Title :EMMFF V. 1.0
Keywords :XEDS,EELS,AES,WDS,CLS,GAM,XRF,PES
Computer :IBM, MAC, DEC
Operating System :ALL
Programming Language :Fortran 77
Hardware Requirements :None
Author(s) :EMSA/MAS TASK FORCE
Ray Egerton ,Charles E. Fiori ,John A. Hunt,
Mike S. Isaacson,Earl J. Kirkland ,Nestor J. Zaluzec
Correspondence Address :R.F. EGERTON-CHAIRMAN
University of Alberta
Dept. of Physics
Edmonton, Alberta, Canada, T6G2J1
Abstract:

A simple format for the exchange of digital spectral data is presented, and proposed as an EMSA/MAS standard. This format is readable by both humans and computers and is suitable for transmission through various electronic networks (BITNET, ARPANET), the phone system (with modems) or on physical computer storage devices (such as floppy disks). The format is not tied to any one computer, programming language or computer operating system. The adoption of a standard format would enable different laboratories to freely exchange spectral data, and would help to standarize data analysis software. If equipment manufacturers were to support a common format, the microscopy and microanalysis community would avoid duplicated effort in writing data-analysis software. This version of EMSAMASFF contains two subroutines which read and write spectral data files Version 1.0 data format. The data are stored as simple ASCII characters at a user defined number of columns per line for the length of the data file. The spectral data is preceeded by a series of header lines, which tell the user about the parameters of the spectrum. The header lines are identified by the first character in the line being the symbol (#) followed by a descriptor and if appropriate its units. An example of a data file format can be found in the EMSAMASFF.DOC file.

Title :EMMFF V. 1.0
Keywords :XEDS,EELS,AES,WDS,CLS,GAM,XRF,PES
Computer :IBM, MAC, DEC
Operating System :ALL
Programming Language :Fortran 77
Hardware Requirements :None
Author(s) :EMSA/MAS TASK FORCE
Ray Egerton ,Charles E. Fiori ,John A. Hunt,
Mike S. Isaacson,Earl J. Kirkland ,Nestor J. Zaluzec
Correspondence Address :R.F. EGERTON-CHAIRMAN
University of Alberta
Dept. of Physics
Edmonton, Alberta, Canada, T6G2J1
Documentation:

EMSA/MAS STANDARD FILE FORMAT FOR SPECTRAL DATA EXCHANGE

EMSA/MAS Task Force:

Ray Egerton - Chairman
University of Alberta

Department of Physics
Alberta, Canada,

Charles E. Fiori
Department of Chemistry
National Institute for Science and Technology
Gaithersburg, MD 20899

John A. Hunt
Department of Materials Science
Lehigh University
Bethlehem, PA 18015-3195

Mike S. Isaacson, Earl J. Kirkland
Department of Applied Physics
212 Clark Hall
Cornell University
Ithaca, NY 14853

Nestor J. Zaluzec
Materials Science Division -212
Electron Microscopy Center for Materials Research
Argonne National Laboratory
Argonne, IL 60439

Implementation Date October 1, 1991

Abstract

A simple format for the exchange of digital spectral data is presented, and has been designated as an EMSA/MAS standard. This format is readable by both humans and computers and is suitable for transmission through various electronic networks (BITNET, ARPANET), the phone system (with modems) or on physical computer storage devices (such as floppy disks). The format is not tied to any one computer, programming language or computer operating system. The adoption of a standard format would enable different laboratories to freely exchange spectral data, and would help to standarize data analysis software. If equipment manufacturers were to support a common format, the microscopy and microanalysis community would avoid duplicated effort in writing data-analysis software.

1.INTRODUCTION

The virtues of a single standard data format have been admirably related by various authors [1], [2], [3], [4]. It would often be convenient, after visiting another laboratory to use a different type of microanalytical spectrometer, to be able to return to one's own laboratory to analyze the data, or for a laboratory to be able to send a spectrum to another group at another location for analysis on their computer. A common format would also enable test spectra to be transported between data acquisition systems, in order to compare different data-analysis routines, and would give users greater choice of analysis procedure, based on commercial or public-domain software.

Obviously, an ideal solution would be for the manufacturers to represent data in a standard format, but they are unlikely to agree on this without some direction from their customers (the microanalysis community). Therefore it is highly desirable for EMSA and MAS to

proceed with adoption of a standard format. Such a format does not preclude any research group or manufacturer from having their own, possibly proprietary, format. Spectral data can be stored internally in any format, as long as there is an option to convert it to the external standard (and vice versa) for the purposes of exchange. We believe that a standard format should poses the following attributes.

1.It should be capable of representing the data exactly (without altering the scientific content).

2.The format should be simple and easy to use.

3.It must NOT be tied to any particular computer, programming language or operating system. It should work on a large number of computers of all sizes, although we cannot guarantee that it will work on all possible computers.

4.The format should be both human and machine (computer) readable.

5.It should be compatible with existing electronic communication networks such as (but not necessarily limited to) BITNET, ARPANET, and with the phone system (using modems). Future networks will likely retain compatibility with these.

6.The format should support spectra of interest to the EMSA/MAS community (such as XEDS, EELS, AES, etc.) and should be flexible enough to accommodate future data sets, not yet specified.

7.Each file should contain enough information to uniquely identify the type and origin of the spectral data and to reconstruct its significance.

8.Where possible, the format should be compatible with various commercial data plotting or analysis programs (i.e. spreadsheets, or graphical analysis packages).

9.The proposed format need not be the most efficient storage mechanism. Its primary goals, stated above, will generally prevent storage efficiency. If anything, this format will err on the side of simplicity and ease of use.

The format originally employed by the Electron Microscopy and Microanalysis Public Domain Library (EMMPDL) at Argonne [1] has the virtue of simplicity, but is too rigid for general use. A recent revision [5] corrected some inadequacies, but a more serious reexamination is in order. The format proposed by a previous EMSA Task Force [3,4] addresses many of the problems but is thought by some microscopists to be too complicated for everyday use. The VMAS format, whose description [6] runs to 60 pages, is also too complex for our perceived purpose. A format, named JCAMP-DX, used by the infrared spectroscopy community [2] is specific and detailed but is somewhat off-target for the spectroscopies of interest to our community.

