSTREC	Last record that can be in this bit map
1022	4085-4088	Int*4	I4PREVBM	Record number of previous bitmap
1023	4089-4092	Int*4	I4THISBM	Record number of this bitmap
1024	4093-4096	Int*4	I4NEXTBM	Record number of next bitmap

File Header Record 3: Curve Names

The third header record contains eight-character curve names for the data curves. As they are added and deleted, unused areas are automatically reused. A maximum of 512 curves can be defined.

Table 7: Curve Names Record (I4BUFK)

4 Byte Elements	Bytes	Type	Parameter	Description
1	0001-0008	Char*8	CHCURV(1)	First curve name
2	0009-0016	Char*8	CHCURV(2)	Second curve name
.	.	.	.	.
.	.	.	.	.
512	4089-4096	Char*8	CHCURV(512)	Last (512) curve name

File Header Record 4: Curve Data Addresses

The fourth file header record contains positional attributes which relate directly to the curves defined in File Header Record 3. For every curve name there is an associated relative address for the start of its contents and a count of its total number of levels. The addresses are in units of 4096 byte blocks and are relative to the start of the file.

Table 8: Curve Data Addresses Record (I4BUFP)

4 Byte Elements	Bytes	Type	Parameter	Description
1	0001-0004	Int*4	I4FIRST(1)	Starting offset for 1st data curve Example: 0000 0009 (Hex representation)
2	0005-0008	Int*4	I4FIRST(257)	Starting offset for 257th curve
3	0009-0012	Int*4	I4FIRST(2)	Starting offset for 2nd curve
4	0013-0016	Int*4	I4FIRST(258)	Starting offset for 258th curve
.	.	.	.	.
.	.	.	.	.
512	2045-2048	Int*4	I4FIRST(512)	Starting offset for 512th curve
513	2049-2052	Int*4	NLEVS(1)	Number of levels for 1st curve
514	2053-2056	Int*4	NLEVS(257)	Number of levels for 257th curve
515	2057-2060	Int*4	NLEVS(2)	Number of levels for 2nd curve
516	2061-2064	Int*4	NLEVS(258)	Number of levels for 258th curve
.	.	.	.	.
.	.	.	.	.
1024	4092-4096	Int*4	NLEVS(512)	Number of levels for 512th curve

File Header Record 5: Curve Data Dimensions

The fifth record contains the dimensions for the curve data. The first element contains the number of dimensions (1-3) and the next three contain the number of elements in their respective dimensions.

Table 9: Curve Data Dimensions Record (I4BUFM)

2 Byte Elements	Bytes	Type	Parameter	Description
1	0001-0002	Int*2	NDIMS(1)	Number of dimensions (1-3) for curve 1
2	0003-0004	Int*2	IDIMS1(1)	Number of elements in 1st dimension for curve 1
3	0005-0006	Int*2	NDIMS(257)	Number of dimensions (1-3) for curve 257
4	0007-0008	Int*2	IDIMS1(257)	Number of elements in 1st dimension for curve 257
5	0009-0010	Int*2	NDIMS(2)	Number of dimensions (1-3) for curve 2
6	0011-0012	Int*2	IDIMS1(2)	Number of elements in 1st dimension for curve 2
.	.	.	.	.
1025	2049-2050	Int*2	IDIMS2(1)	Number of elements in 2nd dimension for curve 1
1026	2051-2052	Int*2	IDIMS3(1)	Number of elements in 3rd dimension for curve 1
1027	2053-2054	Int*2	IDIMS2(257)	Number of elements in 2nd dimension for curve 257
1028	2055-2056	Int*2	IDIMS3(257)	Number of elements in 3rd dimension for curve 257
1029	2057-2058	Int*2	IDIMS2(2)	Number of elements in 2nd dimension for curve 2
1030	2059-2060	Int*2	IDIMS3(2)	Number of elements in 3rd dimension for curve 2
.	.	.	.	.
.	.	.	.	.
2048	4094-4096	Int*2	IDIMS3(512)	Number of elements in 3rd dimension for curve 512

File Header Record 6: Curve Top and Bottom Indices

The sixth header record contains the top and bottom index values (feet, meters, or seconds) for each curve. They are represented as single precision floating point numbers.

Table 10: Curve Top and Bottom Indices Record (I4BUFT)

4 Byte Elements	Bytes	Type	Parameter	Description
1	0001-0004	Real*4	TOPDEPCV(1)	Top index for curve 1
2	0005-0008	Real*4	TOPDEPCV(257)	Top index for curve 257
3	0009-0012	Real*4	TOPDEPCV(2)	Top index for curve 2
4	0013-0016	Real*4	TOPDEPCV(258)	Top index for curve 258
.	.	.	.	.
.	.	.	.	.
512	2045-2048	Real*4	TOPDEPCV(512)	Top index for curve 512
513	2049-2052	Real*4	BOTDEPCV(1)	Bottom index for curve 1
514	2053-2056	Real*4	BOTDEPCV(257)	Bottom index for curve 257
515	2057-2060	Real*4	BOTDEPCV(2)	Bottom index for curve 2
516	2061-2064	Real*4	BOTDEPCV(258)	Bottom index for curve 258
.	.	.	.	.
.	.	.	.	.
1024	4093-4096	Real*4	BOTDEPCV(512)	Bottom index for curve 512

File Header Record 7: Curve Level Spacings and Types

The seventh header record contains the level spacing and data types for the curves. The level spacing is floating point. Attributes are provided to define the curve type, curve sample data type, horizontal organization type, and vertical organization type.

Table 11: Curve Level Spacings and types Record (I4BUFL)

4 Byte Elements	Bytes	Type	Parameter	Description	
1	0001-0004	Real*4	RLEVCV	Level spacing for curve 1	
2	0005-0008	Real*4	RLEVCV(257)	Level spacing for curve 257	
3	0009-0012	Real*4	RLEVCV(2)	Level spacing for curve 2	
4	0013-0016	Real*4	RLEVCV(258)	Level spacing for curve 258	
.	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	
512	2045-2048	Real*4	RLEVCV(512)	Level spacing for curve 512	
513	2049	Byte	ICTYPE(1)	Curve type (1-5)(See footnotes “a”,“e”,“f”) (curve 1)	
	2050	Byte	IDTYPE(1)	Curve sample data type (1-11) (See footnote “b”)	
	2051	Byte	IVTYPE(1)	Vertical type (1-2) (See footnotes “c” and “g”)	
	2052	Byte	IHTYPE(1)	Horizontal type (1-2) (See footnote “d”)	
514	2053	Byte	ICTYPE(257)	Curve type (1-5)	(curve 257)
	2054	Byte	IDTYPE(257)	Curve sample data type (1-11)	
	2055	Byte	IVTYPE(257)	Vertical type (1-2)	
	2056	Byte	IHTYPE(257)	Horizontal type(1-2)	
515	2057	Byte	ICTYPE(2)	Curve type (1-5)	(curve 2)
	.	.	.	.	
	.	.	.	.	
	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	
1024	4096	Byte	IHTYPE(512)	Horizontal type	(curve 512)

a. Curve Type (1-5)
1 - Conventional
2 - Waveform
3 - Matrix
4 - Structure
5 - Composite

b. Curve Sample Data Type (1-11)
1 - Bit
2 - Integer*2 (Signed short)
3 - Byte (Unsigned)
4 - Single precision floating point
5 - 12-bit signed
6 - 12-bit unsigned
7 - Character
8 - Double precision
9 - Unsigned integer*2 (ushort)
10 - Integer*4 (Signed long)
11 - Unsigned integer*4 (ulong)

c. Vertical Type (1-2)
1 - Regularly sampled
2 - Irregularly sampled
(See structured curves)

d. Horizontal Type (1-2)
1 - Regularly sampled
2 - Irregularly sampled

e. ICTYPE = 4 indicates that the curve data is a structured rather than homogeneous. A call to SURVSDG prior to the creation will set up the curve structure decoding groups and annotation of the fields within the curve structure. NDIMS must be set to one (1) and the datum presentation type (ITYPE) must be three (3) for unsigned, binary byte. Refer to the *Data Type 4: Structured Curve* section for additional information.

f. ICTYPE = 5 indicates that the curve data is a composite rather than homogeneous. After the curve is created with IDTYPE = 3, NDIMS = 1, and IDIMS1 set to the number of bytes, one or more calls to SVCCPDEF defines the elements to be placed in the curve. Refer to the *Data Type 5: Composite Curve* section for additional information.

g. IVTYPE = 2 indicates that the curve data is irregularly sampled and the double precision depth control indexing information is embedded in the data at each level. Therefore, the data of an irregularly sampled curve must be created as either an array of at least two double precision variables or as a structure of information that begins with at least one (and preferably two) double precision variables.

