CLIO SOFTWARE RELEASE 4.0: DATA FILES STRUCTURE

CONTENTS

REV 1.0 - 23 Dec 1997

Introduction

This articles covers the data files of CLIO software version 4.0.
The variable types, native in Borland Pascal, can be summarized as follows:

Byte	:	unsigned 8-bit
Word	:	unsigned 16-bit
Integer	:	signed 16-bit
Longint	:	signed 32-bit
Single	:	floating point (single precision IEEE 754/854 standard)
Boolean	:	8-bit ordinal 0 or 1
Char	:	8-bit ASCII character
String	:	sequence of 8-bit ASCII characters preceded by its 8-bit size attribute

FFT files (".FFT" extension)

```
WrHead = Record
  Nome       : String[11];      {12 bytes — usually ''AUDIOMATICA"}
  Programma  : String[8];       {9 bytes - usually ''CLIO"}
  Release    : String[4];       {5 bytes - software release}
  Comm1      : String[40];      {41 bytes - comment}
  Comm2      : String[40];      {41 bytes - comment}
  Comm3      : String[40];      {41 bytes - comment}
  Comm4      : String[106];     {107 bytes - comment}
end;
FFTText = Record
  Titolo     : String[8];       {9 bytes – file name}
```

```
  Commento  : String[50];          {51 bytes – user input comment (with About)}
end;
FFTLocals = Record
  Fcamp     : Word;                {2 bytes – sampling frequency}
  Mode      : MisUnit;             {1 byte – measuring mode (Volt, Pressure)}
  MicASens  : Single;              {4 bytes – A channel sensitivity}
  MicBSens  : Single;              {4 bytes – B channel sensitivity}
  SensUnits : UnitsType;           {1 byte – sensitivity units (mV/Pa, dBV/Pa, dBspl/V)}
  MicCal    : Boolean;             {1 byte – microphone calibration switch}
  FullScale : Single;              {4 bytes – top of screen value}
  Channel   : AcquiMode;           {1 byte – input channel (A,B,A-B)}
end;
FFTSet = Record
  NumPoints  : integer;            {2 bytes – FFT size}
  Nbit       : integer;            {2 bytes – FFT order}
  NumAcqui   : integer;            {2 bytes – number of input samples}
  Average    : boolean;            {1 byte – average mode switch}
  NuAvgStop  : word;               {2 bytes – number of target averages}
  NumAverage: integer;             {2 bytes – number of averages performed}
  Window     : FFTWindowType;      {1 byte - type of window
(Hanning,Hamming,Blackman,Bartlett)}
  WindowState: boolean;            {1 byte – window mode switch}
  IntTrg     : boolean;            {1 byte – internal trigger switch}
  IntTrgDelay : integer;           {2 bytes – internal trigger delay}
  Off1       : boolean;            {1 byte – display 1 off switch}
  Off2       : boolean;            {1 byte - display 2 off switch}
  Graph1     = Record
    What     : ChannelOn;          {1 byte – display 1 channel (A,B,A-B,A/B)}
    Ykind    : FFTYAxisKind;       {1 byte – display 1 function (Mag,Real,Imag)}
    Yaxis    : FFTYAxis;           {1 byte – display 1 y axis mode (linear,dB)}
    Xaxis    : FFTXAxis;           {1 byte – display 1 x axis mode (linear,logarithmic)}
  end;
  Graph2     = Record
    What     : ChannelOn;          {1 byte – display 2 channel (A,B,A-B,A/B)}
    Ykind    : FFTYAxisKind;       {1 byte – display 2 function (Mag,Real,Imag)}
    Yaxis    : FFTYAxis;           {1 byte – display 2 y axis mode (linear,dB)}
    Xaxis    : FFTXAxis;           {1 byte – display 2 x axis mode (linear,logarithmic)}
  end;
  Disp1      = Record
    Fscale   : Single;             {4 bytes – display 1 full scale value}
    FscaleI  : Integer;            {2 bytes – display 1 full scale index}
    IncScale : Single;             {4 bytes – display 1 scale increment value}
    IncI     : Integer;            {2 bytes – display 1 scale increment index}
    Span     : Single;             {4 bytes – display 1 total span value}
  end;
  Disp1      = Record
    Fscale   : Single;             {4 bytes – display 2 full scale value}
    FscaleI  : Integer;            {2 bytes – display 2 full scale index}
    IncScale : Single;             {4 bytes – display 2 scale increment value}
    IncI     : Integer;            {2 bytes – display 2 scale increment index}
    Span     : Single;             {4 bytes – display 2 total span value}
  end;
```

```
 MaxIndex    : integer;                  {2 bytes – last FFT value index}
end;
FFTOut.A = Array [0..FFTSet.MaxIndex] Of    {(4*FFTSet.Maxindex) bytes – FFT channel A
data}
        Complex = Record
          Re    : Single;               {4 bytes – real part}
          Im    : Single;               {4 bytes – imag part}
        end;
FFTOut.B = Array [0..FFTSet.MaxIndex] Of    {(4*FFTSet.Maxindex) bytes – FFT channel B
data}
        Complex = Record
          Re    : Single;               {4 bytes – real part}
          Im    : Single;               {4 bytes – imag part}
        end;
```

