BIOPAC Systems, Inc.
Registered to ISO 9001:2000
Better solutions... Better science.
42 Aero Camino, Goleta, CA 93117
Tel (805) 685-0066 | Fax (805) 685-0067
info@biopac.com | www.biopac.com

Updated 022307

Application Note #156 Acq*Knowledge* File Format for Windows/PC

This document describes file formatting for all Windows versions of Acq*Knowledge* **3.9.0** or below, BSL 3.7.0 or below, and BSL *PRO* v3.7.0 or below. The File Format allows the user to get to the data section of the BIOPAC file format; Journal text is not included but can be exported separately if needed. Where noted, marker details (such as Type: Append or Event) may not be accessible, but the marker tag can be identified.

NOTE Programmers are strongly encouraged to use the BIOPAC File Format API for BSL or Acq*Knowledge*, which should simplify data extraction for most programmers. If the API does not satisfy your needs, additional file format info is in this document.

Document Log

Version 3.9.0	February 23, 2007 **Magenta Bold**
Version 3.8.2	July 15, 2005 **Green Bold**
Version 3.7 (BSL & *PRO*)	February 1, 2005 (markers are generic)
Version 3.8.1	November 09, 2004 Changes are in **Aqua Bold**
Acq*Knowledge* API link	April 28, 2003; Software library for programmers to parse ACQ files
Version 3.7.3	December 25, 2002; Changes are in **Orange Bold**
Version 3.7.0-3.7.2	August 21, 2001; Changes are in **Green Bold**
Version 3.7.x	June 08, 2001; Changes are in **Grayed Bold**
Version 3.5x	September 29, 1998
Version 3.03	October 4, 1995

Graph Header Section...

Item	Type	Size	Offset	Description
nItemHeaderLen	short	2	0	Not currently used.
lVersion	long	4	2	File version identifier:

30 = Pre-version 2.0
31 = Version 2.0 Beta 1
32 = Version 2.0 release
33 = Version 2.0.7 (Mac)
34 = Version 3.0 In-house Release 1
35 = Version 3.03
36 = version 3.5x (Win 95, 98, NT)
37 = version of BSL/*PRO* 3.6.x
38 = version of Acq 3.7.0-3.7.2 (Win 98, 98SE, NT, Me, 2000)
39 = version of Acq 3.7.3 or above (Win 98, 98SE, 2000, Me, XP)
41 = version of Acq 3.8.1 or above (Win 98, 98SE, 2000, Me, XP)
42 = version of BSL/*PRO* 3.7.X or above (Win 98, 98SE, 2000, Me, XP)
43 = version of Acq 3.8.2 or above (Win 98, 98SE, 2000, Me, XP)
44 = version of BSL/*PRO* 3.8.x or above
45 = version of Acq 3.9.0 or above

lExtItemHeaderLen	long	4	6	Extended item header length.
nChannels	short	2	10	Number of channels stored.
nHorizAxisType	short	2	12	Horizontal scale type, one of the

				following

0 = Time in seconds
1 = Time in HMS format
2 = Frequency
3 = Arbitrary

nCurChannel	short	2	14	Currently selected channel.
dSampleTime	double	8	16	The number of milliseconds per sample.
dTimeOffset	double	8	24	The initial time offset in milliseconds.
dTimeScale	double	8	32	The time scale in milliseconds per division.
dTimeCursor1	double	8	40	Cursor 1 time position in milliseconds.
dTimeCursor2	double	8	48	Cursor 2 time position in milliseconds.
rcWindow	RECT	8	56	The chart's size and position relative to the Acq*Knowledge* client area. When each RECT field is set to 0, the chart is displayed with default a size and position.

nMeasurement[6]	short	6*2	64	Describes the currently selected measurements, one of the following:

0 = No measurement
1 = Value Absolute voltage
2 = Delta Voltage difference
3 = Peak to peak voltage
4 = Maximum voltage
5 = Minimum voltage
6 = Mean voltage
7 = Standard deviation
8 = Integral
9 = Area
10 = Slope
11 = LinReg
13 = Median
15 = Time
16 = Delta Time
17 = Freq
18 = BPM
19 = Samples
20 = Delta Samples
21 = Time of Median
22 = Time of Max
23 = Time of Min
25 = Calculation
26 = Correlation

fHilite	BOOL	2	76	Gray non-selected waveforms:

0 = Don't gray
1 = Gray.

dFirstTimeOffset	double	8	78	Initial time offset in milliseconds.
nRescale	short	2	86	Autoscale after transforms:
				0 = Don't autoscale
				1 = Autoscale.
szHorizUnits1	char	40	88	Horizontal units text.
szHorizUnits2	char	10	128	Horizontal units text (abbreviated).
nInMemory	short	2	138	Keep data file in memory:
				0 = Keep
				1 = Don't keep.
fGrid	BOOL	2	140	Enable grid display.
fMarkers	BOOL	2	142	Enable marker display.
nPlotDraft	short	2	144	Enable draft plotting.
nDispMode	short	2	146	Display mode:
				0 = Scope
				1 = Chart.
nRReserved	short	2	148	Reserved.

Version 3.0 and above ...

Item	Type	Size	Offset	Description
BShowToolBar	short	2	150	
BShowChannelButtons	short	2	152	
BShowMeasurements	short	2	154	
BShowMarkers	short	2	156	
BShowJournal	short	2	158	
CurXChannel	short	2	160	
MmtPrecision	short	2	162	

Version 3.02 and above ...

Item	Type	Size	Offset	Description
NMeasurementRows	short	2	164	Number of measurement rows
mmt[40]	short	2 * 40	166	Measurement functions
mmtChan[40]	short	2 * 40	246	Measurement channels

Version 3.5x and above ...

Item	Type	Size	Offset	Description
MmtCalcOpnd1	short	2 * 40	326	Measurement, Calculation - Operand 1
MmtCalcOpnd2	short	2 * 40	406	Measurement, Calculation - Operand 2
MmtCalcOp	short	2 * 40	486	Measurement, Calculation - Operation
MmtCalcConstant	double	8 * 40	566	Measurement, Calculation - Constant

Version 3.7.0 and above ...

Item	Type	Offset	Size	Description
bNewGridwithMinor	BOOL	886	4	New Grid with minor line
colorMajorGrid	long	890	4	COLORREF
colorMinorGrid	long	894	4	COLORREF
wMajorGridStyle	short	898	2	PS_SOLID, PS_DASH, PS_DOT, PS_DASHDOT, PS_DASHDOTDOT
wMinorGridStyle	short	900	2	PS_SOLID, PS_DASH, PS_DOT, PS_DASHDOT, PS_DASHDOTDOT
wMajorGridWidth	short	902	2	width of line in Pixels
wMinorGridWidth	short	904	2	width of line in Pixels
bFixedUnitsDiv	BOOL	906	4	Locked/Unlocked grid lines
bMid_Range_Show	BOOL	910	4	show gridlines as MidPoint and Range
dStart_Middle_Point	double	914	8	Startpoint to draw grid
dOffset_Point	double	922	8 * 60	Offset of VERTICAL value per channel
hGrid	double	1402	8	Horizontal grid spacing
vGrid	double	1410	8 * 60	Vertical grid spacing per channel
bEnableWaveTools	BOOL	1890	4	Enable Wavetools during acquisition

Version 3.7.3 and above ...

Item	Type	Offset	Size	Description
horizPrecision	short	1894	2	digits of precision for units in Horizontal Axis

Version 3.8.1 and above ...

Item	Type	Offset	Size	Description
RESERVED	byte	1896	20	RESERVED
bOverlapMode	BOOL	1916	4	Overlap Mode
bShowHardware	BOOL	1920	4	Hardware visibility
bXAutoPlot	BOOL	1924	4	Autoplot during acquisition
bXAutoScroll	BOOL	1928	4	Autoscroll during acquisition
bStartButtonVisible	BOOL	1932	4	Start button visibility
bCompressed	BOOL	1936	4	The file is compressed
bAlwaysStartButtonVisible	BOOL	1940	4	Always show start button

Version 3.8.2 and above ...

