Performance Oscilloscope Reference Waveform File Format

The Tektronix .wfm file format was designed for the internal save and restore of waveform data and the associated display. As a consequence, more parameters are saved than are required to transfer and create an external Time vs. Voltage value list. The format will handle XY waveform data, XYT data, and pixel maps, such as the waveform database. FastFrame waveforms (multiple acquisitions with individual timestamps) can also be represented. The Tektronix .wfm file format is not intended for you to use to create a waveform for reintroduction into the instrument.

NOTE. *The Tektronix .wfm file format is subject to change in future software versions.*

This document specifies the reference waveform file format for the following instrument families:

Instrument series			
TDS5000/B	TDS6000/B/C	TDS/CSA7000/B	DPO7000
DPO70000/B	DSA70000/B	MSO70000	

The Tektronix .wfm file format is partially based on the SCPI Data Interchange Format. If you are familiar with that standard, some of the variable names and usage will be familiar.

Waveforms are described by X or Y dimension data and Curve data. Curve data for an acquired waveform is in integer format and includes some extra points at the beginning and ending of the data that are normally not seen on screen. These extra points are real acquired data points and are used for display interpolation purposes. There are normally 32 extra points; 16 PreCharge data points and 16 PostCharge data points.

The dimensions can be explicit or implicit. Normal YT (Voltage vs. Time) waveforms have an implicit dimension for time and an explicit dimension for Voltage.

For implicit dimensions, the real value, in this case time, is computed by (Waveform point index * dimensionScale + dimensionOffset).

www.tektronix.com

077-0220-00

Tektronix

For the explicit dimension, the real value, in this case voltage, is computed by (curveBufferPoint * dimensionScale + dimensionOffset).

NOTE. *For Fast Acquisition or Waveform Database waveforms, there are two implicit dimensions and one explicit dimension. Implicit dimension 1 is time and implicit dimension 2 is Volts. Explicit dimension 1 is number of hits at a specific Time and Voltage coordinate.*

For all other Time-Voltage waveforms, implicit dimension 1 is time and explicit dimension 2 is Voltage.

To download software that will convert the waveform into different formats, go to: http://www.tek.com/site/sw/search/ and search for "WFM TO ASCII CONVERTER" software in the Keyword category.

File Layout Overview

The following provides an overview of the waveform header file layout:

Information type		Size	
Waveform static file information		[78 bytes]	
Waveform header			
Reference file data		[88 bytes]	
Explicit Dimension 1		(Usually defines voltage axis)	
	[Description info]	[100 bytes]	
	[userView data]	[56 bytes]	
Explicit Dimension 2			
	[Description info]	[100 bytes]	
	[userView data]	[56 bytes]	
Implicit Dimension 1		(Usually defines time axis)	
	[Description info]	[76 bytes]	
	[userView data]	[56 bytes]	
Implicit Dimension 2			
	[Description info]	[76 bytes]	
	[userView data]	[56 bytes]	
TimeBase Info1		[12 bytes]	
TimeBase Info2		[12 bytes]	
WfmUpdateSpec		[24 bytes]	(Defines 1st waveform of a waveform set)
WfmCurveObject		[30 bytes]	(Defines 1st waveform of a waveform set)
FastFrame Frames {0 to N-1 where N = number of frames}			
N-1 WfmUpdateSpecs		[24 * (N-1) bytes]	
N-1 WfmCurveObjects		[30 * (N-1) bytes]	

CurveBuffer		
	{Variable length; this contains all curve data in the specified format}	
	WfmFileChecksum	{8 bytes}

Internal File Format

Listed below is the current (April 2004 - 2009) file format with descriptions:

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Waveform static file information				
0 [0x000]	2 bytes	Byte order verification	Unsigned short	Hex 0xF0F0 (PPC format) or 0x0F0F (Intel). Used to determine if bytes read from file need to be swapped before processing.
2 [0x002]	8 bytes	Version number [1]	Char[8]	The values will correspond to the following characters: WFM#001 [6], where 001 is replaced by the actual file version. The byte values will be (in hex) 3a 57 46 4d 23 30 30 31 (where 0x303031 == 001).
10 [0x00a]	1 byte	Num digits in byte count	Char	0 – 9
11 [0x00b]	4 bytes	Number of bytes to the end of file (EOF)	Long int	Up to 9 digits in length, which allows for a file size of 999,999,999 bytes. The number of bytes from here to the end of the file.
15 [0x00f]	1 byte	Number of bytes per point	Char	Number of bytes per curve data point. Allows for quick determination of the size of the actual curve data.
16 [0x010]	4 bytes	Byte offset to beginning of curve buffer [2]	Long int	Allows for positioning of file descriptor for recall of the curve buffer. The number of bytes from the beginning of the file to the start of the curve buffer.
20 [0x014]	4 bytes	Horizontal zoom scale factor	Long int	Horizontal scale zoom information. Not for use.
24 [0x018]	4 bytes	Horizontal zoom position	Float	Horizontal position zoom information. Not for use.
28 [0x01c]	8 bytes	Vertical zoom scale factor	Double	Vertical scale zoom information. Not for use.
36 [0x024]	4 bytes	Vertical zoom position	Float	Vertical position zoom information. Not for use.
40 [0x028]	32 bytes	Waveform label	Char[32]	User defined label for the reference waveform.
72 [0x048]	4 bytes	N (number of FastFrames – 1)	Unsigned long	The number of WfmUpdateSpec and Curve objects that follow. This number is equal to the number of FastFrames – 1 (for FastFrame waveform sets).
76 [0x04c]	2 bytes	Size of the waveform header	Unsigned short	The size in bytes of the waveform header, which directly follows this field.
Waveform header				
78 [0x04e]	4 bytes	SetType	Enum (int)	Type of waveform set. 0 = Single waveform set 1 = FastFrame set
82 [0x052]	4 bytes	WfmCnt	Unsigned long	Number of waveforms in the set. FastFrame is a special case in that it describes a waveform set of one waveform with multiple frames. See the numFrames field described below.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
86 [0x056]	8 bytes	Acquisition Counter	Unsigned long long	Internal acquisition counter. This is not a time stamp. It is used to make sure that a set of acquisition waveforms are of the same acquisition. This counter is the number of acquisitions since power on. If nil (0), the internal system will match any waveform with this one. Not for use.
94 [0x05E]	8 bytes	Transaction counter	Unsigned long long	Internal acquisition translation stamp. This is not a time value. It is the time of the transaction that this waveform is based upon. Not for use.
102 [0x066]	4 bytes	Slot ID	Enum (int)	An enumeration based upon the number and type of data slots available in a specified product. These values are subject to change based upon the need of the specified product. Not for use.
106 [0x06a]	4 bytes	Is static flag	Int	Used internally to determine if waveform is static (for example, a reference) or live (for examples, a channel or mat). Not for use.
110 [0x06e]	4 bytes	Wfm update specification count	Unsigned long	Number of wfm update specifications in the waveform set. The waveform update spec holds data, which changes on each acquisition update. FastFrame waveform sets have multiple update specs (for example, each frame has unique timestamps and such).
114 [0x072]	4 bytes	Imp dim ref count	Unsigned long	The number of implicit dimensions for the given waveform. Vector Lists (vector YT and XY) waveform sets have one implicit dimension; Pixel Maps (YT, XY, and XYZ) waveform sets have one explicit dimension; and Scalars (measurements and constants) have zero implicit dimensions.
118 [0x076]	4 bytes	Exp dim ref count	Unsigned long	The number of explicit dimensions for the waveform set. Vector Lists (vector YT, XY, and XYZ) waveforms have one (YT) or two (XY) explicit dimension(s); Pixel Maps (YT, XY, and XYZ) waveform sets have two implicit dimensions; and Scalars (measurements and constants) have zero explicit dimensions. Implicit dimension axis values are determined by a value plus index times increment. Explicit dimension axis values are explicitly defined for each data point.
122 [0x07a]	4 bytes	Data type	Enum (int)	0 = WFMDATA_SCALAR_MEAS 1 = WFMDATA_SCALAR_CONST 2 = WFMDATA_VECTOR (normal YT waveforms) 4 = WFMDATA_INVALID 5 = WFMDATA_WFMDB (for example Waveform database)