** The following part is present only in averaged FFTs.

```
NumAverage : integer;                   {2 bytes – number of averages performed}
FFTVal.1.Data =Array [0..FFTSet.MaxIndex] Of      {(4*FFTSet.Maxindex) bytes – FFT chan-
nel A average data}
        FFTData = Record
          Inst  : single;               {4 bytes – instantaneous value}
          Avg   : single;               {4 bytes – averaged value}
        end;
FFTVal.2.Data =Array [0..FFTSet.MaxIndex] Of      {(4*FFTSet.Maxindex) bytes – FFT chan-
nel B average data}
        FFTData = Record
          Inst  : single;               {4 bytes – instantaneous value}
          Avg   : single;               {4 bytes – averaged value}
        end;
```

Total bytes count = variable.

MLS files (“.MLS” extension)

```
WrHead = Record
 Nome        : String[11];              {12 bytes — usually ‘’AUDIOMATICA”}
 Programma   : String[8];               {9 bytes - usually ‘’CLIO”}
 Release     : String[4];               {5 bytes - software release}
 Comm1       : String[40];              {41 bytes - comment}
 Comm2       : String[40];              {41 bytes - comment}
 Comm3       : String[40];              {41 bytes - comment}
 Comm4       : String[106];             {107 bytes - comment}
end;
MLSText = Record
 Titolo      : String[8];               {9 bytes – file name}
 Commento    : String[50];              {51 bytes – user input comment (with About)}
end;
MLSLocals = Record
 Fcamp       : Word;                    {2 bytes – sampling frequency}
 Mode        : MisUnit;                 {1 byte – measuring mode (Volt, Pressure)}
```

```
 MicASens   : Single;              {4 bytes – A channel sensitivity}
 MicBSens   : Single;              {4 bytes – B channel sensitivity}
 SensUnits  : UnitsType;           {1 byte – sensitivity units (mV/Pa, dBV/Pa, dBspl/V)}
 MicCal     : Boolean;             {1 byte – microphone calibration switch}
 FullScale  : Single;              {4 bytes – top of screen value}
 Channel    : AcquiMode;           {1 byte – input channel (A,B,A-B)}
end;
MLSSet = Record
 dBRange    : byte                 {1 byte – amplitude scale range (20,10,5,2,1 dB/div)}
 FreqStart  : byte                 {1 byte – x-axis start frequency (20,200,2000 Hz)}
 Smooth     : byte                 {1 byte – smoothing factor (no,1/2,1/3,1/6,1/12 octave)}
 PhKind     : byte                 {1 byte – kind of phase data (Normal, Minimum,
DelayFree)}
 NuAvg      : integer              {2 bytes – number of averages}
 AvgMode    : byte                 {1 byte – kind of average mode (Continuous, Manual)}
 TimeW      : byte                 {1 byte – time window (no, HalfHann, Hann, HalfBH,
BH)}
 BeginPoint : integer;             {2 bytes – first sample of selected impulse}
 EndPoint   : integer;             {2 bytes – last sample of selected impulse}
end;
MLSSinglePulse= array [0..16382] of single;    {65532 bytes – impulse data}
***            = array [0..2] of byte;         {3 bytes – reserved}
MLSXReal   = array [0..4095] of single;{16384 bytes – FFT real part}
MLSXImag   = array [0..4095] of single;{16384 bytes – FFT imag part}
```

Total bytes count = 98649 bytes.

