BIOPAC Systems, Inc.

42 Aero Camino
Goleta, CA 93117
Ph (805) 685-0066
Fax (805) 685-0067
www.biopac.com
info@biopac.com

#AS155 - Acq*Knowledge* File Formats for the Macintosh

Following is the Acq*Knowledge* file format for the Macintosh platform. This document describes file formatting for Acq*Knowledge* versions 3.0-3.7.3. This File Format document does not apply to version 3.8.x.

- This document has been formatted for printing and is best viewed as a print out.

```
// BIOPAC Systems Acq Knowledge file format      V3.0                    Nov 20, 1993

// All short doubles are 64 bit IEEE floating point form

// int, & Word are 16 bit, Motorola format (msb first)

// Bool, char & Byte are 8 bits

// Main header

#define nMtMax  6

unsigned int    mainHeaderLen;          // Not used

long            version;                        // File version number:

                                        // 30 = pre-version 2.0, 32 = v 2.0, 33 = v 2.0.7,
35 = v3.0

long            extItemHeaderLen;       // Item header length

int             nChans;                 // Number of channels stored (max = 60)

int             horizAxisType;          // 0 = time, 1 = Freq, 2 = HH:MM:SS, 3=Arbitrary

int             curChannel;             // currently selected channel


short double    sampleTime;             // Sample interval             (msec/sample)

short double    tOffset;                        // time offset                 (msecs)
```

```
short double    tScale;                 // time scale                   (msecs/div)

short double    tCursor1;               // Cursor 1 time position       (msec)

short double    tCursor2;               // Cursor 2 time position       (msec)


Rect            windRect;               // Window rectangle

int             mmt[nMtMax];            // Measurement functions(1st menu = 1), (pre-
version 3.0 only)

Bool            hilite;                 // Gray non-selected waves

char            dummy1;                 // Not used

short double    firstTOffset;           // Global time offset (msec)

int             rescale;                // 0=none,1=Autoscale,2=tile; after modify

char            szHorizUnits1[40];      // Horizontal units text (long)

char            szHorizUnits2[10];      // Horizontal units (short)

int             inMemory;               // If non-zero, Keep data file in memory


Bool            grid;                   // Enable grid display

Bool            markers;                        // Enable marker display

int             plotDraft;              // Enable draft mode plotting

Bool            dispMode;               // 0 = scope mode, 1 = chart mode

char            dummy2;                 // Not used

int             overWritePrompt;        // Not used


// v3.0

int             bShowToolBar;           // Tool bar visible if TRUE

int             bShowChannelButtons;    // Channel numbers visible if TRUE

int             bShowMeasurements;      // Measurements visible if TRUE

int             bShowMarkers;           // Markers visible if TRUE

int             curXChannel;            // Current horizontal axis channel
```

```
int             mmtPrecision;           // Number of digits after decimal point
int             nMeasurementRows;       // Number of measurement rows
int             mmt[40];                // Measurement functions
int             mmtChan[40];            // Measurement channels

// The following structure repeats for nChans as specified above
{
        long            chanHeaderLen;  // Length of channel header
        int             chanNum;        // This channel's number
        char            szComTxt[40];   // Waveform comment text (label)
        long            waveColor;      // waveform color
        int             dispChan;       // 1=invisible, 2 = visible

        short double    vOffset;                // ampl offset (units)
        short double    vScale;         // ampl scale (units/div)

        char            szUnitsTxt[20]; // ampl units text
        long            bufLength;      // samples
        short double    amplScale;      // (units/count) for integer conversion
        short double    amplOffset;     // (units) for integer conversion

        int             chanOrder;      // Channel display order (1 is top)
        int             dispSize;       // Channel area display width
        RGBColor        newWaveColor;   // RGB waveform color value
        short           plotMode;       // 0 = scope, 1 = chart, 2 = X-Y
        double          vMid;           // Midpoint of waveform extents (units)
}
```

```
// End of main header

// Creator specific header


int     creatorHeaderLen;               // Creator header length

int     creatorHeaderType;              // 0x0100 = AcqKnowledge

char    creatorData[creatorHeaderLen-4];// Creator specific data


// End of creator specific header


typedef struct  {       int     dSize;  // Data size (bytes)

                        int     dType;  // Data type

                                        // 1 :  Floating point

                                        //              dSize is precision (Bytes)

                                        //              ( 4, 8, 12, ... )

                                        // 2 :  Binary

                                        //              dSize is word length

                } binSizeType;


// Data header

binSizeType     dataHead[nChans];       // Data size/types

// Actual data

char            data[]                  // Data is interleaved (lowest number channel first)


{

        long            length;         // Total marker storage length

        long            nMarkers;       // Number of markers stored
```

```
// The following structure repeats for the number of markers as specified above (nMarkers)

        {
                long            sample;
                Bool            selected;
                Bool            textLocked;
                Bool            posLocked;
                char            dummy3;                 // Not used
                int             textLen;
                char            markerText[textLen];    // Marker text string
        }
}
```