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
126 [0x07e]	8 bytes	Gen purpose counter	Unsigned long long	Internal usage only. Definition varies by specific system. Not for use.
134 [0x086]	4 bytes	Accumulated waveform count	Unsigned long	Used by the internal Math system. Represents the number of waveforms that have gone into the accumulation. This value is updated on a per acquisition basis. Not for use.
138 [0x08a]	4 bytes	Target accumulation count	Unsigned long	Number of acquisitions requested to be made. Not for use.
142 [0x08e]	4 bytes	Curve ref count	Unsigned long	The number of curve objects for the given waveform set. Normally 1.
146 [0x092]	4 bytes	Number of requested fast frames	Unsigned long	Number of FastFrame acquisitions that were requested for a given acquisition set.
150 [0x096]	4 bytes	Number of acquired fast frames	Unsigned long	The number of frames that the acquisition system actually acquired. This number will be less than or equal to the numRequestedFastFrames field.
154 [0x09a]	4 bytes	Pix map display format	Enum (int)	0 = DSY_FORMAT_INVALID 1 = DSY_FORMAT_YT 2 = DSY_FORMAT_XY 3 = DSY_FORMAT_XYZ
158 [0x09e]	8 bytes	Pix map max value	Unsigned long long	Max value of Pixel map.
Explicit dimension 1 and 2				
Exp dim 1 166 [0x0a6] Exp dim 2 322 [0x142]	8 bytes	Dim scale	Double	The scale of the waveform data for the given dimension. This value is used to interpret the curve data for the given axis. For the vertical (explicit) dimension, scale is used to calculate voltage. Voltage = (wfmCurveData*Scale)+offset. The scale is expressed as volts per LSB of the 16 or 32 bit digitizer values (or 1.0 for the float digitizer values or Math waveforms). Note: For the horizontal implicit dimension, scale is used to specify the sample interval (i.e.: time per point).
Exp dim 1 174 [0x0ae] Exp dim 2 330 [0x14a]	8 bytes	Dim offset	Double	The distance in units from the dimensions' zero value to the true zero value. This is the ground level offset for the explicit dimension. For the implicit dimension, offset is the trigger position.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Ext dim 1 182 [0x0b6] Ext dim 2 338 [0x152]	4 bytes	Dim size	Unsigned long	The size of the explicit dimension in terms of the base storage value. For the explicit dimension, it is the size of the storage element (e.g. 252 for 8 bit values and 65532 for 16 bit integers). Note: For the implicit dimension, size is the record length (including pre and post charge data) and is expressed in data points, not bytes of storage. Precharge points are usually 16 points and postcharge points are 16. These are used for internal display interpolation. Note: Record length computations are made using this field and the curve byte offsets.
Ext dim 1 186 [0x0ba] Ext dim 2 342 [0x156]	20 bytes	Units	Char[20]	A character string, NULL, terminated, which represents the units for the dimension.
Exp dim 1 206 [0x0ce] Exp dim 2 362 [0x16a]	8 bytes	Dim extent min	Double	The minimum attainable data value for the explicit dimension, adjusted for scale and offset. (Not for use.)
Exp dim 1 214 [0x0d6] Exp dim 2 370 [0x172]	8 bytes	Dim extent max	Double	The maximum attainable data value for the explicit dimension, adjusted for scale and offset. (Not for use.)
Exp dim 1 222 [0x0de] Exp dim 2 378 [0x17a]	8 bytes	Dim resolution	Double	For the explicit dimension, this value reflects the smallest resolution (of a voltage) possible given the products digitizer and acquisition mode, i.e. the value of one digitizing level (DL).
Exp dim 1 230 [0x0e6] Exp dim 2 386 [0x182]	8 bytes	Dim ref point	Double	For the explicit dimension, this is the ground-level reference value. Note: For the implicit dimension, this is the horizontal reference point of the time base.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Exp dim 1 238 [0x0ee] Exp dim 2 394 [0x18a]	4 bytes	Format	Enum (int) [3]	The code type of data values stored in the curve buffer. EXPLICIT_INT16 = 0, EXPLICIT_INT32 = 1, EXPLICIT_UINT32 = 2,. EXPLICIT_UINT64 = 3, EXPLICIT_FP32 = 4, EXPLICIT_FP64 = 5, EXPLICIT_INVALID_FORMAT = 6
Exp dim 1 242 [0x0f2] Exp dim 2 398 [0x18e]	4 bytes	Storage type	Enum (int) [3]	Describes the layout of the data values stored in the curve. EXPLICIT_SAMPLE = 0. EXPLICIT_MIN_MAX = 1. EXPLICIT_VERT_HIST = 2. EXPLICIT_HOR_HIST = 3. EXPLICIT_ROW_ORDER = 4. EXPLICIT_COLUMN_ORDER = 5. EXPLICIT_INVALID_STORAGE = 6.