Waterfall files (“.WTF” extension)

```
WrHead = Record
 Nome       : String[11];          {12 bytes — usually ‘’AUDIOMATICA”}
 Programma  : String[8];           {9 bytes - usually ‘’CLIO”}
 Release    : String[4];           {5 bytes - software release}
 Comm1      : String[40];          {41 bytes - comment}
 Comm2      : String[40];          {41 bytes - comment}
 Comm3      : String[40];          {41 bytes - comment}
 Comm4      : String[106];         {107 bytes - comment}
end;
WTFText = Record
 Titolo     : String[8];           {9 bytes – file name}
 Commento   : String[50];          {51 bytes – user comment (input with About)}
end;
WTFSet = Record
 Tot_Spec   : integer;             {2 bytes – total number of spectra}
 End_Spec   : integer;             {2 bytes – last transform sample}
 FrStart    : integer;             {2 bytes – start frequency}
 Wtf_d      : integer;             {2 bytes – no care}
 SmoothState: boolean;             {1 byte – smoothing switch}
 Wtf_f      : boolean;             {1 byte – no care}
end;
WTFSave    = array [0..399,0..30] of byte;    {12400 bytes – waterfall data}
```

Total bytes count = 12726 bytes.

Sinusoidal Frequency Response files (“.FRS” extension)

```
WrHead = Record
  Nome        : String[11];          {12 bytes — usually ‘’AUDIOMATICA”}
  Programma   : String[8];           {9 bytes - usually ‘’CLIO”}
  Release     : String[4];           {5 bytes - software release}
  Comm1       : String[40];          {41 bytes - comment}
  Comm2       : String[40];          {41 bytes - comment}
  Comm3       : String[40];          {41 bytes - comment}
  Comm4       : String[106];         {107 bytes - comment}
end;
FRSText = Record
  Titolo      : String[8];           {9 bytes – file name}
  Commento    : String[50];          {51 bytes – user comment (input with About)}
end;
FRSLocals = Record
  Fcamp       : Word;                {2 bytes – sampling frequency}
  Mode        : MisUnit;             {1 byte – measuring mode (Volt, Pressure)}
  MicASens    : Single;              {4 bytes – A channel sensitivity}
  MicBSens    : Single;              {4 bytes – B channel sensitivity}
  SensUnits   : UnitsType;           {1 byte – sensitivity units (mV/Pa, dBV/Pa, dBspl/V)}
  MicCal      : Boolean;             {1 byte – microphone calibration switch}
  FullScale   : Single;              {4 bytes – top of screen value}
  Channel     : AcquiMode;           {1 byte – input channel (A,B,A-B)}
end;
FRSSet = Record
  dBRange     : byte                 {1 byte – amplitude scale range (20,10,5,2,1 dB/div)}
  FrsFrRge    : FreqRange            {1 byte – frequency scale range
                                     (10-20K,100-20K,1K-20K,10-2K,10-200 Hz)}
  FrsFreqRes  : FreqRes              {1 byte – frequency resolution (1/3, 1/6, 1/12, 1/24, 1/
48 octave)}
  StartF      : Single               {4 bytes – first measurement frequency}
  StopF       : Single               {4 bytes – last measurement frequency}
  FrsSpeed    : Speed                {1 byte – sweep speed (fast, mid, slow)}
  Gtd         : boolean              {1 byte – switch for gated acquisition}
  GtdAutoPh   : boolean              {1 byte – switch for auto phase}
  GtdAutoPhFreq: Single              {4 bytes – reference frequency for gated acquisition}
  GtdAutoMtDel: boolean              {1 byte – switch for auto delay}
  GtdMtDel    : Single               {4 bytes – delay for gated acquisition}
  GtdMtOn     : Single               {4 bytes – sampling time for gated acquisition}
  THD         : boolean              {1 byte – switch for harmonic analysis}
  THD2Dysp    : boolean              {1 byte – switch for second harmonic analysis}
  THD3Dysp    : boolean              {1 byte – switch for third harmonic analysis}
  THDRiseVal  : boolean              {4 bytes – dB rise for harmonic analysis}
end;
FrsArray      = array [0..535] of SinStruct;      {6432 bytes – measurement data}
        SinStruct = Record
          Val   : Complex = Record       {value}
```

```
            Re    : Single;          {4 bytes – real part}
            Im    : Single;          {4 bytes – imag part}
          end;
    Freq : Single                    {4 bytes – frequency}
  End;
```