The format proposed here follows JCAMP-DX in many ways, but is less complicated and has features tailored to x-ray, energy-loss and Auger spectroscopies. We circulated a preliminary version of this proposal to several manufacturers of XEDS systems and have received back comments and suggestions, many of which have been incorporated into this document.

The companion problem of a standard format for digital image storage is

similar to that of spectral data, but is sufficiently different to warrant its own standard. Whereas most spectra are sufficiently compact that they can be stored in a human-readable form, image data are usually so extensive as to require storage of 'raw' binary numbers. There exist formats for image storage that are in widespread use. One of these (probably TIFF) should be endorsed by EMSA and MAS, allowing the microanalysis community to take advantage of the large amount of commercial and public-domain software already available.

2.FORMAT DESCRIPTION

The general structure of the data file format can be summarized as a simple sequential ASCII (text) file. It begins with a series of header lines which serve to define the characteristics of the spectrum. These header lines are identified by unique keyword fields which occupy the first 15 positions of each line, followed by a data field. These are described in detail below. After the header lines, a keyword indicates the start of data, the data then follows on successive lines in a manner which is defined explicitly within the header. Finally after all the data is presented, an end of data keyword indicates the data set is complete. This is diagramatically illustrated below.

Header Lines
*
*
Successive lines begining with EMSA/MAS defined keywords
Some of which are required and some are optional
*
*
Start of Data Keyword
*
*
Experimental Data
*
*
End of Data Keyword

In general, each line of the file either contains a keyword and its associated value or spectral data. All header lines are readily identified as they each begin with '#'in the first character field or column. This symbol demarks the start of a 13 character keyword field, providing descriptive information about the data followed by an associated value. EMSA/MAS defined keywords (whose definition may be changed only by EMSA/MAS) begin with a single # and occupy the first 13 columns (characters) of each header line. The keyword itself consists of at most twelve characters which directly follow the #, shorter keywords may be employed and any remaining spaces following the defined keyword may be filled with descriptive text such as unit designation for ease of legibility when the file is printed (see examples in Tables 1 and 2). If a position in the keyword field is not used it must be filled with a space character. The keywords are not case-sensitive, so that #Xunits is interpreted the same as #XUNITS.

The 14th and 15th character positions (columns) in each header line are occupied by keyword/value field separators, which consist of a colon followed by a space. The value associated with each keyword starts in column 16 and maybe either textual or numeric as defined by the keyword. Each line of the file may contain no more than 79 characters (for

compatibility with the largest number of computers and computer networks, and for general legibility on standard width CRT screens). Since the keyword and its seperator occupies the first 15 positions, this means that all remaining information following the keyword is limited to a maximum of 64 (=79-15) character positions. The end of line indicator is a carriage return, linefeed combination (<CR><LF>).

The only characters allowed in the file are the space (ASCII character 32), carriage return (ASCII character 13) and linefeed (ASCII character 10) characters, plus the printable ASCII character set given below:

!"#$%&'()*+,-./0123456789:;<=>? @ (ASCII characters 33 - 64)
ABCDEFGHIJKLMNOPQRSTUVWXYZ (ASCII characters 65 - 90)
abcdefghijklmnopqrstuvwxyz (ASCII characters 97 - 122)
[\] ` {|}~ (ASCII characters 91 - 96,ASCII 123-126)

Horizontal TAB characters are NOT permitted in this file format as a substitute for spaces or commas. Examples of keywords, seperators and data can be found in Tables 1 and 2, and are further detailed below.

2.1 REQUIRED KEYWORDS

The following keywords are required and must appear at the beginning of the file, in the order specified below. Although some of these may appear arbitrary, it is our judgment that they fulfil a long-term need. After several years of students and outside users passing through a laboratory, the result can be a vast number of data files of unknown origin. Unless some adequate form of labelling is imposed from the start, many worthwhile data files are lost and useless data sits on a disk taking up valuable space. With the following minimum subset of keywords, it should be possible to reconstruct the significance of most spectra.

Note that there must be exactly one of each required keyword, except for the keyword #TITLE which must appear at least once but may at the users decision appear more than once to provide an extended length title.

Required Keywords

#FORMAT = Character string identifies this format as "EMSA/MAS Spectral Data File"
#VERSION = File Format Version Number (1.0 for this implementation)
#TITLE = Gives a short description of the spectra One or more per file. Max = 64 Characters
#DATE = The calendar day-month-year in which the spectra was recorded, in the form: DD-MMM-YYYY.
#TIME = The time of day at which the spectrum was recorded,in 24-hour format: HH:MM.
#OWNER = The name of the person who recorded the spectrum.
#NPOINTS = Total Number of Data Points in X&Y Data Arrays
1. < NPOINTS < 4096.
#NCOLUMNS = Number of columns of data
1. < NCOLUMNS < 5. when DATATYPE = Y
1. < NCOLUMNS < 3. when DATATYPE = XY
#XUNITS = Units for x-axis data, for example: eV.
#YUNITS = Units for y-axis data, for example: counts.

#DATATYPE = Method in which the data values are stored as
Y Axis only values or X,Y data pairs. The
current options are the characters Y and XY.
#XPERCHAN = The number of x-axis units per channel.
#OFFSET = A real (but possibly negative) number
representing value of channel one in xunits
#SPECTRUM = Indicates the next line starts the spectral data
#ENDOFDATA = Indicates the end of the data file

2.2 SPECTRAL DATA

The spectral data must be enclosed between the following keywords:

#SPECTRUM : Signifies the beginning of spectral data (on the
next line).

#ENDOFDATA : Signifies the end of spectral data.

Between these keywords, the spectrum is listed in one of two ways, as specified by the #DATATYPE value.