Exp dim 1 246 [0x0f6]
Exp dim 2 402 [0x192]
	4 bytes	N value	4 bytes	The value that represents the NULL (unacquired) waveform value. This is the value that will be in the waveform data when there is no valid data available for that data element. This is not a number (NAN) for floating point values.
				The 4 bytes can be interpreted as short, long, or float depending upon the curve data type.
				union rangeValues {
				sRange INT16;
				lRange INT32;
				fRange FP32;
				};

Exp dim 1 250 [0x0fa] Exp dim 2 406 [0x196]	4 bytes	Over range	4 bytes	Special value that indicates that a point is over-ranged.
Exp dim 1 254 [0x0fe] Exp dim 2 410 [0x19a]	4 bytes	Under range	4 bytes	Special value that indicates that a point is under-ranged.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Exp dim 1 258 [0x102] Exp dim 2 414 [0x19e]	4 bytes	High range	4 bytes	The largest signed value that can be present in this data. (Except for nValue, under/over-range.)
Exp dim 1 262 [0x106] Exp dim 2 418 [0x1a2]	4 bytes	Low range	4 bytes	The smallest value that can be present in this data. (Except for nValue, under/over-range.)
		User view data for explicit dimensions 1 and 2		The User View data defines the relationship between the description of the raw data and the way the user wants to view and interact with the data. This data defines the meaning of zoom as well as the transformations that relate the data to the screen. This can include specific user modified scale and units values.
Exp dim 1 266 [0x10a] Exp dim 2 422 [0x1a6]	8 bytes	User scale	Double	User view scale expressed in terms of units/divisions and is used to apply additional scale information for display purposes.
Exp dim 1 274 [0x112] Exp dim 2 430 [0x1ae]	20 bytes	User units	Char[20]	User display units string, expressed in Units per Division. Examples are Volts per Div or Time per Div.
Exp dim 1 294 [0x126] Exp dim 2 450 [0x1c2]	8 bytes	User offset	Double	User position expressed in terms of divisions for the explicit dimension, and is used to designate the screen relative position of the waveform. Note: For implicit dimension, this value is expressed in terms of user units (typically time) and is used to designate the value of the horizontal center pixel column in relationship to the trigger, in absolute horizontal user units.
Exp dim 1 302 [0x12e] Exp dim 2 458 [0x1ca]	4 bytes	Point density [6]	Unsigned long	The relationship of screen points to waveform data. For explicit dimensions, this value is 1. Note: For implicit dimensions, this defines the number of points that are compressed into a single pixel column. This compression ratio becomes the definition of a zoom of one.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Exp dim 1 306 [0x132] Exp dim 2 462 [0x1ce]	8 bytes	HRef	Double	The horizontal position of the trigger, in units of percentage of a waveform, in trigger-centric (no delay) mode. This value is 0% - 100% of the waveform.
Exp dim 1 314 [0x13a] Exp dim 2 470 [0x1d6]	8 bytes	TrigDelay	Double	The amount of delay, in seconds, from the trigger to the HRef location. This value is positive for triggers occurring before the HRef and negative for values after the HRef.
Implicit dimensions 1 and 2				Definitions of variables are the same as for the Explicit dimension except where noted.
Imp dim 1 478 [0x1de] Imp dim 2 610 [0x262]	8 bytes	Dim scale	Double	For the horizontal implicit dimension, scale is used to specify the sample interval (for example, time per point).
Imp dim 1 486 [0x1e6] Imp dim 2 618 [0x26a]	8 bytes	Dim offset	Double	For the implicit dimension, offset is the trigger position (in a delay centric mode).
Imp dim 1 494 [0x1ee] Imp dim 2 626 [0x272]	4 bytes	Dim size	Unsigned long	Note: For the implicit dimension, size is the record length (including pre and post charge data) and is expressed in data points, not bytes of storage. Prechange points are usually 16 points and postcharge points are 16. These are used for internal display interpolation. Note: Record length computations are made off of this field and the curve byte offsets.
Imp dim 1 498 [0x1f2] Imp dim 2 630 [0x276]	20 bytes	Units	Char[20]	
Imp dim 1 518 [0x206] Imp dim 2 650 [0x28a]	8 bytes	Dim extent min	Double	Not for use.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Imp dim 1 526 [0x20e] Imp dim 2 658 [0x292]	8 bytes	Dim extent max	Double	Not for use.
Imp dim 1 534 [0x216] Imp dim 2 666 [0x298]	8 bytes	Dim resolution	Double	
Imp dim 1 542 [0x21e] Imp dim 2 674 [0x2a2]	8 bytes	Dim ref point	Double	Note: For the implicit dimension, this is the horizontal reference point of the time base.
Imp dim 1 550 [0x226] Imp dim 2 682 [0x2aa]	4 bytes	Spacing	Unsigned long	Real time point spacing (1).
		User view data for implicit dimensions 1 and 2		
Imp dim 1 554 [0x22a] Imp dim 2 686 [0x2ae]	8 bytes	User scale	Double	
Imp dim 1 562 [0x232] Imp dim 2 694 [0x2b6]	20 bytes	User units	Char[20]	
Imp dim 1 582 [0x246] Imp dim 2 714 [0x2ca]	8 bytes	User offset	Double	