** The following part is present only when harmonic analysis is enabled.

```
THD2Array   = array [0..535] of SinStruct;     {6432 bytes – second harmonic data}
    SinStruct = Record
     Val   : Complex = Record        {value}
            Re    : Single;          {4 bytes – real part}
            Im    : Single;          {4 bytes – imag part}
          end;
     Freq : Single                   {4 bytes – frequency}
    End;
THD3Array   = array [0..535] of SinStruct;     {6432 bytes – third harmonic data}
    SinStruct = Record
     Val   : Complex = Record        {value}
            Re    : Single;          {4 bytes – real part}
            Im    : Single;          {4 bytes – imag part}
          end;
     Freq : Single                   {4 bytes – frequency}
    End;
```

Total bytes count = variable.

Sinusoidal Impedance files (“.IMP” extension)

```
WrHead = Record
 Nome        : String[11];          {12 bytes — usually ‘’AUDIOMATICA”}
 Programma : String[8];             {9 bytes - usually ‘’CLIO”}
 Release     : String[4];           {5 bytes - software release}
 Comm1       : String[40];          {41 bytes - comment}
 Comm2       : String[40];          {41 bytes - comment}
 Comm3       : String[40];          {41 bytes - comment}
 Comm4       : String[106];         {107 bytes - comment}
end;
IMPText = Record
 Titolo      : String[8];           {9 bytes – file name}
 Commento  : String[50];            {51 bytes – user comment (input with About)}
end;
IMPSet = Record
 OhmMax     : Single                {4 bytes – maximum graph value}
 OhmMin     : Single                {4 bytes – minimum graph value}
 LinLogY    : byte                  {1 byte – kind of Y axis (linear, logarithmic)}
 Auto        : boolean              {1 byte – switch for auto display parameters}
 IMPFrRge   : FreqRange             {1 byte – frequency scale range
                                    (10-20K,100-20K,1K-20K,10-2K,10-200 Hz)}
 IMPFreqRes: FreqRes                {1 byte – frequency resolution (1/3, 1/6, 1/12, 1/24, 1/
48 octave)}
```

```
  StartF       : Single                  {4 bytes – first measurement frequency}
  StopF        : Single                  {4 bytes – last measurement frequency}
  FrsSpeed     : Speed                   {1 byte – sweep speed (fast, mid, slow)}
  Mode         : ImpMode                 {1 byte – measurement mode (internal, constant I, constant V)}
  ResVal       : Single                  {4 bytes – sensing resistor value}
end;
IMPArray      = array [0..535] of SinStruct;      {6432 bytes – impedance data}
        SinStruct = Record
          Val   : Complex = Record       {value}
                    Re    : Single;      {4 bytes – real part}
                    Im    : Single;      {4 bytes – imag part}
                  end;
          Freq  : Single                 {4 bytes – frequency}
        End;
```

Total bytes count = 6774 bytes.