In the case of spectra with equally-spaced x-values (equal increments per channel), up to #NCOLUMNS y-values may be given per line. For Y-axis datatype 1 < #NCOLUMNS < 5. Each y-value is either a real number (containing a decimal point, even if there is no fractional component) or is expressed in exponential form (e.g. 3.142E+3), and is followed immediately by a comma. The corresponding x-values can be obtained from the specified values of #XPERCHAN and #OFFSET or #CHOFFSET.

If the user prefers, data may be stored as (x,y) pairs of data points, at #NCOLUMNS per line. For XY-axis datatype 1 < #NCOLUMNS < 3. The x- and y-values are expressed as real numbers or in exponential format, and are separated with a comma. The (x,y) pairs themselves are separated by a comma followed by at least one blank space.

The reason for avoiding the use of integer numbers is that in some instances decimal values are generated, for example if energy- loss spectra have been scaled for normalization or to remove a gain change. If necessary the decimal point and the comma can be removed with a text editor (using a global replace) to give integer values, whereas the reverse process may not be straightforward.

Numbers less than unity can be represented either with or without a zero preceding the decimal point (e.g. 0.1 or .1). In the case of negative numbers, there should be no spaces between the minus sign and the numerical value. We recommend that there should be no trailing spaces after a number (preceding the comma). The (x,y) option has been included to accommodate segmented spectra, containing gaps where y-data are not specified, and to allow for the possibility that the x-axis scale is nonlinear. In addition, it makes the data compatible with most general-purpose graph-plotting software packages. In some future version of the format, this option could be extended by the addition of x and y spatial coordinates, to allow for x-ray maps or energy-selected images, but more a compact representation based on a TIFF standard might be more attractive.

2.3 OPTIONAL KEYWORDS

All optional keywords except #CHECKSUM must appear before the keyword #SPECTRUM and after #CHOFFSET. If used, the check-sum value should be

the last line in the file and is the integer summation of all characters in the file, excluding the last line containing the checksum itself, and excluding any trailing spaces after the line terminator. This option is provided to test the integrity of data transmission and/or storage.

With one exception, the keywords listed below require the user to specify an associated value as a real number (with a decimal point and not more than 20 characters in length, including the decimal point) or as an ASCII character string < 64 characters long. The exception is #CHECKSUM, which requires a signed integer value. For ease of classification we have grouped the optional keywords in order of their function, thus keywords dealing with the spectrum, type of spectroscopy, specimen are presented together in the following listing. This grouping is not required within the file format, however, it is strongly recommended. In the keyword list below the abbreviation [RN] means the keyword is a real number while [nCS] indicates a character string of n characters.

Keywords relating mainly to Spectrum Characteristics

#SIGNALTYPE = Type of Spectroscopy, allowed values are [3CS]:
EDS = Energy Dispersive Spectroscopy
WDS = Wavelength Dispersive Spectroscopy
ELS = Energy Loss Spectroscopy
AES = Auger Electron Spectroscopy
PES = Photo Electron Spectroscopy
XRF = X-ray Fluorescence Spectroscopy
CLS = Cathodoluminescence Spectroscopy
GAM = Gamma Ray Spectroscopy
#XUNITS = X-Axis Data units [64CS]
#YUNITS = Y-Axis Data units [64CS]
#XLABEL = X-Axis Data label [64CS]
#YLABEL = Y-Axis Data label [64CS]
#CHOFFSET = A real (but possibly negative) number representing the channel number whose value corresonds to zero units on the x-axis scale.
#COMMENT = Comment line [64CS]
The comment keyword may be repeated as often as desired within header lines of the file.

Keywords relating mainly to Microscope/Instrument

#BEAMKV = Accelerating Voltage of Instrument in kilovolts [RN]
#EMISSION = Gun Emission current in microAmps [RN]
#PROBECUR = Probe current in nanoAmps [RN]
#BEAMDIAM = Diameter of incident probe in nanometers [RN]
#MAGCAM = Magnification or Camera Length [RN]
Mag in x or times, Cl in mm
#CONVANGLE = Convergence semi-angle of incident beam in milliRadians [RN]
#OPERMODE = Operating Mode, allowed values are [5CS]:
IMAGE = Imaging Mode
DIFFR = Diffraction Mode
SCIMG = Scanning Imaging Mode
SCDIF = Scanning Diffraction Mode

Keywords relating mainly to Specimen

#THICKNESS = Specimen thickness in nanometers [RN]
#XTILTSTGE = Specimen stage tilt X-axis in degrees [RN]
#YTILTSTGE = Specimen stage tilt Y-axis in degrees [RN]
#XPOSITION = Specimen/Beam position along the X axis [RN]
#YPOSITION = Specimen/Beam position along the Y axis [RN]
#ZPOSITION = Specimen/Beam position along the Z axis [RN]

Keywords relating mainly to ELS

#DWELLTIME = Dwell time/channel for serial data
collection in msec [RN]
#INTEGTIME = Integration time per spectrum for parallel data
collection in milliseconds [RN]
#COLLANGLE = Collection semi-angle of scattered beam in mR [RN]
#ELSDET = Type of ELS Detector, allowed values are [6CS]:
SERIAL = Serial ELS Detector
PARALL = Parallel ELS Detector

Keywords relating mainly to EDS

#ELEVANGLE = Elevation angle of EDS,WDS detector in degrees [RN]
#AZIMANGLE = Azimuthal angle of EDS,WDS detector in degrees [RN]
#SOLIDANGLE = Collection solid angle of detector in sR [RN]
#LIVETIME = Signal Processor Active (Live) time in seconds [RN]
#REALTIME = Total clock time used to record the spectrum in
seconds [RN]
#TBEWIND = Thickness of Be Window on detector in cm [RN]
#TAUWIND = Thickness of Au Window/Electrical Contact in cm [RN]
#TDEADLYR = Thickness of Dead Layer in cm [RN]
#TACTLYR = Thickness of Active Layer in cm [RN]
#TALWIND = Thickness of Aluminium Window in cm [RN]
#TPYWIND = Thickness of Pyrolene Window in cm [RN]
#TBNWIND = Thickness of Boron-Nitride Window in cm [RN]
#TDIWIND = Thickness of Diamond Window in cm [RN]
#THCWIND = Thickness of HydroCarbon Window in cm [RN]
#EDSDET = Type of X-ray Detector, Allowed values are [5CS]:
SIBEW = Si(Li) with Be Window
SIUTW = Si(Li) with Ultra Thin Window
SIWLS = Si(Li) Windowless
GEBEW = Ge with Be Window
GEUTW = Ge with Ultra Thin Window
GEWLS = Ge Windowless