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Imp dim 1 590 [0x24e] Imp dim 2 722 [0x2d2]	4 bytes	Point density [6]	Unsigned long	
Imp dim 1 594 [0x252] Imp dim 2 726 [0x2d6]	8 bytes	HRef	Double	
Imp dim 1 602 [0x25a] Imp dim 2 734 [0x2de]	8 bytes	TrigDelay	Double	
Time Base 1 and 2 information				The time base information describes how the waveform data was acquired and the meaning of the acquired points.
Time Base 1 742 [0x2e6] Time Base 2 754 [0x2f2]	4 bytes	Real point spacing	Unsigned long	This is an integer count of the difference, in points, between the acquired points. If a waveform has not been created using interpolation, then this value is one.
Time Base 1 746 [0x2ea] Time Base 2 758 [0x2f6]	4 bytes	Sweep	Enum (int) [3]	Type of acquisition SWEEP_ROLL = 0, SWEEP_SAMPLE = 1, SWEEP_ET = 2, SWEEP_INVALID = 3
Time Base 1 750 [0x2ee] Time Base 2 762 [0x2fa]	4 bytes	Type of base	Enum (int) [3]	Defines the kind of base BASE_TIME = 0, BASE_SPECTRAL_MAG = 1, BASE_SPECTRAL_PHASE = 2, BASE_INVALID = 3