Sinusoidal Parameters files (“.SML” extension)

```
WrHead = Record
  Nome         : String[11];             {12 bytes — usually ‘’AUDIOMATICA”}
  Programma    : String[8];              {9 bytes - usually ‘’CLIO”}
  Release      : String[4];              {5 bytes - software release}
  Comm1        : String[40];             {41 bytes - comment}
  Comm2        : String[40];             {41 bytes - comment}
  Comm3        : String[40];             {41 bytes - comment}
  Comm4        : String[106];            {107 bytes - comment}
end;
SMLText = Record
  Titolo       : String[8];              {9 bytes – file name}
  Commento     : String[50];             {51 bytes – user comment (input with About)}
end;
Parameters   : Record
  Manufacturer: String[20]               {21 bytes – Manufacturer’s name}
  Model        : String[20]              {21 bytes – Model’s name}
  ***          : String[11]              {12 bytes – reserved}
  Fs           : Single                  {4 bytes – resonance frequency of driver}
  FsAdMa       : Single                  {4 bytes – resonance frequency with added mass}
  FsKnVl       : Single                  {4 bytes – resonance frequency with known volume}
  AdMass       : Single                  {4 bytes – added mass}
  KnVol        : Single                  {4 bytes – known volume}
  D            : Single                  {4 bytes – diameter}
  Zm           : Single                  {4 bytes – maximum impedance of driver}
  ***          : Single                  {4 bytes – reserved}
  ***          : Single                  {4 bytes – reserved}
  ZF1F2        : Single                  {4 bytes – impedance of driver at –3 dB frequencies}
  F1           : Single                  {4 bytes – lower frequency at –3 dB}
  F2           : Single                  {4 bytes – upper frequency at –3 dB}
  Re           : Single                  {4 bytes – DC resistance of voice coil}
  Rms          : Single                  {4 bytes – mechanical resistance of driver suspension}
```

```
  Qms          : Single                    {4 bytes}
  Qes          : Single                    {4 bytes}
  Qts          : Single                    {4 bytes}
  Cms          : Single                    {4 bytes – mechanical compliance of suspension}
  Mms          : Single                    {4 bytes – mechanical mass of diaphragm}
  Bl           : Single                    {4 bytes – magnetic flux density}
  Vas          : Single                    {4 bytes – volume of air with same compliance of sus-
pension}
  dBspl        : Single                    {4 bytes – calculated at 1m with 2.83V}
  L1K          : Single                    {4 bytes – inductance at 1 KHz}
  L10K         : Single                    {4 bytes – inductance at 10 KHz}
  Cas          : Single                    {4 bytes – acoustical compliance of suspension}
  ***          : Single                    {4 bytes – reserved}
  ***          : Single                    {4 bytes – reserved}
  ***          : Single                    {4 bytes – reserved}
  SD           : Single                    {4 bytes – effective area of diaphragm}
  ***          : array[0..10] of Single    {44 bytes – reserved}
End;
ParamIMPArray= array [0..535] of SinStruct;   {6432 bytes – impedance data}
        SinStruct = Record
          Val   : Complex = Record        {value}
                    Re    : Single;       {4 bytes – real part}
                    Im    : Single;       {4 bytes – imag part}
                  end;
          Freq : Single                   {4 bytes – frequency}
        End;

Total bytes count =  6962 bytes.
```

Sinusoidal Distortion files (“.DST” extension)

```
WrHead = Record
  Nome         : String[11];              {12 bytes — usually ‘’AUDIOMATICA”}
  Programma    : String[8];               {9 bytes - usually ‘’CLIO”}
  Release      : String[4];               {5 bytes - software release}
  Comm1        : String[40];              {41 bytes - comment}
  Comm2        : String[40];              {41 bytes - comment}
  Comm3        : String[40];              {41 bytes - comment}
  Comm4        : String[106];             {107 bytes - comment}
end;
DSTText = Record
  Titolo       : String[8];               {9 bytes – file name}
  Commento     : String[50];              {51 bytes – user comment (input with About)}
end;
DSTData = Record
  THDFreq      : Single                   {4 bytes – THD frequency}
  CCIFFreq     : Single                   {4 bytes – CCIF frequency}
  Start        : Single                   {4 bytes – Start voltage}
  Stop         : Single                   {4 bytes – Stop voltage}
  ***          : Single                   {4 bytes – reserved}
  ***          : Single                   {4 bytes – reserved}
```

```
  Load          : Single                    {4 bytes – Load resistance}
  Steps         : Single                    {4 bytes – Number of steps}
end;DSTUnit     : DSTUnitType               {1 byte – measurement unit (volts, watts)}
DSTKind         : DSTKindType               {1 byte – measurement kind (THD, SMPTE, DIN,
CCIF)}
DSTArray        = array [0..400] of DSTVal;{3208 bytes – impedance data}
          DSTVal = Record
            Lev   : Single;               {4 bytes – level}
            Dist  : Single                {4 bytes – distortion}
          End;
```

Total bytes count = 3558 bytes.