#CHECKSUM = 32 BIT INTEGER NUMBER

The #CHECKSUM optional keyword and values is used to test for data transmission errors in the file. The check sum is calculated by adding the ASCII values of all characters in the file, including spaces and line terminators. Trailing blanks (on each line) and the last line containing the checksum itself are excluded from this sum. If the calculated checksum agrees with that stored in the data file, the user can be reasonably sure that data set has been faithfully transmitted.

Optional user-defined keywords may be also included within the header of the data file. These keywords must begin with the double ## sign and follow all EMSA/MAS defined keywords, except for the start of data #SPECTRUM keyword.All field restrictions as outlined above apply to user-defined keywords and their values. These user defined allow for

expansion of this file structure to cover items that have not yet been explicitly defined by the EMSA/MAS Task Force.

2.4 ENDING A FILE

The last line of every file should be either an #ENDOFDATA or a #CHECKSUM line. The #ENDOFDATA keyword must immediately follow the spectral data, if present the #CHECKSUM keyword must follow the #ENDOFDATA keyword.

3.EXAMPLE OF A DATA FILE IN THE EMSA SPECTRAL FORMAT

The following examples show spectral files the minimum-required keyword information. Table 1 shows an ELS data set using the x,y pair data format, while Table 2 shows and EDS data set using y axis data only.

TABLE 1.
Example of a single column X,Y Data Set

```
#FORMAT      : EMSA/MAS Spectral Data File
#VERSION     : 1.0
#TITLE       : NIO EELS OK SHELL
#DATE        : 01-OCT-1991
#TIME        : 12:00
#OWNER       : EMSA/MAS TASK FORCE
#NPOINTS     : 20.
#NCOLUMNS    : 1.
#XUNITS      : Energy Loss (eV)
#YUNITS      : Intensity
#DATATYPE    : XY
#XPERCHAN    : 3.1
#OFFSET      : 520.13
#CHOFFSET    : -168
#SIGNALTYPE  : ELS
#XLABEL      :  Energy
#YLABEL      :  Counts
#BEAMKV   -kV: 120.0
#EMISSION -uA: 5.5
#PROBECUR -nA: 12.345
#BEAMDIAM -nm: 100.0
#MAGCAM      : 100.
#CONVANGLE-mR: 1.5
#COLLANGLE-mR: 3.4
#OPERMODE    : IMAG
#THICKNESS-nm: 50.
#DWELLTIME-ms: 100.
#ELSDET      : SERIAL
#SPECTRUM    : Spectral Data Starts Here
520.13,  4066.0
523.22,  3996.0
526.32,  3932.0
529.42,  3923.0
532.51,  5602.0
535.61,  5288.0
538.70,  7234.0
541.80,  7809.0
544.90,  4710.0
547.99,  5015.0
551.09,  4366.0
554.18,  4524.0
```

557.28, 4832.0
560.38, 5474.0
563.47, 5718.0
565.79, 5034.0
568.89, 4651.0
571.99, 4613.0
574.31, 4637.0
577.40, 4429.0
580.50, 4217.0
#ENDOFDATA   :
==========

TABLE 2.
Example of a 5 Column Y-Axis only Data Set

==========
#FORMAT   : EMSA/MAS SPECTRAL DATA STANDARD
#VERSION   : 1.0
#TITLE   : NIO Windowless Spectra OK NiL
#DATE   : 01-OCT-1991
#TIME   : 12:00
#OWNER   : EMSA/MAS TASK FORCE
#NPOINTS   : 80.
#NCOLUMNS   : 5.
#XUNITS   : Energy (eV)
#YUNITS   : Intensity
#DATATYPE   : Y
#XPERCHAN   : 10.
#OFFSET   : 200.
#CHOFFSET   : -20.
#SIGNALTYPE : EDS
#XLABEL   : X-RAY ENERGY
#YLABEL   : X-RAY INTENSITY
#BEAMKV   -kV: 120.0
#EMISSION -uA: 5.5
#PROBECUR -nA: 12.345
#BEAMDIAM -nm: 100.0
#MAGCAM   : 100
#OPERMODE   : IMAG
#THICKNESS-nm: 50
#XTILTSTGE-dg: 45.
#YTILTSTGE-dg: 20.
#XPOSITION   : 123.
#YPOSITION   : 456.
#ZPOSITION   : 000
#ELEVANGLE-dg: 20.
#AZIMANGLE-dg: 90.
#SOLIDANGL-sR: 0.13
#LIVETIME -s: 100.
#REALTIME -s: 150.
#TBEWIND -cm: 0.00
#TAUWIND -cm: 2.0 E-06
#TDEADLYR -cm: 1.0 E-06
#TACTLYR -cm: 0.3
#EDSDET   : SIWLS
#COMMENT   : The next two lines are User Defined Keywords and values
##ALPHA-1   : 3.1415926535
##RESTMASS   : 511.030
#SPECTRUM   : DATA BEGINS HERE
65.820,   67.872,   65.626,   68.762,   71.395,
74.996,   78.132,   78.055,   77.861,   84.598,
83.088,   85.372,   89.786,   93.464,   93.387,
97.452,   109.96,   111.08,   119.64,   128.77,

```
138.38,    152.35,    176.01,    192.12,    222.12,
254.22,    281.55,    328.33,    348.92,    375.33,
385.51,    389.54,    378.77,    353.80,    328.91,
290.07,    246.09,    202.73,    176.47,    137.64,
119.56,    106.40,    92.496,    96.213,    94.664,
101.13,    114.57,    118.82,    131.68,    145.04,
165.44,    187.51,    207.49,    238.04,    269.71,
301.46,    348.65,    409.36,    475.30,    554.51,
631.64,    715.19,    793.44,    847.99,    872.97,
862.59,    834.87,    778.49,    688.63,    599.39,
495.39,    403.48,    312.88,    237.34,    184.14,
129.86,    101.59,    80.107,    58.657,    49.442,
#ENDOFDATA   :
```