File Header Record 8: Wellsite Attributes

The eighth header record contains wellsite information arranged in four quadrants where each quadrant contains attributes which have the same data type.

Table 12: Wellsite Attributes Record (I4BUFWSI) - First Quadrant - ASCII

8 Byte Elements	Bytes	Type	Parameter	Description
1	0001-0008	Char*8	CH8WSIG	"WSI" signature to indicate valid wellsite information record
2-11	0009-0088	Char*80	CH80WELL	Well name
12-21	0089-0168	Char*80	CH80FLD	Field name
22-31	0169-0248	Char*80	CH80COMP	Company name
32-35	0249-0280	Char*32	CH32CTRY	Country name
36-39	0281-0312	Char*32	CH32COUN	County or parish name
40-41	0313-0328	Char*16	CH16STAT	State name
42	0329-0336	Char*8	CH8OCT	Operational control table name
43-47	0337-0376	Char*40	CH40WLOC	Well location
48	0377-0380	Char*4	CH4SECT	Section
	0381-0384	Char*4	CH4TWNSP	Township
49	0385-0388	Char*4	CH4RGN	Range
	0389-0390	Char*2	CH2IPD	Permanent datum (gl or sl)
	0391-0392	Char*2	CH2IDM	Drilling measurement datum (kb, gl, etc.)
50(lh)	0393-0396	Char*4	CH4LAC	Log analysis center (LAC) number
50(rh)-51(lh)	0397-0404	Char*8	CH8WSU	Wellsite units for elevations
51(rh)-53	0405-0424	Char*20	CH20API	API number
54-56(lh)	0425-0444	Char*20	CH20UWI	Unique well identification
56(rh)-59(lh)	0445-0468	Char*24	CH24RDAT	Raw data, date and time
59(rh)-71(lh)	0469-0492	Char*24	CH24ADAT	AFF data, date and time
72(rh)-128	0493-1024	Char	Unused	N / A

Table 13: Wellsite Attributes Record (I4BUFWSI) - Second Quadrant - Floating Point

4 Byte Elements	Bytes	Type	Parameter	Description
257	1025-1028	Real*4	WSLAT	Latitude
258	1029-1032	Real*4	WSLONG	Longitude
259	1033-1036	Real*4	ELEVKB	Kelly bushing elevation
260	1037-1040	Real*4	DRKFLR	Derrick floor elevation
261	1041-1044	Real*4	ELEVSL	Elevation of permanent datum
262	1045-1048	Real*4	DECMAG	Magnetic declination
263	1049-1052	Real*4	ECCNUM	ECC number
264	1053-1056	Real*4	GRNSEA	Ground to sea level elevation
265	1057-1060	Real*4	INCMAG	Magnetic inclination
266-384	1061-1536	Real*4	Unused	N / A
385-512	1537-2048	Real*8	Unused	N / A

Table 14: Wellsite Attributes Record (I4BUFWSI) - Third Quadrant - Integer

4 Byte Elements	Bytes	Type	Parameter	Description
513-640	2049-2560	Int*4	Unused	N / A
641	2561-2562	Int*2	I2MFG	Metric flag
	2563-2564	Int*2	I2MAXV	Maximum curve versions (1-99)
642-768	2565-3072	Int*2	Unused	N / A

Note: 128 Integer*4 values followed by 256 Integer*2 values.

Table 15: Wellsite Attributes Record (I4BUFWSI) - Fourth Quadrant - Binary

1 Byte Elements	Bytes	Type	Parameter	Description
3073-4096	3073-4096	Unused	Unused	Unformatted binary information

DATA TYPES

The XTF format supports the storage of various categories of data as named data types. The format currently accommodates both a predefined set of standard data types and also optional application-defined data types:

- *Data Type 1: Bitmap*
- *Data Type 2: Curve Header and Data*
- *Data Type 3: Extended Curve Header*
- *Data Type 4: Structured Curve*
- *Data Type 5: Composite Curve*
- *Data Type 6: Wellvar*
- *Data Type 7: Namezone*
- *Data Type 8: Service Header*
- *Data Type 9: Zonelist*
- *Data Type 10: Application Defined*

The locations of the data blocks are independent of their type, but are stored relative to their time of creation within the XTF survey file. This allows and causes intermixing of curves, bitmaps, and defined data types.

Data Type 1: Bitmap

Refer to the *File Header Record 2: Bitmap* for the descriptions and data layout of these additional blocks. Type link-listing of bitmap information allows for additional blocks when necessary. The Bitmap data records are used in the management of the record allocations within the XTF file.

Data Type 2: Curve Header and Data

Basic curve header records are subdivided into four quadrants. Some information in these quadrants are also contained in the file header records 3-7. These quadrants are laid out as:

- 1024 bytes of ASCII (Character data)
- 1024 bytes of floating point (128 Real*4 followed by 64 Real*8)
- 1024 bytes of integer (128 Integer*4 followed by 256 Integer*2)
- 1024 bytes of binary data

The system code is stamped into byte 4096 (last byte of the curve header record). In files created with earlier versions of the software that did not have the system code stamped, the value will be zero, but it can be assumed that it is in the same system format as the survey file's system code.

In the following descriptions, the parameter name can be used in one of the SVCH*xx*GT and SVCH*xx*PT routines to retrieve and put specific curve header information data where *xx* is "AS" for ASCII, "FL" for floating point or real data, "I2" for Integer*2 or short data, or "I4" for Integer*4 or long data.

Table 16: Curve Header Record - First Quadrant - ASCII

8 Byte Elements	Bytes	Type	Parameter	Description
1	1-8	Char*8	CHCURV	Curve name - eight chars that must begin with alpha, the rest can be alphanumeric, no embedded blanks or other characters (also see LONGNAME)
2	9-16	Char*8	CHUNITS	Curve units mnemonic - must be left-justified and contain printable characters (also see LONGUNIT)
3, 4, 5	17-40	Char*24	CHCRVCMT	Primary curve comment field - (i.e., AFF file specification comment
6	41-48	Char*8	CHSERVCO	Service company ('ATLAS ', etc.)
7	49-56	Char*8	CHTOOLTY	Tool type ('ARRAY ', etc.)
8	57-64	Char*8	CHCURVTY	Curve CLASS type ('DIPLOG ', etc.)

Table 16: Curve Header Record - First Quadrant - ASCII (Continued)

8 Byte Elements	Bytes	Type	Parameter	Description
9	65-72	Char*8	CHDEPUNT	Units of primary level indexing - provides units meaning for RLEVCV, DEPTOP and DEPBOT below (lowercase)
10	73-80	Char*8	CHTIMINC	Time increment units for waveform curves (units of time between samples)
11	81-88	Char*8	CHSTIMUN	Start time units (such as array acoustic curves which may have a fixed or variable start time) for STARTIM
12	89-96	Char*8	CHTAPHED	Raw tape header mnemonic - related to variables LNTAPHED and I2TAPHED (refer to CHDRTHGT and CHDTRTPT routines).
13	97-104	Char*8	CHSTADEP	Units of stationary depth for STADEPTH attribute
14	105-112	Char*8	CHCPUNTS	Correlogram parameter units (refer to CHDRCPGT and CHDRCPPT routines)
15	113-120	Char*8	CHBOMNEM	Orientation Reference. ('High', 'North', etc.) (Refer to CHDRBOGT and CHDRBOPT routines)
16	121-128	Char*8	DIRECTON	Logging direction ('up', 'down'. etc.)
17-19	129-152	Char*24	AIFCNAM	AIF's curve name, '.', and version
20	153-160	Char*8	OCTNAM	OCT used to create this curve
21 -26	161-208	Char*48	TOOLNAM	Array of eight - six-character tool names
27	209-216	Char*8	IDXUNIT2	Units of secondary level index
28	217-224	Char*8	Unused	N / A
29	225-228	Char*4	LISNAME	LIS curve mnemonic
	229-232	Char*4	Unused	N / A
30-31	233-248	Char*16	CREATOR	Creation routine name and version
32-33	249-264	Char*16	WRITER	Last writing routine name and version
34	265-268	Char*4	RUN	Run value from AFF file
	269-272	Char*4	TRIP	Trip value from AFF file
35-37(lh)	273-292	Char*20	MODOCT	Modified OCT name
37(rh)-53(lh)	293-420	Char*128	LONGNAME	Long curve name
53(rh)-59(lh)	421-468	Char*48	CURVDESC	Long curve description
59(rh)-61	469-488	Char*20	LONGUNIT	Long curve unit description

Table 16: Curve Header Record - First Quadrant - ASCII (Continued)