Polar files (“.POL” extension)

```
WrHead = Record
  Nome          : String[11];             {12 bytes — usually ‘’AUDIOMATICA”}
  Programma     : String[8];              {9 bytes - usually ‘’CLIO”}
  Release       : String[4];              {5 bytes - software release}
  Comm1         : String[40];             {41 bytes - comment}
  Comm2         : String[40];             {41 bytes - comment}
  Comm3         : String[40];             {41 bytes - comment}
  Comm4         : String[106];            {107 bytes - comment}
end;
POLText = Record
  Titolo        : String[8];              {9 bytes – file name}
  Commento      : String[50];             {51 bytes – user comment (input with About)}
End;
RTALocals = Record
  Fcamp         : Word;                   {2 bytes – sampling frequency}
  Mode          : MisUnit;                {1 byte – measuring mode (Volt, Pressure)}
  MicASens      : Single;                 {4 bytes – A channel sensitivity}
  MicBSens      : Single;                 {4 bytes – B channel sensitivity}
  SensUnits     : UnitsType;              {1 byte – sensitivity units (mV/Pa, dBV/Pa, dBspl/V)}
  MicCal        : Boolean;                {1 byte – microphone calibration switch}
  FullScale     : Single;                 {4 bytes – top of screen value}
  Channel       : AcquiMode;              {1 byte – input channel (A,B,A-B)}
end;
PolStruct = Record
  Freq          : Single;                 {4 bytes – measurement frequency (Hz)}
  Step          : Single;                 {4 bytes – measurement angle step (degrees)}
  Data          : array [0..71] of Single {288 bytes – polar data (dB)}
End;
```

Total bytes count = 630 bytes.

RTA files (“.PNK” extension in 3.21, “.RTA” extension in 4.00)

```
WrHead = Record
  Nome        : String[11];                 {12 bytes — usually ‘’AUDIOMATICA”}
  Programma   : String[8];                  {9 bytes - usually ‘’CLIO”}
  Release     : String[4];                  {5 bytes - software release}
  Comm1       : String[40];                 {41 bytes - comment}
  Comm2       : String[40];                 {41 bytes - comment}
  Comm3       : String[40];                 {41 bytes - comment}
  Comm4       : String[106];                {107 bytes - comment}
end;
RTAText = Record
  Titolo      : String[8];                  {9 bytes – file name}
  Commento    : String[50];                 {51 bytes – user comment (input with About)}
End;
RTALocals = Record
  Fcamp       : Word;                       {2 bytes – sampling frequency}
  Mode        : MisUnit;                    {1 byte – measuring mode (Volt, Pressure)}
  MicASens    : Single;                     {4 bytes – A channel sensitivity}
  MicBSens    : Single;                     {4 bytes – B channel sensitivity}
  SensUnits   : UnitsType;                  {1 byte – sensitivity units (mV/Pa, dBV/Pa, dBspl/V)}
  MicCal      : Boolean;                    {1 byte – microphone calibration switch}
  FullScale   : Single;                     {4 bytes – top of screen value}
  Channel     : AcquiMode;                  {1 byte – input channel (A,B,A-B)}
end;
RTANuAver     : integer;                    {2 bytes – number of averages}
RTAAvgMod     = array [0..30] of single;    {124 bytes – RTA data}
RTAAvgTotMod= single;                       {4 bytes – total level dB (linear)}
RTAAAvgTotMod= single;                      {4 bytes – total level dBA (A-weighted)}
```

Total bytes count = 468 bytes.

RT60 files (“.T60” extension)

```
WrHead = Record
  Nome        : String[11];                 {12 bytes — usually ‘’AUDIOMATICA”}
  Programma   : String[8];                  {9 bytes - usually ‘’CLIO”}
  Release     : String[4];                  {5 bytes - software release}
  Comm1       : String[40];                 {41 bytes - comment}
  Comm2       : String[40];                 {41 bytes - comment}
  Comm3       : String[40];                 {41 bytes - comment}
  Comm4       : String[106];                {107 bytes - comment}
end;
T60Text = Record
  Titolo      : String[8];                  {9 bytes – file name}
  Commento    : String[50];                 {51 bytes – user comment (input with About)}
End;
T60Locals = Record
  Fcamp       : Word;                       {2 bytes – sampling frequency}
  Mode        : MisUnit;                    {1 byte – measuring mode (Volt, Pressure)}
```

```
  MicASens    : Single;                 {4 bytes – A channel sensitivity}
  MicBSens    : Single;                 {4 bytes – B channel sensitivity}
  SensUnits   : UnitsType;              {1 byte – sensitivity units (mV/Pa, dBV/Pa, dBspl/V)}
  MicCal      : Boolean;                {1 byte – microphone calibration switch}
  FullScale   : Single;                 {4 bytes – top of screen value}
  Channel     : AcquiMode;              {1 byte – input channel (A,B,A-B)}
end;
T60SingState= array [0..7] of boolean   {8 bytes – switch active for each measured octave}
```

** For each measured octave the following data is appended

```
T60SinglePulse= array [0..16382] of single;{65532 bytes – measured octave impulse data}
T60NoCare   = array [0..2] of byte;     {3 bytes – no care data}
T60FileFCamp        : integer;          {2 bytes – measured octave sampling frequency}
```

Total bytes count = variable.