==========

4. EXAMPLES OF READ AND WRITE ROUTINES

Examples of a program (EMSAMASFF) and subroutines (RDEMSAMAS, WREMSAMAS) which utilize this format are available to all interested parties. They may be downloaded from either the EMMPDL or the EMSA/MAS BBS, information on accessing these is given in Tables 3 and 4 as well as articles in this issue of the EMSA Bulletin (In the Computer Corner, and EMSA/MAS BBS Instruction Manual) . Alternatively, they may also be obtained from the EMSA/MAS Task Force Chairman, R.F. Egerton.

Table 3
EMMPDL Telecommunications Protocol

Communication Lines: 300,1200-9600 Baud 8 data _& 1 stop bit, no parity
Modem Protocol : Hayes Smartmodem 1200, Bell 212A orcompatible
Transfer Protocol : Downloading using XON/XOFF ASCII transfer
handshaking or by error-checking using KERMIT
BITNET Electronic Mail
Charges : None, except for your own phone bill.
Lines/Times : Two Lines/24 Hours Per Day
Phone Number : 708-972-7919 (300-1200 Baud)
: 708-972-7918 (1200-9600 Baud)
Login Sequence : Username = EMMPDL
Password = EMMPDL
Electronic Mail : ZALUZEC at ANLEMC.Bitnet

Table 4
EMSA/MAS BBS Telecommunications Protocol

Communication Lines: 1200-2400 Baud 8 data _& 1 stop bit, no parity
Modem Protocol : Hayes Smartmodem 1200, Bell 212A or compatible
Transfer Protocol : Downloading using XON/XOFF ASCII transfer
handshaking or by error-checking using X,Y,Zmodem
Charges : None
Lines/Times : One Lines/24 Hours Per Day
Phone Number : 800-627-3672 (1200-2400 Baud)
: 800-MAS-EMSA
: 708-972-7917 (Commerical Number for EMSA/MAS

BBS)
Login Sequence : Username = Your Own Name
Password = Your Own Password

REFERENCES

[1]N.J. Zaluzec, `In the Computer Corner', EMSA Bulletin 17::1 (1987) p.93-94, and 'RWEMMPDL Abstract', EMSA Bulletin 17::2 (Nov 1987) p.92.

[2]R.S. McDonald and P.A. Wilks, `JCAMP-DX: A Standard for Exchange of Infrared Spectra in Computer Readable Form', Applied Spectroscopy 42 (1988) p.151-162.

[3]C. Lyman, `Task Force Committee Report', EMSA Bulletin 19 (1989) p.97-100.

[4]J.A. Hunt and C.E. Fiori, `A Proposed EMSA/MAS Data Format for the Transfer of Spectra', EMSA Bulletin 19 (1989) p.100-105.

[5]N.J. Zaluzec, `RWEMMPDL-VERSION 1.1 Abstract', EMSA Bulletin 19 (1989) p.114.

[6] W.A. Dench, L.B. Hazel, M.P Seah, and the VMAS Community, 'VMAS Surface Chemical Analysis Standard Data Transfer Format with Skeleton Decoding Programs', Surface and Interface Analysis, 13 (1988) p.63-122.

==========

COMPILATION, LINKING AND TESTING PROCEDURE

==========

The following procedure can be used to compile EMMFF using
IBM Professional Fortran Compiler by Ryan-McFarland Corp (1984)

```
C:>PROFORT EMMFF.SRC/L>EMMFF.LST
C:>LINK EMMFF
```

The following is a partial example of the I/O for EMMFF
at the present time all I/O options of the program
will not be demonstrated below.

Using EMMFF you may:

1.) Read an EMSA/MAS File
2.) Write an EMSA/MAS File
3.) Read and/or Edit (Modify) a System Parameters File
which contains defaults for your Spectroscopy system
4.) Translate EMMPDL files into EMSA/MAS files

==========

THE FOLLOWING IS A PARTIAL SCREEN CAPTURE OF THE EMMFF
PROGRAM RUNNING ON AN IBM PC AT

==========

```
C:\>EMMFF

          ************************************
          *                                  *
          * EMSA / MAS  File  Format  Program *
          *      Version Number :   1.00      *
          *          October 1991            *
          *                                  *
          ************************************



Chooze option by typing indicated character then a <RETURN>
------------------------------------------------------------

     R = Read  an EMSA/MAS Spectral Data File
     W = Write an EMSA/MAS Spectral Data File
     M = Modify System or File Parameters
     S = Read a System Parameters File
     T = Translate a NonEMSA/MAS Spectral Data File
     Q = Quit this Program

Select Option: [R,W,M,S,T,Q] ?=> R


Enter name of file [DEV:NAME.EXT]=> NIO.TXT

FORMAT      : EMSA/MAS SPECTRAL DATA STANDARD
VERSION     :  1.00000
TITLE       : NIO EELS OK SHELL AND NI L SHELL
DATE        : 01-OCT-1991
TIME        : 12:00
OWNER       : EMSA/MAS TASK FORCE
NPOINTS     :  10.0000
NCOLUMNS    :  1.00000
XUNITS      : Energy Loss (eV)
YUNITS      : Intensity
DATATYPE    : Y
XPERCHAN    :  2.50000
OFFSET      :  40.5780
CHOFFSET    :  100.000
SIGNALTYPE  : ELS
XLABEL      : xlabel
YLABEL      : ylabel
BEAMKV   -kV:  120.000
EMISSION -uA:  5.50000
PROBECUR -nA:  12.3450
BEAMDIAM -nm:  100.000
MAGCAM      :  100.000
CONVANGLE-mR:  15.0000
COLLANGLE-mR:  3.40000
OPERMODE    : IMAG
```