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Wfm Update specification [4]				
766 [0x2fe]	4 bytes	Real point offset	Unsigned long	Integer offset from the beginning of valid data (held in the attribute preChargeStart of the Curve info) that indicates the first acquired, non-interpolated point in the record.
770 [0x302]	8 bytes	TT offset	Double	The time from the point the trigger occurred to the next data point in the waveform record. This value represents the fraction of the sample time from the trigger time stamp to the next sample.
778 [0x30a]	8 bytes	Frac sec	Double	The fraction of a second when the trigger occurred. Used in combination with gmtSec.
786 [0x312]	4 bytes	Gmt sec	Long	The time based upon gmt time that the trigger occurred to seconds, fracSec gives the trigger to a fraction of a second.
Wfm Curve information [4]				(CURVE) (Number of frames – 1)
790 [0x316]	4 bytes	State flags	Unsigned long	Internal usage flag only. Is used to indicate validity of curve buffer data. Typedef struct { unsigned flagOver : 2; unsigned flagUnder : 2; unsigned flagValid : 2; unsigned flagNulls : 2; } WfmCurveStateFlags; #define WFM_CURVEFLAG_YES 0 #define WFM_CURVEFLAG_NO 1 #define WFM_CURVEFLAG_MAYBE 2
794 [0x31a]	4 bytes	Type of check sum	Enum (int) [3]	Indicates the algorithm used to calculate the waveform checksum. Typedef enum { NO_CHECKSUM = 0, CTYPE_CRC16 = 1, CTYPE_SUM16 = 2, CTYPE_CRC32 = 3, CTYPE_SUM32 = 4 } CurveCheckType;
798 [0x31e]	2 bytes	Check sum	Short	Curve checksum. Currently not implemented. File checksum used only.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
800 [0x320]	4 bytes	Precharge start offset	Unsigned long	The byte offset from the start of the curve buffer to the first valid waveform point. This first portion of data from preChargeStart to dataStart is intended to be used by an interpolation algorithm and is not guaranteed to be available to the oscilloscope user.
804 [0x324]	4 bytes	Data start offset	Unsigned long	The byte offset from the beginning of the curve buffer to the first point of the record available to the oscilloscope user.
808 [0x328]	4 bytes	Postcharge start offset	Unsigned long	The byte offset to the point after the last user accessible waveform point. There is still valid waveform data following, but that data is intended to be used by an interpolator and is not guaranteed to be accessible to the oscilloscope user.
812 [0x32c]	4 bytes	Postcharge stop offset	Unsigned long	The byte offset to the portion of memory just past the last valid waveform point in the curve. The buffer may extend beyond this point (and often will), as described above.
816 [0x330]	4 bytes	End of curve buffer offset	Unsigned long	This value is used only for Roll mode acquisitions and is not meant for oscilloscope users. It designates a specific number of bytes that are to be allocated for the curve buffer, but not available for curve data except as defined by Roll mode. When roll is not enabled, this offset is the same as postChargeStopOffset.
FastFrame Frames				Optional
820 [0x334]	N * 24 bytes	N WfmUpdateSpec object	Variable	This is for FastFrame waveform sets and defines trigger time stamp data. N = number of frames – 1.
820+N*24	N * 30 bytes	N WfmCurveSpec objects	Variable	This is for FastFrame waveform sets, and defines curve data offsets for individual frames. N = number of frames – 1.
CurveBuffer				
820+N*(54)	Variable	Curve buffer [5]	Variable	Contains curve data (inclusive of pre/post charge) for all waveforms in set in a contiguous block (see notes below).
WfmFileChecksum				
	8 bytes	Waveform file checksum	Unsigned long long	Checksum for the waveform file. The checksum is calculated by summing all data values from the Waveform header through the Curve data as unsigned chars.

[1] Version number:

This string identifies this file as a saved waveform in the instrument format. The format of this string was chosen to make it easier to version future variations of the waveform reference file format. As noted in the table above, the first 4 bytes of the version numbers remain constant through all instrument file formats, and could potentially be used as a byte-order verification.

2 Byte offset to beginning of curve buffer:

This offset denotes where, from the start of the reference file, the curve buffer begins. The start of the curve buffer generally contains the pre-charge data, although this is better defined by the offsets contained within the reference header.