8 Byte Elements	Bytes	Type	Parameter	Description
62-128	489-1024	Char	Unused	N / A

Table 17: Curve Header Record - Second Quadrant (First Half) - Floating Point

4 Byte Elements	Bytes	Type	Parameter	Description
257	1025-1028	Real*4	DEPTOP	Top index of curve (depth, time)
258	1029-1032	Real*4	DEPBOT	Bottom index of curve (depth, time)
259	1033-1036	Real*4	RLEVCV	Level spacing of curve
260	1037-1040	Real*4	CURVMIN	Curve minimum value
261	1041-1044	Real*4	CURVMAX	Curve maximum value
262	1045-1048	Real*4	CURVAVG	Curve average value
263	1049-1052	Real*4	TIMEINC	Time increment for time-based curves (Units in CHTIMINC)
264	1053-1056	Real*4	STARTIM	If non-zero, array acoustic offset start time
265	1057-1060	Real*4	STADEPTH	Stationary survey depth (Units in CHSTADEP)
266	1061-1064	Real*4	STDDEV	Standard deviation
267	1065-1068	Real*4	CPWINBEG	Correlogram parameter, window begin
268	1069-1072	Real*4	CPWINSTP	Correlogram parameter, window step
269	1073-1076	Real*4	CPWINLEN	Correlogram parameter, window length
270	1077-1080	Real*4	CBILODEG	CBIL orientation degrees
271	1081-1084	Real*4	TRAN2REC	Transmitter to receiver distance in curve units
272	1085-1088	Real*4	REC2REC	Receiver to receiver distance in curve units
273	1089-1092	Real*4	DEPOFF	Original data offset measured to tool reference point
274	1093-1096	Real*4	DATADLAY	Delay of data from sensor along primary index
275	1097-1100	Real*4	BUTANGL	Angle between reference arm and arm containing the button (sensor)
276	1101-1104	Real*4	BUTXLOC	X location of the button on its pad
277	1105-1108	Real*4	BUTYLOC	Y location of the button on its pad
278-283	1109-1132	Real*4	SMVALSFL	Array of six(6) secondary missing datum values

Table 17: Curve Header Record - Second Quadrant (First Half) - Floating Point (Continued)

4 Byte Elements	Bytes	Type	Parameter	Description
284-384	1133-1536	Real*4	Unused	N / A

Table 18: Curve Header Record - Second Quadrant (Second Half) - Double Precision

8 Byte Elements	Bytes	Type	Parameter	Description
193	1537-1544	Real*8	DBLINIT	Datum initialization value for 8 byte double precision data
194	1545-1552	Real*8	DBTOPDEP	Double precision top index value
195	1553-1560	Real*8	DBBOTDEP	Double precision bottom index value
196	1561-1568	Real*8	DBRLEVEL	Double precision level spacing
197-202	1569-1616	Real*8	SMVALSDB	Array of six (6) secondary missing datum values
203	1617-1624	Real*8	IDXTOP2	Top of secondary level index
204	1625-1632	Real*8	IDXBOT2	Bottom of secondary level index
205	1633-1640	Real*8	IDXRLEV2	Secondary level index spacing
206-256	1641-2048	Real*8	Unused	N / A

Table 19: Curve Header Record - Third Quadrant (First Half) - Integer*4

4 Byte Elements	Bytes	Type	Parameter	Description
513	2049-2052	Int*4	IECDATE	Encoded creation date
514	2053-2056	Int*4	IECTIME	Encoded creation time
515	2057-2060	Int*4	IEADATE	Encoded last access date
516	2061-2064	Int*4	IEATIME	Encoded last access time
517	2065-2068	Int*4	NCRVHDR	Number of curve header records
518	2069-2072	Int*4	LASTREC	Record number of last curve data record
519	2073-2076	Int*4	I4INIT R4INIT	Datum initialization value for 4-byte data (Integer*4 or Real*4 according to sample data type)
520	2077-2080	Int*4	NLEVLS	Number of levels for the curve
521	2081-2084	Int*4	IESDATE	Current encoded date signature
522	2085-2088	Int*4	IESTIME	Current encoded time signature

Table 19: Curve Header Record - Third Quadrant (First Half) - Integer*4 (Continued)

4 Byte Elements	Bytes	Type	Parameter	Description
523-528	2089-2112	Int*4	SMVALSI4	Array ofsix(6) secondary missing datum values
529-530	2113-2120	Int*4	STRUCRCS	Array of two(2) values representing first and last record for structured curve definition
531-532	2121-2128	Int*4	ANNOTRCS	Array of two(2) values representing first and last record for structured curve annotation
533	2129-2132	Int*4	EXHDSIG	Extended header signature (12548)
534-535	2133-2140	Int*4	EXHDINDX	Array of 2 values representing extended header index and count
536	2141-2144	Int*4	COMPSTRT	First composite definition record
537	2145-2148	Int*4	COMPEND	Last composite definition record
538	2149-2152	Int*4	COELECNT	Number of composite elements
539	2153-2156	Int*4	COELESIZ	Size of composite element (128 bytes)
540	2157-2160	Int*4	COMPDSIZ	Composite curve data length in bytes
541-640	2161-2560	Int*4	Unused	N / A

Table 20: Curve Header Record - Third Quadrant (Second Half) - Integer*2

2 Byte Elements	Bytes	Type	Parameter	Description
1281	2561-2562	Int*2	IMAJOR	SURVLIB library major version
1282	2563-2564	Int*2	IMINOR	SURVLIB library minor version
1283	2565-2566	Int*2	IHCURV	Curve handle at time of creation
1284	2567-2568	Int*2	ICTYPE	Curve type (1-5) (See footnotes “a”, “e”, “f”)
1285	2569-2570	Int*2	IDTYPE	Curve sample data type (1-11) (See footnote “b”)
1286	2571-2572	Int*2	IVTYPE	Vertical index type (1-2) (See footnotes “c” and “g”)
1287	2573-2574	Int*2	IHTYPE	Horizontal index type (1-2) (See footnote “d”)
1288	2575-2576	Int*2	NDIMS	Number of dimensions
1289	2577-2578	Int*2	IDIMS1	Number of elements in dimension 1
1290	2579-2580	Int*2	IDIMS2	Number of elements in dimension 2
1291	2581-2582	Int*2	IDIMS3	Number of elements in dimension 3
1292	2583-2584	Int*2	IDIMS4	Expansion space for dimension 4 (unused)

HOME

Table 20: Curve Header Record - Third Quadrant (Second Half) - Integer*2 (Continued)

2 Byte Elements	Bytes	Type	Parameter	Description
1293	2585-2586	Int*2	IDIMS5	Expansion space for dimension 5 (unused)
1294	2587-2588	Int*2	IDIMS6	Expansion space for dimension 6 (unused)
1295	2589-2590	Int*2	IDIMS7	Expansion space for dimension 7 (unused)
1296	2591-2592	Int*2	I2INIT	Datum initialization value for 2-byte integer data
1297	2593-2594	Int*2	INTFLFIX	Flag for Integer / Float / Fixed
1298	2595-2596	Int*2	ISIGN	Signed / unsigned indicator
1299	2597-2598	Int*2	NUMBITS	Number of bits per datum element
1300	2599-2600	Int*2	LISREPCD	Positive value - curve is in raw LIS rep code form (ICVREPCD)
1301	2601-2602	Int*2	LISNSAMP	Nsamp associated with LISREPCD
1302	2603-2604	Int*2	LISRCSIZ	LIS rep code size in bytes
1303	2605-2606	Int*2	LISNSIZE	LIS total number of bytes per level
1304	2607-2608	Int*2	ISTARTFV	Start time definition 0 - Start time for acoustic is fixed 1 - Start time is variable
1305	2609-2610	Int*2	IGAINFV	Array acoustic gain 0 - Gain on array acoustic is fixed
1306	2611-2612	Int*2	IGAINEPX	Gain exponent for array or curve with fixed gain
1307	2613-2614	Int*2	IGAINMTH	Gain method for array or curve with gain, either fixed or variable
1308	2615-2616	Int*2	IGANAPP	Gain applied flag
1309	2617-2618	Int*2	LNTAPHED	Length of RAWTPHD
1310	2619-2620	Int*2	NTOOLS	Number of tools
1311	2621-2622	Int*2	IDXINGRP	Index of curve in ordered group of curves
1312-1317	2623-2634	Int*2	SMVALS12	Array of six(6) secondary missing datum values
1318	2635-2636	Int*2	INVERTED	Inverted section flag (0=normal, 1=inverted borehole section)
1319	2637-2638	Int*2	SEQUENCE	Segment number in inverted borehole space
1320-1536	2639-3072	Int*2	Unused	N / A

a. Curve Type (1-5)
1 - Conventional
2 - Waveform
3 - Matrix
4 - Structure
5 - Composite

b. Curve Sample Data Type (1-11)
1 - Bit
2 - Integer*2 (short)
3 - Unsigned byte
4 - Real*4
5 - 12 bit signed
6 - 12 bit unsigned
7 - Character
8 - Double precision
9 - Unsigned integer*2 (ushort)
10 - Integer*4 (signed long)
11 - Unsigned integer*4 (ulong)

c. Vertical Type (1-2)
1 - Regularly sampled
2 - Irregularly sampled
(See structured curves)

d. Horizontal Type (1-2)
1 - Regularly sampled
2 - Irregularly sampled

e. ICTYPE = 4 indicates that the curve data is a structured rather than homogeneous. A call to SURVSDG prior to the creation will set up the curve structure decoding groups and annotation of the fields within the curve structure. NDIMS must be set to one (1) and the datum presentation type (ITYPE) must be three (3) for unsigned, binary byte. Refer to the *Data Type 4: Structured Curve* section for additional information.