THICKNESS-nm: 50.0000
XTILTSTAGE : 45.0000
YTILTSTAGE : 20.0000
XPOSITION   : 123.000
YPOSITION   : 456.000
ZPOSITION   : 0.000000E+00
DWELLTIME-ms: 100.000
INTEGTIME-ms: 100.000
ELSDET   : Serial
ELEVANGLE   : 20.0000
AZIMANGLE   : 90.0000
SOLIDANGL-sR: 0.130000
LIVETIME -s: 100.000
REALTIME -s: 150.000
TBEWIND -cm: 0.800000E-03
TAUWIND -cm: 0.200000E-05
TDEADLYR -cm: 0.100000E-05
TACTLYR -cm: 0.300000
TALWIND -cm: 0.100000E-05
TPYWIND -cm: 0.100000E-05
TBNWIND -cm: 0.000000E-00
TDIWIND -cm: 0.000000E+00
THCWIND -cm: 0.000000E+00
EDSDET   : SIBEW
SPECTRUM   : Spectral Data Starts Here
==========Reading Spectral Data==========
ENDOFDATA   :
==========Data Input Complete==========
Do you want to print the data on your screen [Y,N]?=>Y

43.078	1.10000
45.578	22.2000
48.078	333.300
50.578	4444.40
53.078	55555.5
55.578	666667.
58.078	0.777778E+07
60.578	0.888889E+08
63.078	0.100000E+10
65.578	0.100000E+10

Chooze option by typing indicated character then a <RETURN>

R = Read an EMSA/MAS Spectral Data File
W = Write an EMSA/MAS Spectral Data File
M = Modify System or File Parameters
S = Read a System Parameters File
T = Translate a NonEMSA/MAS Spectral Data File
Q = Quit this Program

Select Option: [R,W,M,S,T,Q] ?=> Q

Quiting Program:

EMSA/MAS File Format Committee
==========

Ray F. Egerton------University of Alberta -Chairman
Charles E. Fiori----Nat. Inst. for Science & Technology
John A. Hunt--------Lehigh University
Michael S. Isaacson-Cornell University
Earl J. Kirkland----Cornell University
Nestor J. Zaluzec---Argonne National Laboratory