3 Enums:

Enums may be expanded in the future by adding additional members to the end of the enum, (for example, adding 8 bit data types to the end.) This can be done without changing the overall structure of the file, thus the file version number would not need to change even though the enum was expanded.

4 Per frame information:

(Wfm Update Spec and Wfm Curve Information)

The following two groups, Update Specification and Curve Information, define a single frame. When the waveform defines a set of frames, there is a set for each frame but a single curve buffer.

5 Length of curve buffer in bytes:

The format of the values within the buffer is variable (INT16, INT64 and/or FP32). The size of the curve buffer will be the number of data points within the buffer (preCharge + postCharge + recordLength) times the number of bytes in the given data format (2 for INT16, 4 for FP32, etc.). It is also important to note that the creator of the original waveform has the ability to assign a byte offset at the beginning and/or end of the file. These offsets will increase the size of the buffer beyond the standard preCharge + recordLength + postCharge size. As mentioned earlier, the waveform set curve buffer is saved as one contiguous block of data. Individual curves are defined by virtue of the local curve byte offset values.

6 Version Number:

TDS5000B Series instruments have a new waveform file format, with the following changes:

- Version Number (at Offset 0x002) is WFM#002.
- At Offset 0x9a, after the number of acquired fast frames, an unsigned short value has been added representing the summary frame type, if any. Offsets shown in the table assume that this unsigned short value is not present. This short can take on three values:
  0 = SUMMARY_FRAME_OFF,
  1 = SUMMARY_FRAME_AVERAGE, or
  2 = SUMMARY_FRAME_ENVELOPE.
- The file format then continues on with the pixmap display format.

DPO7000, DPO70000/B, DSA70000/B and MSO70000 Series instruments have a new waveform file format, with the following changes:

- Version Number (at Offset 0x002) is WFM#003.
- All TDS5000B changes except for the Version Number.
- All four point density values (at Offset 0x12e, 0x1ca, 0x24e, and 0x2d2) are doubles. Offsets shown in the table assume that these values are unsigned longs.
- Pixmaps are 1000x252x64 bits in size.
- The data-size enum (Exp dim 1 & 2, offset 0x0ee and 0x18a) has more options:
  EXPLICIT_UINT8 = 6,
  EXPLICIT_INT8 = 7,
  EXPLICIT_INVALID_FORMAT = 8

Marks

When a .wfm file is written, all user marks (not search marks) on that waveform are also saved. If a waveform has no marks, this information is not written to the file. Marks are written at the end of the file, but do not modify the header. Each mark requires about 75 bytes.

Default setup and TekSecure delete all marks (user and search marks).

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
Mark information				
0 [0x000]	8 bytes	Mark information start	Char[8]	A value which indicates mark data follows (MARKS).
8 [0x008]	4 bytes	Number of bytes to the end of the mark information	32 bit Long int	The number of mark related bytes that follow. Currently this is the end of file (EOF).
12 [0x00c]	4 bytes	Mark version number	32 bit Long int	The initial version number is 3. If later releases store more mark information, they will increase the version number. Later releases will know how to read all previous versions of marks. Mark information is not used if there is no version number, or if the checksum is bad.
16 [0x010]	4 bytes	Number of marks	32 bit Long int	The number of marks that follow (could be 0).
20 [0x014]	8 bytes	Start	64-bit double	The binary start, in waveform units, of the mark. Start through Label repeated for each mark.
28 [0x01c]	8 bytes	Focus	64-bit double	The focus point of the mark.

Offset Dec [Hex]	Size	File field	Field type	Comments/examples
36 [0x024]	8 bytes	Zoom factor	64-bit double	The zoom factor.
44 [0x02c]	8 bytes	End	64-bit double	The end of the mark.
52 [0x034]	33 bytes	Label	33 characters	Label [owner is always USER]. 32 characters plus null-termination.
				For each mark, Start Focus, Zoom factor, end, and label are repeated.
Depends on number of marks	8 bytes	Checksum	64 bits	Checksum including the character array and the mark related bytes integer (everything new after the ISDB data).
Depends on number of marks	8 bytes	Checksum	64 bits	Duplicate of checksum.