f. ICTYPE = 5 indicates that the curve data is a composite rather than homogeneous. After the curve is created with IDTYPE = 3, NDIMS = 1, and IDIMS1 set to the number of bytes, one or more calls to SVCCPDEF defines the elements to be placed in the curve. Refer to the *Data Type 5: Composite Curve* section for additional information.

g. IVTYPE = 2 indicates that the curve data is irregularly sampled and the double precision depth control indexing information is embedded in the data at each level. Therefore, the data of an irregularly sampled curve must be created as either an array of at least two double precision variables or as a structure of information that begins with at least one (and preferably two) double precision variables.

Table 21: Curve Header Record - Fourth Quadrant - Binary (Not converted between Systems)

Bytes	Parameter	Description
3073-3096	TRANIDS	24 bytes for 24 transmitter id fields (Refer to CHDRTRGT/CHDRTRPT)
3097-3120	RECVIDS	24 bytes for 24 receiver id fields (Refer to CHDRTRGT/CHDRTRPT)
3121-3396	RAWTPHD	276 bytes for raw tape header information
3397-3402	SMVCODES	6 bytes for array of secondary missing values codes
3403-3408	SMVALSBY	6 bytes for array of secondary missing datum values
3409-3416	PROCFLGS	8 bytes for process flags
3417-4095	Unused	N/A
4096-4096	SYSCODE	System code: 1 - PC 2 - Perkin Elmer 3 - Vax 4 - IBM mainframe 5 - Unix -- HP, Sun, SGI

Data Type 3: Extended Curve Header

The extended header is a means of providing a flexible and extensible technique of storing additional curve header entries on a per curve basis. *Figure 2* displays a visual presentation of an extended header. It consists of any number of blocks of data to be called tables.

A table consists of these items:

- Header
- List of field specifications for the record
- Data records in the specified format

Extended Curve Header Definition

An extended curve header is defined when a signature (12548) is placed in EXHDSIG of the curve's header. The two long words (EXHDINDX) after the signature define the relative starting address and the number of table definitions that follow.

Table 22: Curve Header Record - Third Quadrant - Partial

4 Byte Elements	Bytes	Type	Parameter	Description
.	.	.	.	.
.	.	.	.	.
533	2129-2132	Int*4	EXHDSIG	Extended header signature (12548)
534	2133-2136	Int*4	EXHDINDX(1)	Extended header index (points to first table definition)
535	2137-2140	Int*4	EXHDINDX(2)	Extended header count (number of table definitions)
.	.	.	.	.
.	.	.	.	.

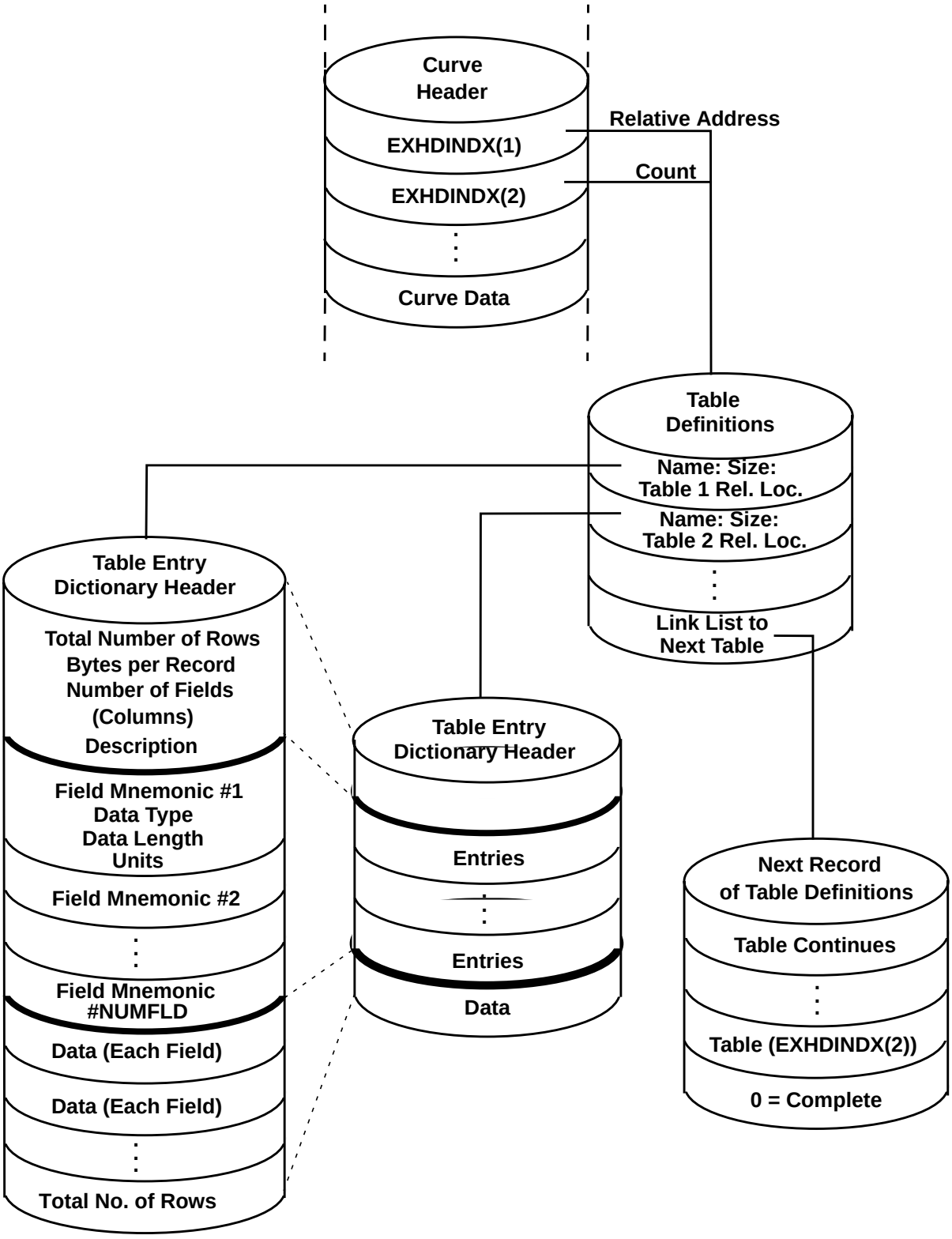

Figure 2

Extended Curve Header

Table Definition

Each of the table definitions consists of one or more records of one or more definitions. There are up to 14 definitions per record and as many records as necessary. Each definition contains information about a table of data including name, code, length, and location of the table relative to the start of the well data file.

Table 23: Extended Curve Header Table Definitions Record

4 Byte Elements	Bytes	Type	Parameter	Description
1-64	1-256	Char*256	TableName	Table definition name (1)
65	257-260	Int*4	TableCode	User-defined table definition code
66	261-264	Int*4	TableLen	Size of table (bytes) being described
67	265-268	Int*4	TableLoc	Relative location of table being described
68	269-272	Int*4	Offset	Undefined
69	273-276	Int*4	Flags	Undefined
70	277-280	Int*4	Unused	N / A
71-134	281-536	Char*256	TableName	Table definition name (2)
135	537-540	Int*4	TableCode	User-defined table definition code
136	541-544	Int*4	TableLen	Size of table (bytes) being described
137	545-548	Int*4	TableLoc	Relative location of table being described
138	549-552	Int*4	Offset	Undefined
139	553-556	Int*4	Flags	Undefined
140	557-560	Int*4	Unused	N / A
.	.	.	.	.
910-974	3640-3896	Char*256	TableName	Table definition name (14)
975	9897-3900	Int*4	TableCode	User-defined table definition code
976	3901-3904	Int*4	TableLen	Size of table (bytes) being described
977	3905-3908	Int*4	TableLoc	Relative location of table being described
978	3909-3912	Int*4	Offset	Undefined
979	3913-3916	Int*4	Flags	Undefined
980	3917-3920	Int*4	Unused	N / A
981-1023	3921-4092	Int*4	Unused	N / A
1024	4093-4096	Int*4	TableLink	Link list to next TableDefinition (0 => last entry)

The TableLoc parameter is used to set a relative address for the table definition's dictionary and data. From that address, until all the data is complete, continuous records are allocated. Each table definition contains a dictionary header and as many dictionary entries as defined in the dictionary prior to the actual data.