Execution terminated : 0

C:\>
==========
END OF DOCUMENTATION FILE

Title :EMMFF V. 1.0
Keywords :XEDS,EELS,AES,WDS,CLS,GAM,XRF,PES
Computer :IBM, MAC, DEC
Operating System :ALL
Programming Language :Fortran 77
Hardware Requirements :None
Author(s) :EMSA/MAS TASK FORCE
Ray Egerton ,Charles E. Fiori ,John A. Hunt,
Mike S. Isaacson,Earl J. Kirkland ,Nestor J. Zaluzec
Correspondence Address :R.F. EGERTON
University of Alberta
Dept. of Physics
Edmonton, Alberta, Canada, T6G2J1
SourceCode:

```
c=============================================================================
      Program EMMFF
c=============================================================================
c
c This program demonstrates the EMSA/MAS Spectrum File Format
c and the Read/Write Routines
c
c REVISION HISTORY
c
c Version  Date  Initials  Reason
c-----------------------------------------------------------------------------
c
c 1.0   10/01/91  NJZ    Creation of the World
c
c-----------------------------------------------------------------------------
c Required Subroutines:
c
c          RDEMMFF  - Reads an EMSA/MAS File
c          WREMMFF  - Writes an EMSA/MAS File
c          RDSYPARM - Reads a System Parameter File
c          RDNEMMFF - Reads/Translates a Non EMSA/MAS File
c                   *Note: RDNEMMF must be updated for
c                    Each Non Standard Format to be accessed
c          MODPARM  - Modifies System Parameters
c              TWRITE   - Utility routine to write a parameter
c              VWRITE   - Utility routine
c              TMODIFY  - Utility routine to modify/edit a parameter
c              VMODIFY  - Utility routine
c
c-----------------------------------------------------------------------------
c
c
```

```
c      This program is INTENTIONALLY written using a very structured
c      but simple architecture and contains lots of comments. Because of this
c      it is inefficient but informative to the inexperienced programmer.
c      Expert programmers will be able to modify the code easily. It was
c      purposefully written using simple constructions to aid
c      understand what each section is attempting to do for the novice user.
c      The key subroutines are RDEMMFF, and WREMMFF
c      and are located seperate to this main program code
c
c
c Title                :EMMFF
c Keywords             :EDS,ELS,AES,WDS,PES,XRF,CLS,GAM
c Computer             :VAX, Macintosh,IBM, PDP 11
c Operating System     :Any
c Programming Language :Fortran 77
c Hardware Requirements :None
c
c
c
c Definitions of Arrays used in this program
c
c      Note the symbol "<=" means "less than or equal to"
c           in the comment statements below
c
c XADATA   = Real Array <= 4096 values containing X-Axis Data
c YADATA   = Real Array <= 4096 values containing Y-Axis Data
c EXPARSPT = Real Array <=20 values containing Expt Parameters of Spectrum
c EXPARMSC = Real Array <=20 values containing Expt Parameters of Microscope
c EXPARSAM = Real Array <=20 values containing Expt Parameters of Sample
c EXPAREDS = Real Array <=20 values containing Expt Parameters of EDS
c EXPARELS = Real Array <=20 values containing Expt Parameters of ELS
c EXPARAES = Real Array <=20 values containing Expt Parameters of AES
c EXPARWDS = Real Array <=20 values containing Expt Parameters of WDS
c EXPARPES = Real Array <=20 values containing Expt Parameters of PES
c EXPARXRF = Real Array <=20 values containing Expt Parameters of XRF
c EXPARCLS = Real Array <=20 values containing Expt Parameters of CLS
c EXPARGAM = Real Array <=20 values containing Expt Parameters of GAM
c SIGNALTY = Byte Array of 3 characters describing the signal type
c DATATYPE = Byte Array of 2 characters describing the format
c OPERMODE = Byte Array of 5 characters describing
c           the instrument operating mode
c EDSDET   = Byte Array of 6 characters describing the EDS detector type
c ELSDET   = Byte Array of 6 characters describing the ELS detector type
c XUNITS   = Byte Array <= 64 characters describing the X-Axis Data units
c YUNITS   = Byte Array <= 64 characters describing the Y-Axis Data units
c XLABEL   = Byte Array <= 64 characters describing the X-Axis Data label
c YLABEL   = Byte Array <= 64 characters describing the Y-Axis Data label
c TITLE    = Byte Array <= 64 characters with a spectrum Title
c DATE     = Byte Array of 12 characters with the date of acquisition
c           in the form DD-MMM-YYYY
c TIME     = Byte Array of 5 characters with the time of acquisition
c           in the form HH:MM
c OWNER    = Byte Array <= 64 characters with the owner/analysts name
c COMMENT  = Byte Array <= 64 character used to hold a text comment string
c
c-------------------------------------------------------------------------
c
c Current Definitions of EXPAR (Experimental Parameter Array) Values
c
c-------------------------------------------------------------------------
c  Parameters relating to the Spectrum Characteristics = EXPARSPT
```

```
c--------------------------------------------------------------------------
c
c Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
c Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),XLABEL(64),YLABEL(64),DATATYPE(2)
c
c EXPARSPT(1) = #VERSION  = File Format Version Number
c EXPARSPT(2) = #NPOINTS  = Total Number of Data Points in X&Y Data Arrays
c               1 <= NPOINTS <= 4096
c EXPARSPT(3) = #NCOLUMNS = Number of columns of data
c               1 <= NCOLUMNS <= 5
c EXPARSPT(4) = #XPERCHAN = Increment of X-axis units per channel (eV/Channel)
c              This is only useful if (x,y) paired data are not provided
c EXPARSPT(5) = #OFFSET   = Energy value of first data point in eV
c             This is only useful if (x,y) paired data are not provided
c EXPARSPT(6) = #CHOFFSET = Channel number which corresponds to zero units along
c             x-axis, this may be either a positive or negative value
C SIGNALTY(.) = #SIGNALTYPE = Type of Spectroscopy
c       EDS = Energy Dispersive Spectroscopy
c       WDS = Wavelength Dispersive Spectroscopy
c       ELS = Energy Loss Spectroscopy
c       AES = Auger Electron Spectroscopy
c       PES = Photo Electron Spectroscopy
c       XRF = X-ray Fluorescence Spectroscopy
c       CLS = Cathodoluminescence Spectroscopy
c       GAM = Gamma Ray Spectroscopy
c XUNITS(.)   = #XUNITS = Up to 64 characters describing the X-Axis Data units
c YUNITS(.)   = #YUNITS = Up to 64 characters describing the Y-Axis Data units
c XLABEL(.)   = #XLABEL = Up to 64 characters describing the X-Axis Data label
c YLABEL(.)   = #YLABEL = Up to 64 characters describing the Y-Axis Data label
C DATATYPE(.) = #DATATYPE = Type of data format
c          Y = Spectrum Y axis data only, X-axis data to be calculated
c           using XPERCHAN and OFFSET and the following formulae
c            X = OFFSET + CHANNEL*XPERCHAN
c          XY = Spectral data is in the form of XY pairs
c
c
c--------------------------------------------------------------------------
c  Microscope/Microanalysis Instrument Parameters = EXPARMSC
c--------------------------------------------------------------------------
c
c Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,
c CONVANGLE,COLLANGLE