Item	Type	Offset	Size	Description
pathVideo	char	1944	260	Path to playback video file
optSyncDelay	BOOL	2204	4	Option: use sync delay between start of video file and graph start.
syncDelay	double	2208	8	Value of sync delay (ms)
bHRP_PasteMeasurements	BOOL	2216	4	Option: paste measurements to

				journal (when Hold Relative Position is selected)

Version 3.9.0 and above ...

† 256 is a string required for the expression; 40 = 5*8 (max columns * max rows)

Item	Type	Offset	Size †	Description
graphType	DWORD	2220	4	Type of the graph. The graph type identifies the source of the graph and whether any special transformations apply.
mmtCalcExpr[40][256]	char	2224	1*256*40	Measurements parameters: holds the expression entered by the user.
mmtMomentOrder[40]	long	12464	4*40	Measurements parameters: the order of the moment for moment measurements.
mmtTimeDelay[40]	long	12624	4*40	Measurements parameters: the time delay to use for the computation in sample intervals.
mmtEmbedDim[40]	long	12784	4*40	Measurements parameters: the embedding dimension for a measurement.
mmtMIDelay[40]	long	12944	4*40	Measurements parameters: the delay for which the mutual information should be computed.

Per Channel Data Section...

Item	**Type**	**Offset**	**Size**	**Description**
lChanHeaderLen	long	0	4	Length of channel header.
nNum	short	4	2	Channel number.
szCommentText	char	6	40	Comment text.
rgbColor	RGB	46	4	Color.
nDispChan	short	50	2	Display option.
dVoltOffset	double	52	8	Amplitude offset (volts).
dVoltScale	double	60	8	Amplitude scale (volts/div).
szUnitsText	char	68	20	Units text.
lBufLength	long	88	4	Number of data samples.
dAmplScale	double	92	8	Units/count.
dAmplOffset	double	100	8	Units
nChanOrder	short	108	2	Displayed channel order.
nDispSize	short	110	2	Channel partition size.

Version 3.0 and above ...

Item	Type	Offset	Size	Description
plotMode	short	112	2	
vMid	double	114	8	

Version 3.7.0 and above ...

Item	Type	Offset	Size	Description
szDescription	char	122	128	String of Channel description
nVarSampleDivider	short	250	2	Channel divider of main frequency

Version 3.7.3 and above ...

Item	Type	Offset	Size	Description
vertPrecision	short	252	2	digits of precision for units in Vertical Axis for each channel

Version 3.8.2 and above ...

Item	Type	Offset	Size	Description
ActiveSegmentColor	COLORREF	254	4	Color of active segment (Overlapped Mode)
ActiveSegmentStyle	long	258	4	Style of active segment (Overlapped Mode)

Foreign Data Section...

Item	Type	Size	Offset	Description
nLength	short	2	0	Total length of foreign data packet.
nID	short	2	2	ID of foreign data.
ByForeignData	BYTE	nLength-4	4	Foreign data.

Per Channel Data Types Section...

This block is repeated for as many channels that were detected in the graph header packet nChannels field.

Item	Type	Size	Offset	Description
nSize	short	2	0	Channel data size in bytes.
nType	short	2	2	Channel data type:

1 = double

2 = int

Channel Data Section...

The individual channel data is stored after the Per Channel Data Types Section. The channel data is in an interleaved format.

The start of the real Channel Data is described by the following equation:

[lExtItemHeaderLen] + [lChanHeaderLen * nChannels] + [nlength] + [4 * nChannels]

The size of the Graph header section is defined in lExtItemHeaderLen (for Acq 3.7x this is 1894 bytes) if you take this value it will give you a pointer to the **Per Channel Data** section...

- The **Per Channel Data** section size is defined by lChanHeaderLen for each channel. Use the nChannels variable from the Graph header section and multiply by lChanHeaderLen (252 bytes for Acq 3.7x), add to the value from above this will then give you a pointer to the Foreign Data section.

- The **Foreign Data** section length is defined by nLength and will vary depending on your acquisition setup. Add this to your running sum to get a pointer to the Per Channel Data Types section.
- The **Per Channel Data Types** section is where you find out if your data is double or integer and 2 bytes (Analog) or 8 bytes (Calculation channel) again this is repeated for each channel so multiply nChannels by 4 to get the total length of this section and add to your running sum to get a pointer to the real data.

