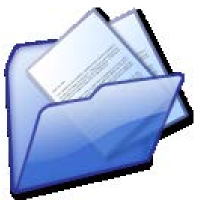
Document No.: 98-0451
Release Date: September 28, 2012
Revision: Issue-4

GUIDE TO THE ANALECT SPECTRAL FILE FORMAT



Applied Instrument Technologies, Inc.
2121 Aviation Dr
Upland, CA 91786
Phone: (909)204-3700

DOC. NO.: 98-0451

DOCUMENT RELEASE RECORD (DRR)

RELEASES (& DATES)	Update this document by: ACTIVE PAGES	PAGES ADDED	PAGES DELETED	AUTHORIZATIONS (& REMARKS)
Issue-1 (4/20/2004)	All.	All.	None.	Initial release.
Issue-2 (5/4/2006)	All.	ii	None.	SPR#1280 Added document release record as page ii. Updated document information and indices throughout. Corrected document number in Appendix and Index. Updated division name on cover page and in section 1.1. Added RPM product line reference in section 1.2. Corrected figure 1 alignment and caption in section 2. Corrected abbreviation definition on first use throughout. Added new section 5.
Issue-3 (6/20/2006)	All.	None.	None.	SPR#1290 Corrected headers in Appendix & Index, updated field codes and issue number.
Issue-4 (9/28/2012)	All	None	None	SPR #122 Company Name and Logo updated on front cover page.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Purpose

The purpose of this document is to serve as an overview of the Analect Spectral File (ASF) format used by the Applied Instrument Technologies (AIT), Upland, California. This document is identified as the Guide to the Analect Spectral File Format, document number 98-0451.

1.2 Scope

The Analect spectral file (ASF) format is the primary data file format used for the Analect and RPM product lines. Providing a clear and concise description of the file format allows for quicker and more reliable internal and external software development. This guide describes the overall structure of the Analect spectral file format, and details the individual elements and definitions used within the file header and various descriptors.

1.3 Definitions, Acronyms, and Abbreviations

Please refer to Appendix A for clarification of terminology used in this document.

2 Basic Structure

The basic structure of an Analect spectral file consists of a number of descriptor blocks that indicate the type, size, and location of a corresponding file component. In addition, each descriptor block in a file contains the location of the next descriptor block and thus the next component. The following diagram illustrates the overall structure of an Analect spectral file.

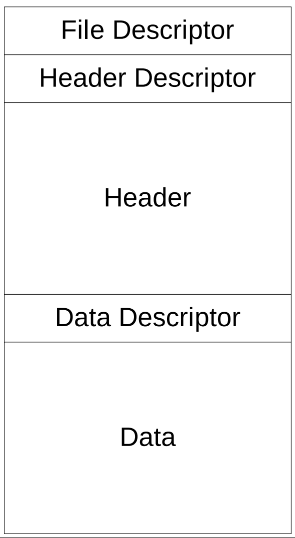


Figure 1 – Analect spectral file block diagram

3 Descriptors

A descriptor is a 16 byte block that indicates a components type, size, and location in the file. The structure of a desciptor block is as follows:

```
typedef struct
{
  long ld;      // link down, position of next descriptor
  long la;      // link across, unused at this time
  long size;    // size of corresponding component, including descriptor
  int version;// version of ASF file structure
  char ctype; // component type
  char ftype; // file type
} DESCRIPTOR;
```

The component type is defined as follows:

```
#define     UNDEFINED     0
#define     TRACEDATA     1
#define     TRACEHEADER   2
#define     PEAKTABLE     3
#define     COMMENT       4
#define     CMDHISTORY    5
#define     AFHEADER      6
```

The file type is defined as follows:

```
#define     UNDEFINED     0
#define     TRACEFILE     1
#define     GCFILE        2
#define     PARMFILE      3
#define     DATABASEFILE  4
```