Table Dictionary

The dictionary of a table definition contains all the information about the table entries necessary to allocate the space in the well data file. Consider this table to have rows of data and columns of fields.

The fields are defined by table entries of field mnemonics, one for each field (column) that will be used. The TotalRec parameter defines the number of rows of field data that will follow the table entries.

Table 24: Extended Curve Header Table Dictionary Record

Int*2 Elements	Bytes	Type	Parameter	Description
1	1-2	Int*2	Version	Format version generating the data
2	3-4	Int*2	TotalRec	Total number of records (rows) in table
3	5-6	Int*2	RecSize	Number of bytes per record
4	7-8	Int*2	NumFld	Number of fields (columns) in table
5	9-10	Int*2	FirstRec	(Unused) Record number of first record
6-46(lh)	11-91	Char*81	TabTitle	Title - description for reports (80 bytes and null)
46(rh)	92-92	Byte	Unused	(Place on short word boundary)

Table Entry Definition

The NumFld parameter defines how many table column entries will follow this header. The data follows and TotalRec defines how many data records there will be.

Table 25: Extended Curve Header Table Entry Definition Record

Int*2 Elements	Bytes	Type	Parameter	Description
1-5(lh)	1-9	Char*9	DataName	Field mnemonic (Eight chars plus null) #1
5(rh)	10	Byte	Unused	Place back on short word boundary
6	11-12	Int*2	DataType	Sample data type (1-11) (IDTYPE)
7	13-14	Int*2	DataLen	Data length (Bytes)
8	15-16	Int*2	DataUnit	Data units type
9-13(lh)	17-25	Char*9	DataName	Field mnemonic (Eight chars plus null) #2
13(rh)	26	Byte	Unused	Place back on short word boundary
14	27-28	Int*2	DataType	Sample data type (1-11) (IDTYPE)
15	29-30	Int*2	DataLen	Data length (Bytes)
16	31-32	Int*2	DataUnit	Data units type
.	.	.	.	.
.	.	.	.	.
X-X+5(lh)	X-(X+9)	Char*9	DataName	Field mnemonic (Eight chars plus null) *#NumFld*
X+5(rh)	(X+10)	Byte	Unused	Place back on short word boundary
X+6	(X+11) - (X+12)	Int*2	DataType	Sample data type (1-11) (IDTYPE)
X+7	(X+13) - (X+14)	Int*2	DataLen	Data length (Bytes)
X+8	(X+15) - (X+16)	Int*2	DataUnit	Data units type

Table Entry Data

The NumFld parameter specifies the total number of data fields that will be output for each of the TotalRec records. The DataLen and DataType parameters define the size and type of data in each of the fields.

Table 26: Extended Curve Header Table Entry Data Record

Bytes	Description
DataLen #1	Data of DataType #1 for record #1
DataLen #2	Data of DataType #2 for record #1
DataLen #3	Data of DataType #3 for record #1
.	.
.	.
DataLen #*NumFld*	Data of DataType #*NumFld* for record #1
DataLen #1	Data of DataType #1 for record #2
DataLen #2	Data of DataType #2 for record #2
DataLen #3	Data of DataType #3 for record #2
.	.
.	.
DataLen #*NumFld*	Data of DataType #*NumFld* for record #2
.	.
.	.
DataLen #1	Data of DataType #1 for record #*TotalRec*
DataLen #2	Data of DataType #2 for record #*TotalRec*
DataLen #3	Data of DataType #3 for record #*TotalRec*
.	.
.	.
DataLen #*NumFld*	Data of DataType #*NumFld* for record #*TotalRec*

Data Type 4: Structured Curve

The structured curve contains heterogeneous data structures that are similar to tables in a relational database where each level contains the same fields of data. The structure is precisely the same at each level of the curve since it has been pre-defined at curve creation time, and in program memory, it is literally a structure. *Figure 3* illustrates the organization of a structured curve and its components.

Note: The **Composite Curve** (Curve Type 5) has been designed as an improved replacement for the Structured Curve (DataType 4). New applications should use the Composite Curve rather that the Structured Curve. Support for the Structured Curve datatype is retained only for backwards compatibility of applications.

At each level of a structured curve, the curve's datums resemble a table of elements. There can be up to 1023 datum decoding groups where each field can be scalar or an array of datums of up to 32767 items. The group indicates how may decoding groups (elements) there are and the associated arrays indicate the count and datum type of each decoding group (element). Fields cannot be defined as substructures or multi-dimensional arrays, but they may be defined as scalar entities or single-dimension arrays of data.

In the following tables, NGROUPS indicates how may decoding groups there are and the associated arrays I2COUNTS and I2TYPES indicate the count and datum type of each decoding group. Fields cannot be defined as substructures or multi-dimensional arrays, but they may be defined as either scalar entries or single dimension arrays of datums.

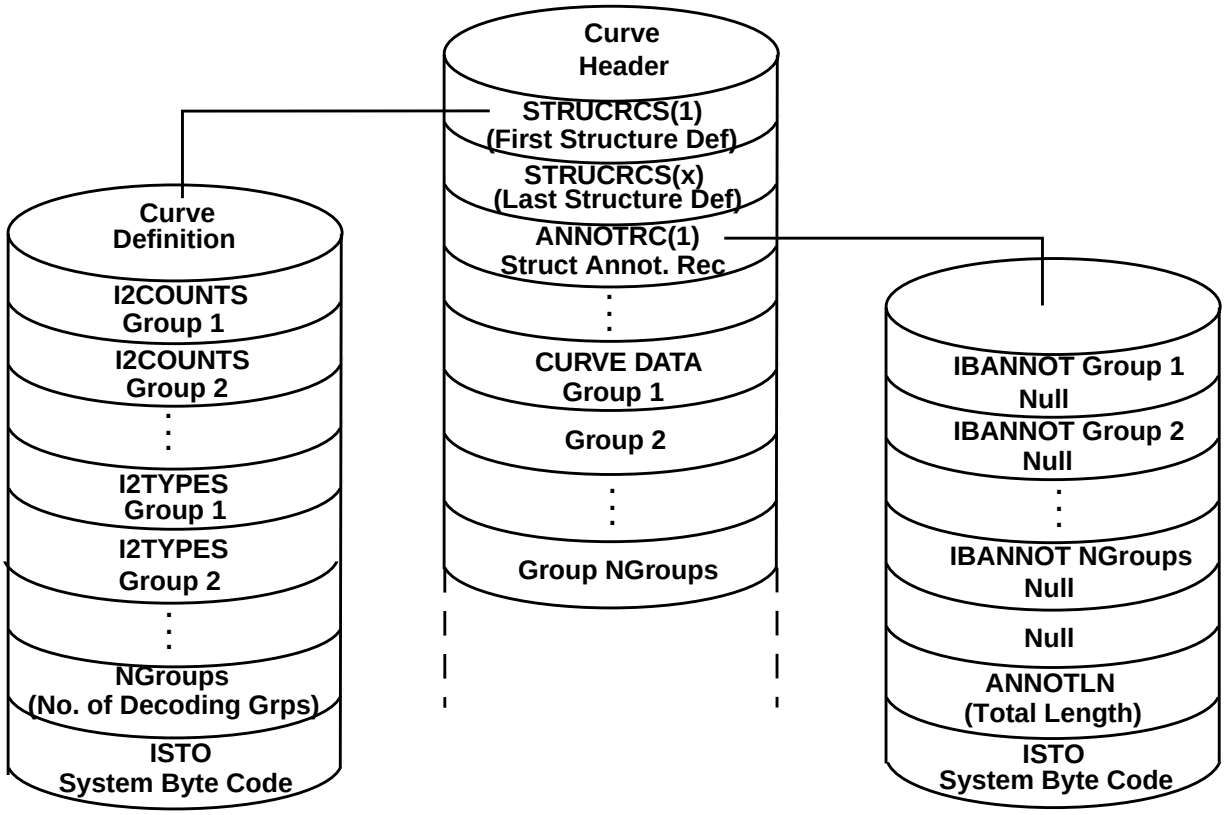

Figure 3

Structured Curve

Structured Curve Definition

A structured curve is defined by ICTYPE=4 (curve type) and pointers in the curve header's third quadrant (Integer*4) area that points to the curve's decoding information.