Leq files (“.LEQ” extension)

```
LEQData     : array[0..Samples] of single       {4*Samples bytes – time history data}
WrHead = Record
  Nome        : String[11];             {12 bytes — usually ‘’AUDIOMATICA”}
  Programma   : String[8];              {9 bytes - usually ‘’CLIO”}
  Release     : String[4];              {5 bytes - software release}
  Comm1       : String[40];             {41 bytes - comment}
  Comm2       : String[40];             {41 bytes - comment}
  Comm3       : String[40];             {41 bytes - comment}
  Comm4       : String[106];            {107 bytes - comment}
end;
LEQText = Record
  Titolo      : String[8];              {9 bytes – file name}
  Commento    : String[50];             {51 bytes – user comment (input with About)}
End;
LEQLocals = Record
  Fcamp       : Word;                   {2 bytes – sampling frequency}
  Mode        : MisUnit;                {1 byte – measuring mode (Volt, Pressure)}
  MicASens    : Single;                 {4 bytes – A channel sensitivity}
  MicBSens    : Single;                 {4 bytes – B channel sensitivity}
  SensUnits   : UnitsType;              {1 byte – sensitivity units (mV/Pa, dBV/Pa, dBspl/V)}
  MicCal      : Boolean;                {1 byte – microphone calibration switch}
  FullScale   : Single;                 {4 bytes – top of screen value}
  Channel     : AcquiMode;              {1 byte – input channel (A,B,A-B)}
end;
LEQSet = Record
  Mode        : IntegrationType;        {1 byte – kind of integration (slow, fast, impulse)}
  TimeUnit    : Char;                   {1 byte – ‘s’ per second or ‘m’ per minute}
  dB          : dBType;                 {1 bytes – equal dB or dBA)
  UnitPerDiv  : integer;                {2 bytes – number of units per time division}
  StopTime    : longint;                {4 bytes – integration stop time in s}
  CountPerSec  : integer;               {2 bytes – equal 8 (fast or slow) or 32 (impulse)}
End;
```

```
LEQValue    : Single                    {4 bytes – final Leq value}
LEQMax      : Single                    {4 bytes – maximum level}
LEQOvl      : Boolean                   {1 bytes – switch active when overload occurred during
measure}
```

Total bytes count = variable.

Oscilloscope files (“.SPE” extension)

```
WrHead = Record
  Nome        : String[11];             {12 bytes — usually ‘’AUDIOMATICA”}
  Programma   : String[8];              {9 bytes - usually ‘’CLIO”}
  Release     : String[4];              {5 bytes - software release}
  Comm1       : String[40];             {41 bytes - comment}
  Comm2       : String[40];             {41 bytes - comment}
  Comm3       : String[40];             {41 bytes - comment}
  Comm4       : String[106];            {107 bytes - comment}
end;
ScopeText = Record
  Titolo      : String[8];              {9 bytes – file name}
  Commento    : String[50];             {51 bytes – user input comment (with About)}
end;
Soglia        : integer;                {2 bytes – trigger level index}
AmpIndex      : integer;                {2 bytes – vertical amplification index}
TBIndex       : integer;                {2 bytes – time base index}
ScopeCh       : ChannelOn;              {1 byte – display mode (A,B,A-B,Dual)}
IntTrig       : boolean;                {1 byte – internal trigger switch}
IntTrigDly    : integer;                {2 bytes – internal trigger delay}
ScoS          = array [0..510] of integer; {1022 bytes – channel a data}
ScoSb         = array [0..510] of integer; {1022 bytes – channel b data}
```

Total bytes count = 2370 bytes.

‘CONV3TO4.EXE’ Translation software from release 3.21 to 4.00

The program CONV3TO4.EXE can be used to translate CLIO data files in the old 3.21 format to a format compatible with release 4.00. To avoid wrong results the following precautions have to be taken.

NOTE 1: The conversion program has to be saved in and run from the C:\CLIO40\ directory.

NOTE 2: Before running the conversion program perform the standard CLIO 4 system calibration.

The program directs the user with simple prompts while doing its job.