A note of caution here, since we have added the variable sampling ability this could make the interleaved data format very complex to understand. For this reason, you will want to be sure all channels are at the same rate and length in order to export the data correctly. The data is stored in an interleaved format.

For 3 channels it would look something like:

Ch1 sample1, Ch2 sample1, Ch3 sample1, Ch1s2, Ch2s2, ch3s2, Ch1s3, Ch2s3, Ch3s3, etc.

If channels are sampled at different rates rather than having 3 channels of data for each sample it will skip the channel that has no value at that sample. For 3 channels, Channel 2 is half the sample rate it would look like:

Ch1 sample1, Ch2 sample1, Ch3 sample1, Ch1s2, ch3s2, Ch1s3, Ch2s3, Ch3s3, etc.

With Ch2s2 missing you can see how this would create disruption if the reading program is expecting a sample for each waveform.

Markers Header Section...

Item	Type	Size	Offset	Description
lLength	long	4	0	Total length of all markers.
lMarkers	long	4	4	Number of markers.

Marker Item Section...

Item	Type	Size	Offset	Description
lSample	long	4	0	Location of marker.
fSelected	BOOL	2	4	Select this marker.
fTextLocked	BOOL	2	6	Lock this text.
fPositionLocked	BOOL	2	8	Lock this location.
nTextLength	short	2	10	Length of marker text (including NULL).
szText	char	nTextLength	12	Marker text string.

Hardware structure...

For MP100 and MP150 (added adaptive filter channel settings); 16 is the number of calculation channels.

Item	Type	Size	Description
adaptiveFilter_valueChan	short	2*16	Value (source) channel number: this is the data consisting of the source data mixed with noise.
adaptiveFilter_noiseChan	short	2*16	Noise channel number: it is correlated with the noise that is desired to be eliminated from the source channel.
adaptiveFilter_order	long	4*16	The number of terms that are used in the internal FIR filter.
adaptiveFilter_stepSize	double	8*16	The rate of adaptation of the coefficients within the FIR filter.

BIOPAC File Format API for BSL or Acq*Knowledge* Files

The **BIOPAC File Format Application Programming Interface (API)** is a software library that people with programming knowledge can use to identify and parse information in BIOPAC's ACQ binary file format.

- The API is a software library programmers can use to create applications—it is not a data extraction program.

The API was developed using Microsoft Visual Studio C/C++ and is designed to be compatible with Microsoft C++ and Microsoft Visual Basic. The API may work with other developer's tools, but has not been tested for compatibility.

The API is the easiest way for people with programming knowledge to extract data from ACQ files for advanced analysis.

The API can be used for *.ACQ files created by BSL Lessons, BSL *PRO* or Acq*Knowledge*. It works with acquisitions of a single sample rate saved in the standard (not compressed) ACQ file format. The API can be used for files created on PCs with Windows (98, 98SE, Me, 2000 and XP) or Macintosh files that have been transferred to a PC running Windows.

The Acq*Knowledge* API allows you to:

- Initialize an ACQ file structure
- Close an ACQ file structure
- Retrieve channel information
- Retrieve samples by segment of a specified channel
- Retrieve all the samples of a specified channel
- Retrieve a particular sample of a specified channel
- Retrieve samples by time slice of a specified channel
- Retrieve Journal Text for Acq*Knowledge* 3.7.3 or below
- Retrieve marker information
- Retrieve marker text of a specified marker

Use these base functions in a variety of combinations to extract specific data from an ACQ file and then use or transform the data in other analysis programs.

The API includes:
acqfile.dll Dynamic link library*
acqfile.lib COFF import library for acqfile.dll
acqfile.h Header file
sample.acq A small .acq file to use for testing

* To run executables generated by the sample projects, a copy of acqfile.dll must exist in the same directory as the executable.

Directories
docs html documentation for the *acq* File Format
cplusplusample sample VC++ 6.0 project folder
vbsample sample VB 6.0 project folder

Open **acqfile_8h.htm** from the "docs" folder for the Main Page of the documentation and navigate as desired from there, or start with a sample file for C++ or VB projects.