Table 27: Curve Header - Third Quadrant - Partial

2 Byte Elements	Bytes	Type	Parameter	Description
.	.	.	.	.
.	.	.	.	.
1284	2567-2568	Int*2	ICTYPE	Curve type - Set to structured (4)
1285	2569-2570	Int*2	IDTYPE	Curve sample data type - Set to unsigned byte (3)
.	.	.	.	.
.	.	.	.	.

8 Byte Elements	Bytes	Type	Parameter	Description
.	.	.	.	.
.	.	.	.	.
529	2113-2116	Int*4	STRUCRCS(1)	First structure definition record
530	2117-2120	Int*4	STRUCRCS(2)	Unused - Last structure definition record
531	2121-2124	Int*4	ANNOTRC(1)	First structure annotation record
532	2135-2138	Int*4	ANNOTRC(2)	Unused - Last structure annotation record
.	.	.	.	.
.	.	.	.	.

Structured Curve Element Definition

The elements of a structured curve are described by a structured curve element definition table. This provides the data type and dimensionality of each element in the structured curve.

The following table illustrates the general layout of a structured curve element definition table:

Table 28: Structured Curve Element Definition Record

2 Byte Elements	Bytes	Type	Parameter	Description
1	1-2	Int*2	I2COUNTS(1)	Count of elements in group #1
2	3-4	Int*2	I2COUNTS(2)	Count of elements in group #2
3	5-6	Int*2	I2COUNTS(3)	Count of elements in group #3
.	.	.	.	.
.	.	.	.	.
1023	2046	Int*2	I2COUNTS(*ngroups*)	Count of elements in group #*ngroups*
1024	2047-2048	Int*2	I2TYPES(1)	Sample data type (IDTYPE) of group #1
1025	2049-2050	Int*2	I2TYPES(2)	Sample data type (IDTYPE) of group #2
1026	2051-2052	Int*2	I2TYPES(3)	Sample data type (IDTYPE) of group #3
.	.	.	.	.
.	.	.	.	.
2046	4092	Int*2	I2TYPES(*ngroups*)	Sample data type (IDTYPE) of group #*ngroups*
2047	4093-4094	Int*2	*ngroups*	Number of decoding groups
2048	4095	Int*1	Unused	N / A
	4096	Int*1	ISTO	System code: 1 - PC 2 - Perkin Elmer 3 - Vax 4 - IBM mainframe 5 - Unix -- HP, Sun, SGI

Structured Curve Annotation

The structure annotation is an array of null-terminated strings up to a maximum of 4092 bytes. The last string must be double null-terminated. Each string is of variable length and describes the contents of its respective decoding group. Originally designed to provided annotation information for the structured curve, the annotation record has been used to provide additional information about the structured curve elements and can be considered to be an

extension of the structured curve element definition block.

Table 29: Structured Curve Annotation Record

2 Byte Elements	Bytes	Type	Parameter	Description
1	1 - NULL	Char	IBANNOT(1)	Annotation for group element #1
.	next - NULL	Char	IBANNOT(2)	Annotation for group element #2
.	next - NULL	Char	IBANNOT(3)	Annotation for group element #3
.	.	.	.	.
.	.	.	.	.
.	next - NULL	Char	IBANNOT(X)	Annotation for group element #*ngroups*
.	NULL	Char		Final NULL character, ends annotation list
.	.	Char	None	Uninitialized space holders
.	.	Char	None	Uninitialized space holders
2047	4093-4094	Int*2	ANNOTLN	Annotation length in bytes including double NULL
2048	4095	Int*1	Unused	N / A
	4096	Int*1	ISTO	System code: 1 - PC 2 - Perkin Elmer 3 - Vax 4 - IBM mainframe 5 - Unix -- HP, Sun, SGI

Structured Curve Data Definition

The data for the structure follows the curve's header and is defined as an array of byte data. When decoding the data, each group can be of different length and different data type.

Table 30: Structured Curve Data Definition Record

Bytes	Type	Parameter	Description
1 - (isize1)	I2TYPES(1)	IBGROUP (1)	All data for group #1
(isize1+1) - (isize2)	I2TYPES (2)	IBGROUP (2)	All data for group #2
(isize2+1) - (isize3)	I2TYPES (3)	IBGROUP (3)	All data for group #3
.	.	.	.
.	.	.	.
(isize(N-1)+1)- (isize*n*)	I2TYPES (*ngroups*)	IBGROUP (*ngroups*)	All data for group #*ngroups*

Structured Curve Access Routines

Documentation from the following *libsurv* library routines will be helpful in the use of structured curves.

Table 31: Structured Curve Access Routines

libsurv Library Routine	Description
CVPSMVLS	Put (store) arrays of curve's secondary missing datum values and codes
CVGILVLS	Get levels of structured, irregularly sampled curve that fall within the depth range specified
CVGILVLR	Get levels of structured, irregularly sampled curve that falls within depth range specified. (Review subroutine's documentation as this is unlike CVGILVLS)
CVPILVLS	Put (write) one or more levels of irregularly sampled curve data
CVDILVLS	Delete levels of structured, irregularly sampled curve that fall within the specified depth range
CVGDGINF	Get structured curve group decoding information
CVGSMVLS	Get arrays of curve's secondary missing datum values and codes
CVEDDEPS	Edit depths in an irregular curve
CVGTDEPS	Scan irregular curve to establish its depth limits and live data statistic
SURVSDG	Set curve decoding group parameters for subsequent structure curve creation

In order for the structured curve to be generated, a call to SURVSDG must first be made to define the decoding group parameters.

Data Type 5: Composite Curve

The composite curve, like the structured curve accommodates heterogeneous data structures that are similar to tables in a relational database, where each level contains the same fields of data. the composite curve has, however, been designed to provide a more generic data structure than the structured curve and has eliminated several of the limitations present in the structured curve's design. The major difference is in its element descriptions. *Figure 4* illustrates the organization of a composite curve and its components.

Note: The **Composite Curve** (Curve Type 5) has been designed as an improved replacement for the Structured Curve (DataType 4). New applications should use the Composite Curve rather that the Structured Curve. Support for the Structured Curve datatype is retained only for backwards compatibility of applications.

Create the composite curve using SURVCRCV, set the curve sample data type to three (unsigned, binary byte), set the curve type to five (composite), and vertical sampling to regularly sampled data. The number of dimensions will be one and the dimension will be the total number of bytes in one level of the data elements. These elements are made up of conventional, wave-form, or matrix data.

Once the composite curve is created, a call to SVCCPDEF sets up the initialization and storage area within the well data file for all of the element definitions. These definitions for the elements can be created in a single pass or they can be generated one at a time until the total number of elements have been defined.

When the last definition is generated, the data within the curve is initialized just as individual curves are initialized at creation time. Each operation of SVCCPDEF (Survey Composition Curve Put Definition) places a block of element definition in the well data file. The starting and ending location, plus additional information about the elements are stored in the curve header as defined by the names COMPSTRT (First Composition Definition Record), COMPEND (Last Composite Definition Record), COELECNT (Number of Composite Elements), and COELESIZ (Size of Composition Element - 128 bytes).