4 ASF Headers

An ASF header is an 898 byte block containing information about the collection of the ASF data that follows later in the file. The structure of the ASF header is as follows:

```
typedef struct
{
  long
    time,               // time elapsed in seconds since 00:00:00 Jan 01 1970.
    serial_no,          // serial number
    ndata,              // # of trace data points
    ig_size,            // # of points in original interferogram
    fft_size,           // # of points in interferogram fast Fourier transform (FFT)
    fft_spin,           // # of points skipped at beginning of transform result
    scans_sig,          // # of signal scans
    scans_bkg,          // # of background scans
    _L_FILL[6];         // room for additional longs in the future
  float
    xleft,              // x-axis leftmost value
    xright,             // x-axis rightmost value
    yorg,               // y-axis default origin for display (center of full scale)
    ymax,               // y-axis default maximum for display
    yscale,             // y-axis scale factor for integer format trace functions
    ig_step,            // retardation step for points in original interferogram
    resolution,         // spectral resolution in wave numbers(WN)
    mol_wt,             // molecular weight in atomic units (A.U.)
    bp,                 // boiling point (degrees Celsius)
    mp,                 // melting point (degrees Celsius)
    xdelta,             // x-axis delta between points, = (xright-xleft)/(ndata-1)
    laserwn,            // laser wave number for Raman instruments
    _F_FILL[2];         // room for additional floats in the future
  int
    lgain_sig,          // 2*log2(signal gain)
    lgain_bkg,          // 2*log2(background gain)
    phig_len,           // size of phase interferogram if Mertz method
                        // phase-corrected spectrum, or Foreman method
                        // phase-corrected interferogram
    ver_num,            // header structure version number
    transept,           // flag word for transept inteferometer data,
                        // bit 0 set if non-linear spectrum or igram
    pc_flags,           // phase correct flag word, see definitions
    _I_FILL[6];         // room for additional integers in the future
  TFMT_TYPE trace_fmt;  // trace format, see definitions
  DFMT_TYPE data_fmt;   // trace data format, see definitions
  XAXIS_TYPE xaxis;     // x-axis units, see definitions
  YAXIS_TYPE yaxis;     // y-axis units, see definitions
  BSPLT_TYPE bs_type;   // beamsplitter and wedge type, see definitions
  APOD_TYPE ap_type;    // apodization type, see definitions
  int _E_FILL[2];       // room for additional enumerations in the future
  char
    title[60],          // trace title for display and plot
    desc1 [60],         // trace description lines, usually brief
    desc2 [60],         // sample description if spectrum trace
    mfgr [24],          // instrument manufacturer
    model [24],         // instrument model
    origin [60],        // trace origin
    owner [60],         // trace (copyright) owner
    ioperator [60],     // instrument operator
    casnumber [16],     // Chemical Abstracts Servise (CAS) registry number
    casname [60],       // CAS name
    mol_form [60],      // molecular formula
    wws [32],           // Wiswesser index
    xunits [8],         // x-axis display units
```

```
        yunits [8],          // y-axis display units
        detector [16],       // detector type
        int_type [16],       // interferometer type
        ap_comm [26],        // apodization type annotation
        _C_FILL[96];         // room for additional characters in the future
    } ASFHEADER;
```

5 Header Versions and Raman Redefinitions

In order to maintain compatibility with the existing header structure, several of the data items defined in the ASFHEADER structure shown in section 4 are redefined for Raman systems.

The ASF header contains an internal header structure version number in the integer “ver_num”. The current header structure version is 3.10, and is defined in “ver_num” as the integer value 310. Any spectra with header version numbers less than 310 are considered to contain Fourier transform infrared (FTIR) data.

Any spectra with header version numbers greater than or equal to 310 may contain either FTIR or Raman data. The floating point value “laserwn” is new with version 3.10, and is used to determine if the data is from an FTIR or Raman system. Header structures prior to version 3.10 were the same size, but the number of spare floats was 3 instead of 2, and there was no “laserwn” floating point value.

If the “ver_num” is greater than or equal to 310 and the “laserwn” value is between 9,400 and 50,000 inclusive, then the data is considered to be Raman. Otherwise, the data is considered to be FTIR.

Since the display of FTIR data items follows the header structure parameters, the items redefined for Raman systems are shown in the following table.

Table 1 - Raman redefinitions of ASF header parameters

ASF Header Parameter	Raman Meaning
Title	Acquisition information
Desc1	Comment
Desc2	X correction information
ScansSig	Exposures co-added
Wws	Exposure period, milliseconds (ms)
Igstep	ASF point spacing, inverse centimeters (cm-1)
Fftsize	Grating period, line pairs (lp)/millimeter (mm)
Molwt	Grating blaze, nanometers (nm)
Mp	Camera Temp, degrees Celsius
Bp	Camera Temp locked 0 or 1
Inttype	Spectrograph serial number
Laserwn (was _F_FILL[0])	Laser wavenumber

The “title” entry in Raman spectral file headers, as shown in Table 1, is filled with acquisition information. The acquisition information is further broken out as follows:

“S=” The number of the strip that produced the data.

“AQ=” The name of the acquire parameter file that directed the acquisition and calibration of the data.

“F=” A nine character code detailing the corrections performed upon the data defined as follows:

1st: Dark correct: N = none, F = File based, A = Automatic.

2nd: X correction performed on this spectrum true or false (T or F).

3rd: X correction data obtained from this spectrum T or F.

4th: Y correction performed T or F.

5th-9th: X correction points N or 0 or 1. For this spectrum the neon correction list is examined if a point is defined then a 0 or 1 is entered. If no point is defined, a N is entered.

“%F=” The maximum signal in percent of full scale.

An example of a Raman title entry is:

S=3 AQ=N1S_30Z F=FTTT11111 %F=24.2%

The “desc2” entry in Raman spectral file headers, as shown in Table 1, is filled with X correction information. The X correction information is further broken out as follows:

“RA=” Average offset between known and measured Raman references.

“LO=” Laser offset correction from AQP file.

“A0=” Constant term of quadratic correction equation.

“A1=” Proportional term of quadratic correction equation.

“A2=” Square term of quadratic correction equation.