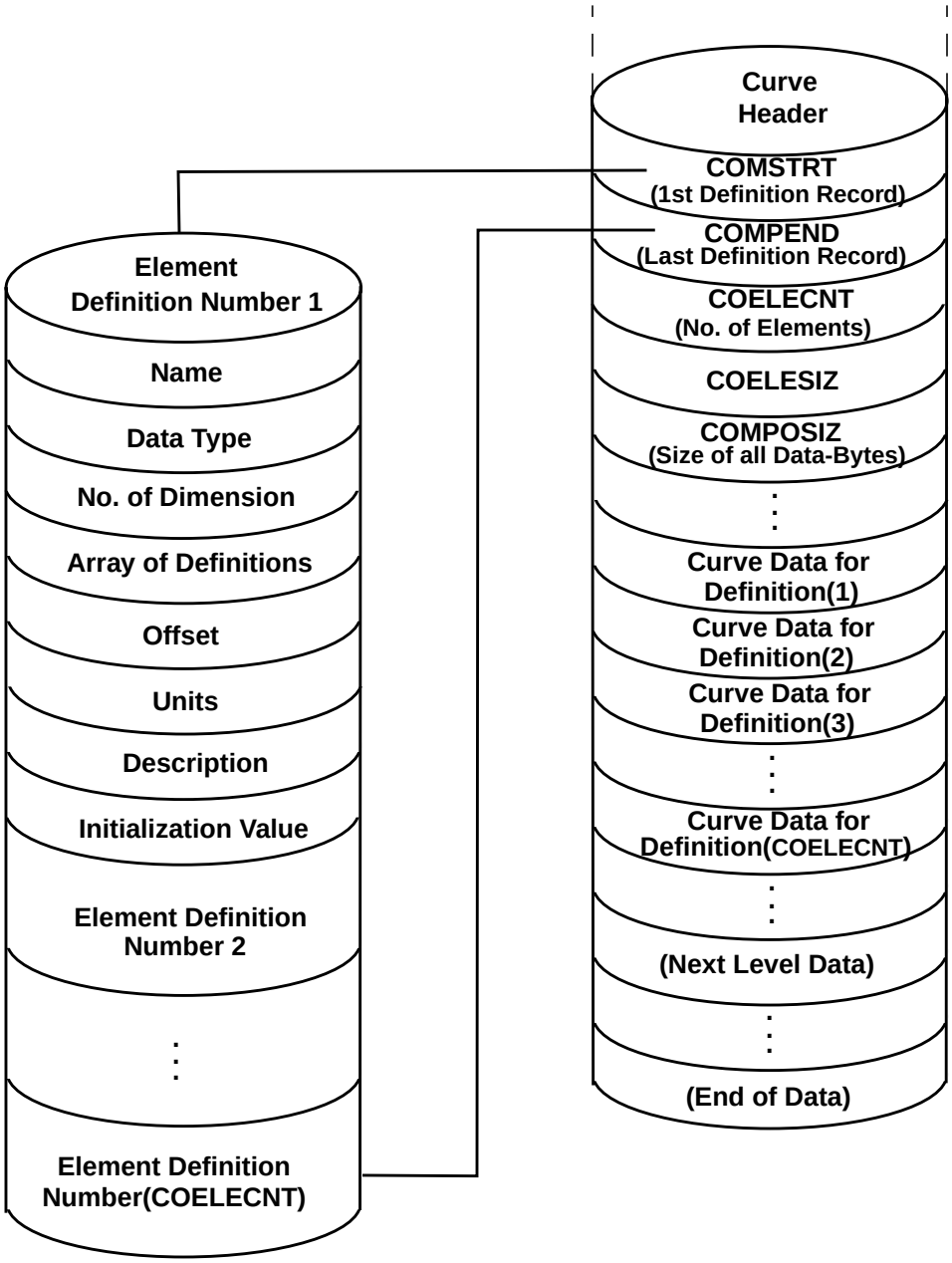

Figure 4

Composite Curve

Composite Curve Definition

A composite curve is defined by ICTYPE=5 (curve type) and pointers in the curve header's third quadrant (Integer*4) area that point to the curve's decoding information.

Table 32: Curve Header - Third Quadrant - Partial

2 Byte Elements	Bytes	Type	Parameter	Description
.	.	.	.	.
.	.	.	.	.
1284	2567-2568	Int*2	ICTYPE	Curve type - Set to composite (5)
1285	2569-2570	Int*2	IDTYPE	Curve sample data type - Set to unsigned byte (3)
.	.	.	.	.
.	.	.	.	.

4 Byte Elements	Bytes	Type	Parameter	Description
.	.	.	.	.
.	.	.	.	.
536	2141-2144	Int*4	COMPSTRT	First composite definition record
537	2145-2148	Int*4	COMPEND	Last composite definition record
538	2149-2152	Int*4	COELECNT	Number of composite elements
539	2153-2156	Int*4	COELESIZ	Size of composite element (128 bytes)
540	2157-2160	Int*4	COMPDSIZ	Composite curve data length (bytes)
541	2161-2154	Int*4	Unused	N / A

Composite Curve Element Definition

Each element of a composite curve is described by a composite curve element definition block. This provides the name, sample data type, dimensionality, position, measurement units, initialization value and textual description of the element.

The following table illustrates the general layout of an element definition block:

Table 33: Composite Curve Element Definition Block

4 Byte Elements	2 Byte Elements	Bytes	Type	Name	Description
1-5	1-10	1-20	Char*20	ElementName	Name
6	11	21-22	Int*2	DataType	Sample data type
	12	23-24	Int*2	NumberDims	Number of dimensions
7-8	13-16	25-32	Int*2	Dims(4)	Array of four (4) dimensions
9	17-18	33-36	Int*4	ByteOffset	Data start point
10-14	19-28	37-56	Char*20	ElementUnits	Unit of measure
15-26	29-52	57-104	Char*48	Description	Description
27-28	53-56	105-112	Int*4	Empty Value(2)	Empty or initialization value
29-32	57-64	113-128	Int*4	Spares(4)	Spares for future needs

Composite Curve Data Definition

The data for the composite curve follows the curve's header and is defined as an array of byte data. When decoding the data, each definition describes the length and data type.

Table 34: Composite Curve Data Definition Block

Bytes	Type	Parameter	Description
Datasize	Datatype(1)	N / A	All data for group #1
Byte offset-Datasize(2)	Datatype(2)	N / A	All data for group #2
Byte offset-Datasize(3)	Datatype(3)	N / A	All data for group #3
.	.	.	.
.	.	.	.
.	.	.	.
Byte offset-Datasize (COELECNT)	Datatype(COELECNT)	N / A	All data for group #*COELECNT*

Composite Curve Access Routines

The following routines access the data and information stored in the composite curve element definition blocks:

Table 35: Composite Curve Access Routines

libsurv Library Routine	Description
SVCCPDEF	Put composite curve's element definitions in well data file
SVCCIDAT	Initialize element data using composite curve definitions
SVCCCNT	Retrieve number of elements in a composite curve
SVCCGNMS	Retrieve all element names in a composite curve
SVCCEVAL	Retrieve element's empty value, offset and data type
SVCCGDEF	Retrieve element's definition information
SVCCGSEQ	Retrieve element's sequence number
SVCCCINF	Copy element definition information from curve to curve
SVCCGINF	Retrieve element definition information for range of elements

Data Type 6: Wellvar

Wellvars are scalar floating point well and symbolic variables which are accessed via name-oriented retrieval. The wellvar datatype is designed to hold numeric scalar variables dealing with the well generally (as opposed to information which is related to a specific zone, curve, or other data item within the well).

The wellvar datatype is self-initializing. That is, in an attempt to write a wellvar to the survey file, if the wellvar datatype is not found, it is created. The wellvar datatype is also self-expanding, so space will always be available to store wellvars as long as there is disk space available. When the current space allotment for the wellvar datatype is filled, and there is an attempt to write a new wellvar to the survey file, more space is allocated. These allocations are made in blocks of 341 wellvars.

Each wellvar block is organized as two sections: one section of 341 eight character symbol names, followed by a second section containing 341 symbol values (4 byte floating point).

Wellvar Definition

The following table illustrates the layout of the wellvar definition record:

Table 36: Wellvar Definition Record

8Byte Elements	4 Byte Elements	Bytes	Type	Description
1	1-2	0001-0008	Char*8	First symbol name
2	3-4	0009-0016	Char*8	Second symbol name
.	.	.	Char*8	Next symbol name
.	.	.	.	.
341	681-682	2721-2728	Char*8	Last symbol name
342	683	2279-2732	Real*4	First symbol value
	684	2733-2736	Real*4	Second symbol value
.	.	.	.	Next symbol value
	.	.	.	.
.	.	.	.	.
	.	.	.	.
512	1023	4089-4092	Real*4	Last symbol value
	1024	4093-4096	Real*4	System code 1-5 (for conversion operation)

Data Type 7: Namezone

The namezone datatype accommodates a named zone delineated by a specified top and bottom depth (or time) and may optionally contained associated zone variables and their values.

The namezone datatype is self-initializing. That is, in an attempt to write a namezone to the survey file, if the namezone datatype is not found, it is created. The namezone datatype is also self-expanding, so space will always be available to store namezones as long as there is disk space available. When the current space allotment for the namezone datatype is filled, and there is an attempt to write a new namezone to the survey file, more space is allocated. These allocations are made in blocks of 512 zones.

Namezone Reserved Record

For historical reasons, the namezone datatype area contains one completely empty 4096 byte record, followed by the actual zone information record(s):

Table 37: Namezone Reserved Record

Elements	Bytes	Description
1	0001-4096	Reserved for future expansion of zone information

Zone Name Definition

The following is the general layout for a zone name:

Table 38: Zone Name Definition Record

8 Byte Elements	Bytes	Type	Description
1	0001-0008	Char*8	Zone name #1
2	0009-0016	Char*8	Zone name #2
.	.	.	.
512	4089-4096	Char*8	Zone name #512
.	.	.	(Expandable in blocks of 512)
.	.	.	.

Zone Variable Definition

The following is the general layout for two blocks of variable names:

Table 39: Zone Variable Name Definition Record(s)

8 Byte Elements	Bytes	Type	Description
1	0001-0008	Char*8	Variable name #1
2	0009-0016	Char*8	Variable name #2
- Second-	- Block -		- Continues past 512 variables -
1022	8171-8178	Char*8	Variable name #1022 for all zones
1023-1024	8177-8192	Char*16	Unused

The following is the general layout for variable data for each zone:

Table 40: Zone Variable Value Definition Records

4 Byte Elements	Bytes	Type	Description
1	0001-0004	Real*4	Variable value for variable #1
2	0005-0008	Real*4	Variable value for variable #2
.	.	.	.
1022	4081-4088	Real*4	Variable value for variable #1022
1023-1024	4089-4096	Char*8	Zone name (1) for this data

4 Byte Elements	Bytes	Type	Description
1	0001-0004	Real*4	Variable value for variable #1
2	0005-0008	Real*4	Variable value for variable #2
.	.	.	.
1022	4081-4088	Real*4	Variable value for variable #1022
1023-1024	4089-4096	Char*8	Zone name (2) for this data

. . . .

. . . .

Continue with these blocks until all zones are complete

. . . .

. . . .

4 Byte Elements	Bytes	Type	Description
1	0001-0004	Real*4	Variable value for variable #1
2	0005-0008	Real*4	Variable value for variable #2
.	.	.	.
1022	4081-4088	Real*4	Variable value for variable #1022
1023-1024	4089-4096	Char*8	Zone name (*n*) for this data

Data Type 8: Service Header

The service header datatype is self-initializing. That is, in an attempt to write a service header to the survey file, if the service header datatype is not found, it is created. The service header datatype is also self-expanding, so space will always be available to store service headers as long as there is disk space available. When the current allotment of space for the service header datatype is filled, and there is an attempt to write a new service header to the survey file, more space is allocated. These allocations are made in blocks of 4 service headers. Refer to routines RDSERV, WTSERV, RDWDINF, EDT008 for additional information.

Service Header Definition

The following table illustrates the layout of the service header definition record:

Table 41: Service Header Definition Record

Bytes	Type	Parameter	Description
0001-0004	Int*2	ISERV(2)	Service name
0005-0010	Int*2	ISVCD(3)	Unused
0011-0012	Int*2	IRUN	Run number
0013-0020	Int*2	IDATE(4)	Date well was logged
0021-0026	Int*2	ISON(3)	Service order number
0027-0028	Int*2	ILDP	Logging measured from (reference)
0029-0032	Real*4	DP	Depth driller measured
0033-0036	Real*4	DL	Depth logger measured
0037-0040	Real*4	BLD	Bottom of logged interval
0041-0044	Real*4	TLD	Top of logged interval
0045-0048	Real*4	CSZD	Casing size
0049-0052	Real*4	CSZDD	Bottom depth of casing (driller)
0053-0056	Real*4	CSZLD	Bottom depth of casing (logger)
0057-0060	Real*4	BITS	Bit size
0061-0072	Int*2	IFLDT(6)	Borehole fluid
0073-0076	Real*4	FDN	Drilling fluid density
0077-0080	Int*2	IFDNU(2)	Drilling fluid density units

Table 41: Service Header Definition Record (Continued)

Bytes	Type	Parameter	Description
0081-0084	Real*4	FV	Drilling fluid viscosity
0085-0088	Real*4	PH	Drilling fluid pH
0089-0092	Real*4	FLOSS	Drilling fluid loss
0093-0096	Int*2	ISAMS(2)	Mud sample source
0097-0100	Real*4	RMTMP1	Mud sample resistivity
0101-0104	Real*4	RMTMP2	Mud sample temperature
0105-0108	Real*4	RMFTMP1	Mud filtrate sample resistivity
0109-0112	Real*4	RMFTMP2	Mud filtrate sample temperature
0113-0116	Real*4	RMCTMP1	Mud cake sample resistivity
0117-0120	Real*4	RMCTMP2	Mud cake sample temperature
0121-0124	Int*2	IRMFS(2)	Mud filtrate sample source
0125-0128	Int*2	IRMCS(2)	Mud cake sample source
0129-0132	Real*4	RMBHT	Resistivity of RM at BHT
0133-0136	Real*4	BHT	Bottom hole temperature
0137-0140	Real*4	TMAX	Maximum recorded temperature
0141-0144	Real*4	TIMEC	Time since circulation
0145-0148	Int*2	IEQP(2)	Logging unit number
0149-0152	Int*2	IELOC(2)	Logging unit location
0153-0156	Real*4	TDEPS	Unused
0157-0160	Real*4	BDEPS	Unused
0161-0168	Int*2	ITS1(4)	Tool series number
0169-0176	Int*2	ITS2(4)	Tool serial number
0177-0184	Int*2	ITS3(4)	Orientation type
0185-0192	Int*2	ITS4(4)	Orientation series number
0193-0200	Int*2	ITS5(4)	Panel series number
0201-0208	Int*2	ITS6(4)	Panel serial number
0209-0216	Int*2	ITS7(4)	Electronics number
0217-0224	Int*2	ITS8(4)	Tape series number
0225-0232	Int*2	ITS9(4)	Tape serial number
0233-0240	Int*2	ITSA(4)	Plotted by

Table 41: Service Header Definition Record (Continued)

Bytes	Type	Parameter	Description
0241-0248	Int*2	ITSB(4)	Unused
0249-0256	Int*2	ITSC(4)	Unused
0257-0260	Real*4	TLN	Unused
0261-0460	Int*2	ICVNM(100)	Unused
0461-0560	Int*2	ITRK(50)	Unused
0561-0660	Real*4	SCLL(25)	Unused
0661-0760	Real*4	SCLR(25)	Unused
0761-0860	Real*4	CHDEP(25)	Unused
0861-0864	Int*2	IGRID(25)	Unused
0865-0884	Char*20	IENG(10)	Engineer's name
0885-0904	Char*20	IWITN(10)	Witness's name
0905-0984	Char*80	ICOMH(40)	Comments
0985-1024	Real*4	FILB(10)	Unused

Data Type 9: Zonelist

Zonelist datatypes are set up to hold depth pairs where conditional rules are satisfied. There can be up to 51 sets per given record, however, it is self-expanding and can have as many records as necessary.

The start of pairs of records and end of pairs of records contain pointers to records which contain the list I4COUNT double precision depth pairs. Each depth pair record can hold 256 depth pairs. A contiguous set of records is allocated to hold the depth pairs (256 depth pairs per record).

Zonelist Definition

The following table illustrates the layout of the zonelist definition record:

Table 42: Zonelist Definition Record

4 Byte Elements	Bytes	Type	Parameter	Description
1-2	0001-0008	Char*8	CHNAMEZL	Zonelist name
3-4	0009-0016	Char*8	CHDPUNTS	Unit of index (depth)

Table 42: Zonelist Definition Record (Continued)

4 Byte Elements	Bytes	Type	Parameter	Description
5-9	0017-0036	Char*20	CH20COND	Conditional text info (e.g. “abc.gt.xyz”)
10	0037-0040	Int*4	I4COUNT	Number of interval pairs
11-12	0041-0048	Real*8	RSSCNTOP	Overall scan top depth that created zonelist
13-14	0049-0056	Real*8	RSSCNBOT	Overall scan bottom depth that created zonelist
15-16	0057-0064	Real*8	R8MINDEP	Minimum depth contained in any depth pairs
17-18	0065-0072	Real*8	R8MAXDEP	Maximum depth contained in any depth pairs
19	0073-0076	Int*4		Start of pairs record
20	0077-0080	Int*4		End of pairs record

Zonelist Condition-of-Scan

The 20-byte condition-of-scan field is in the form of:

XXXXXXXX.op.ZZZZZZZZ

where *XXXXXXXX* and *ZZZZZZZZ* can be a curve or zonelist name and the operator “*.op.*” can be any of those listed in the following table:

Table 43: Zonelist Condition-of-Scan Operators

Operator	Description
.gt.	Greater than
.ge.	Greater than or equal to
.eq.	Equal to
.ne.	Not equal to
.le.	Less than or equal to
.lt.	Less than
.an.	And
.or.	Or
.xo.	Exclusive or
.no.	Not

The depth pairs contained in the zonelist are the set of depth pairs which satisfied the *condition-of-scan* at the time the zonelist was created.

Data Type 10: Application Defined

The *libsurv* interface has built-in routines to help the user create and manage their own “datatype”. Space is allocated for user datatypes by giving it a name and the number of records necessary to add the data type (SURVADDT). The following tasks are available:

Table 44: Datatype Access Routines

Task / Function	libsurv Routine
Expand datatype	SURVXPDT
Delete datatype	SURVDLDT
Rename datatype	SURVRNDL
Find datatype	SURVFNDT
Find datatype associated records	SURVNNDT
Read data	SURVRDDT
Write data	SURVWRDT

There is no automatic conversion of the data between systems for user generated datatypes. Therefore, it is the responsibility of the creator to apply these in a given system mode or to check the system code for conversion prior to reading the defined format.