6 Various Type Definitions

The trace format is defined as follows:

```
typedef enum TFMT_TAG
{
  TFMT_UNK,     // unknown or undefined
  TFMT_SPC,     // signal spectrum (ratioed)
  TFMT_BKG,     // background or single-beam spectrum
  TFMT_IGM,     // interferogram trace
  TFMT_TIM,     // time function trace
  TFMT_CGM,     // chromatogram,
                // gas chromatogram (GC)
                // routine gas chromatogram (RGC)
                // liquid chromatogram (LC), etc.
  TFMT_ARB      // arbitrary trace function
} TFMT_TYPE;
```

The trace data format is defined as follows:

```
typedef enum DFMT_TAG
{
  DFMT_UNK,     // unknown or undefined (no data)
  DFMT_INT2,    // 2-byte (16-bit) signed binary integer data
  DFMT_INT4,    // 4-byte (32-bit) signed binary integer data
  DFMT_INT8,    // 8-byte (64-bit) signed binary integer data
  DFMT_FLT4,    // 4-byte (32-bit) IEEE floating point data
  DFMT_FLT8,    // 8-byte (64-bit) IEEE floating point data
} DFMT_TYPE;
```

The x-axis units is defined as follows:

```
typedef enum XAXIS_TAG
{
  XAX_UNK,      // unknown or undefined x units
  XAX_WN,       // trace function of frequency, normally in wavenumbers (cm-1)
  XAX_MICR,     // trace function of wavelength, normally in microns
  XAX_TIME,     // trace function of time
  XAX_ARB       // trace function of arbitrary x units
} XAXIS_TYPE;
```

The y-axis units is defined as follows:

```
typedef enum YAXIS_TAG
{
  YAX_UNK,      // unknown or undefined y units
  YAX_TR,       // transmittance
  YAX_AB,       // absorbance (or kumu)
  YAX_PAS,      // photoacoustic trace
  YAX_ARB       // arbitrary y-axis units
} YAXIS_TYPE;
```

The beamsplitter and transept wedge composition are defined as follows:

```
typedef enum BSPLT_TAG
{
  BSPLT_UNK,    // unknown beamsplitter type
  BSPLT_KBR,    // potassium bromide (KBr)
  BSPLT_CSI,    // cesium iodide (CsI)
  BSPLT_CAF,    // calcium fluoride/fluorite (CaF2)
  BSPLT_MYLAR,  // mylar [(r) DuPont]
```

```
    BSPLT_NACL, // sodium chloride (NaCl)
    BSPLT_BAF   // barium fluoride (BaF2)
} BSPLT_TYPE;
```

The apodization is defined as follows:

```
typedef enum APOD_TAG
{
  APDT_UNK,     // unknown or not applicable
  APDT_BOX,     // Boxcar (no apodization)
  APDT_NBWEAK,  // Norton-Beer (weak)
  APDT_NBMED,   // Norton-Beer (medium)
  APDT_NBSTR,   // Norton-Beer (strong)
  APDT_GAUSS,   // Gaussian
  APDT_HG,      // Happ-Genzel
  APDT_TRAP,    // Trapezoidal
  APDT_TRI,     // Triangular
  APDT_TRI2,    // Triangular squared
  APDT_BESS,    // Bessel
  APDT_COS,     // Cosine
  APDT_SINC2,   // Sinc squared
  APDT_BH3T,    // 3 term Blackmann-Harris
  APDT_BH4T     // 4 term Blackmann-Harris
} APOD_TYPE;
```

The bit definitions for the transept flag word are as follows:

```
#define TSI_NONLIN   0x0001 // File is nonlinear (interferogram or spectrum)
#define TSI_TSI      0x0002 // File taken on an Analect Transept class
                            // interferometer.
```

The bit definitions for the phase correct flag word are as follows:

```
#define PCF_MSHFT    0      // Bits 0-3 are method
#define PCF_MMASK    0x000F
#define PCM_SSMERTZ  0      // 0 = Single-sided Mertz Method
#define PCM_SSMAG    1      // 1 = Single-sided Magnitude Method
#define PCM_DSMAG    2      // 2 = Double-sided Magnitude Method
#define PCM_SSFOR    3      // 3 = Single-sided Forman Method
#define PCF_TSHFT    4      // Bits 4-7 are truncation type
#define PCF_TMASK    0x00F0 // (Reserved for future implementation)
#define PCT_RAMP     0      // 0 = Ramp, over width of phase igram
#define PCT_RECT     1      // 1 = Rectangular truncation at
                            // zero path difference (ZPD)
```

Appendixes

Appendix A - Definitions, Acronyms, and Abbreviations

A.U.	Atomic units.
AIT	Applied Instrument Technologies.
ASF	Analect spectral file.
BaF2	Barium fluoride.
CaF2	Calcium fluoride.
CAS	Chemical abstracts service.
cm-1	Inverse centimeters.
CsI	Cesium iodide.
FFT	Fast Fourier transform.
FTIR	Fourier transform infrared.
GC	Gas chromatogram.
SLS	Space, Land & Sea.
KBr	Potassium bromide.
LC	Liquid chromatogram.
lp	Line pairs.
ms	Milliseconds.
NaCl	Sodium chloride.
nm	Nanometers.
RGC	Routine gas chromatogram.
WN	Wavenumber.
ZPD	Zero path difference.

Index