c Character*1 OPERMODE(5)
c
c EXPARMSC(1) = #BEAMKV    = Accelerating Voltage of Instrument in kV
c EXPARMSC(2) = #EMISSION   = Gun Emission current in microAmps
c EXPARMSC(3) = #PROBECUR   = Probe current in nanoAmps
c EXPARMSC(4) = #BEAMDIAM   = Diameter of incident probe in nanometers
c EXPARMSC(5) = #MAGCAM   = Magnification or Camera Length (Mag in x, Cl in mm)
c EXPARMSC(6) = #CONVANGLE  = Convergence semi-angle of incident beam in milliRadians
c EXPARMSC(7) = #COLLANGLE  = Collection semi-angle of scattered beam in milliRad
c OPERMODE(.) = #OPERMODE   = Operating Mode of Instrument
c                       IMAGE = Imaging Mode
c                       DIFFR = Diffraction Mode
c                       SCIMG = Scanning Imaging Mode
c                       SCDIF = Scanning Diffraction Mode
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating to the Sample = EXPARSAM
c--------------------------------------------------------------------------
c
```

```
c Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
c      ZPOSITION
c
c EXPARSAM(1) = #THICKNESS = Specimen thickness in nanometers
c EXPARSAM(2) = #XTILTSTGE = Specimen stage tilt X-axis in degrees
c EXPARSAM(3) = #YTILTSTGE = Specimen stage tilt Y-axis in degrees
c EXPARSAM(4) = #XPOSITION =  X location of beam or specimen
c EXPARSAM(5) = #YPOSITION =  Y location of beam or specimen
c EXPARSAM(6) = #ZPOSITION =  Z location of beam or specimen
c
c
c-------------------------------------------------------------------------
c  Experimental Parameters relating mainly to ELS = EXPARELS
c-------------------------------------------------------------------------
c
c Real EXPARELS(20),DWELLTIME,INTEGTIME
c Character*1 ELSDET(6)
c
c EXPARELS(1) = #DWELLTIME  = Dwell time per channel for serial data collection in msec
c EXPARELS(2) = #INTEGTIME  = Integration time per spectrum for parallel data collection
c            in milliseconds
c ELSDET(.)  = #ELSDET   = Type of ELS Detector
c            Serial = Serial ELS Detector
c            Parall = Parallel ELS Detector
c
c-------------------------------------------------------------------------
c  Experimental Parameters relating mainly to EDS = EXPAREDS
c-------------------------------------------------------------------------
c
c Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
c Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
c Character*1 EDSDET(6)
c
c EXPAREDS(1) = #ELEVANGLE = Elevation angle of EDS,WDS detector in degrees
c EXPAREDS(2) = #AZIMANGLE = Azimuthal angle of EDS,WDS detector in degrees
c EXPAREDS(3) = #SOLIDANGLE = Collection solid angle of detector in sR
c EXPAREDS(4) = #LIVETIME  = Signal Processor Active (Live) time in seconds
c EXPAREDS(5) = #REALTIME  = Total clock time used to record the spectrum in seconds
c EXPAREDS(6) = #TBEWIND  = Thickness of Be Window on detector in cm
c EXPAREDS(7) = #TAUWIND  = Thickness of Au Window/Electrical Contact in cm
c EXPAREDS(8) = #TDEADLYR  = Thickness of Dead Layer in cm
c EXPAREDS(9) = #TACTLYR  = Thickness of Active Layer in cm
c EXPAREDS(10) = #TALWIND  = Thickness of Aluminium Window in cm
c EXPAREDS(11) = #TPYWIND  = Thickness of Pyrolene Window in cm
c EXPAREDS(12) = #TBNWIND  = Thickness of Boron-Nitride Window in cm
c EXPAREDS(13) = #TDIWIND  = Thickness of Diamond Window in cm
c EXPAREDS(14) = #THCWIND  = Thickness of HydroCarbon Window in cm
c EDSDET(.)    = #EDSDET  = Type of X-ray Detector
c        SIBEW = Si(Li) with Be Window
c        SIUTW = Si(Li) with Ultra Thin Window
c        SIWLS = Si(Li) Windowless
c        GEBEW = Ge with Be Window
c        GEUTW = Ge with Ultra Thin Window
c        GEWLS = Ge Windowless
c
c-------------------------------------------------------------------------
c  Experimental Parameters relating mainly to WDS = EXPARWDS
c-------------------------------------------------------------------------
c
c      Nothing currently defined
c
```

```
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to XRF = EXPARXRF
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to AES = EXPARAES
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to PES = EXPARPES
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to CLS = EXPARCLS
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to GAM = EXPARGAM
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c============================================================================
c         END OF DEFINITIONS
c============================================================================
c
c Start Code
c
c
c   Dimension all arrays, & Declare Variable Types
C    REMEMBER we have no integer parameters!!!!!
c
c
       Character*1 TITLE(64),DATE(12),TIME(5),OWNER(64),COMMENT(64)
       Real XADATA(4096),YADATA(4096)
       Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
       Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),
     *         XLABEL(64),YLABEL(64),DATATYPE(2)
       Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,CONVANGLE,
     *       COLLANGLE
       Character*1 OPERMODE(5)
       Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
     *       ZPOSITION
       Real EXPARELS(20),DWELLTIME,INTEGTIME
       Character*1 ELSDET(6)
       Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
       Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
       Character*1 EDSDET(6)
       Real EXPARWDS(20),EXPARXRF(20),EXPARAES(20),EXPARPES(20),EXPARCLS(20),
     *       EXPARGAM(20)
C
c-----------------------------------------------------------------------------
c Set up all Variable arrays to be passed in labeled common block for ease of
```

```
c passing parameters between different subroutines
c-------------------------------------------------------------------------------
c
      Common /XYDATA/ XADATA,YADATA
      Common /EXPPAR/ EXPARSPT,EXPARMSC,EXPARSAM,EXPARELS,EXPAREDS,EXPARWDS,EXPARAES,
     *    EXPARPES,EXPARXRF,EXPARCLS,EXPARGAM
      Common /TEXT/ TITLE,DATE,TIME,OWNER,COMMENT,
     *    SIGNALTY,XUNITS,YUNITS,XLABEL,YLABEL,DATATYPE,
     *    OPERMODE,ELSDET,EDSDET
c
      Character*1 ANS,OKAY
c
c     Now Equivalence everything so that it will be readable
c     rather than cryptic
c
      EQUIVALENCE ( EXPARSPT(1),VERSION)
      EQUIVALENCE ( EXPARSPT(2),NPOINTS)
      EQUIVALENCE ( EXPARSPT(3),NCOLUMNS)
      EQUIVALENCE ( EXPARSPT(4),XPERCHAN)
      EQUIVALENCE ( EXPARSPT(5),OFFSET)
      EQUIVALENCE ( EXPARSPT(6),CHOFFSET)
      EQUIVALENCE ( EXPARMSC(1),BEAMKV)
      EQUIVALENCE ( EXPARMSC(2),EMISSION)
      EQUIVALENCE ( EXPARMSC(3),PROBECUR)
      EQUIVALENCE ( EXPARMSC(4),BEAMDIAM)
      EQUIVALENCE ( EXPARMSC(5),MAGCAM)
      EQUIVALENCE ( EXPARMSC(6),CONVANGLE)
      EQUIVALENCE ( EXPARMSC(7),COLLANGLE)
      EQUIVALENCE ( EXPARSAM(1),THICKNESS)
      EQUIVALENCE ( EXPARSAM(2),XTILTSTGE)
      EQUIVALENCE ( EXPARSAM(3),YTILTSTGE)
      EQUIVALENCE ( EXPARSAM(4),XPOSITION)
      EQUIVALENCE ( EXPARSAM(5),YPOSITION)
      EQUIVALENCE ( EXPARSAM(6),ZPOSITION)
      EQUIVALENCE ( EXPARELS(1),DWELLTIME)
      EQUIVALENCE ( EXPARELS(2),INTEGTIME)
      EQUIVALENCE ( EXPAREDS(1),ELEVANGLE)
      EQUIVALENCE ( EXPAREDS(2),AZIMANGLE)
      EQUIVALENCE ( EXPAREDS(3),SOLIDANGLE)
      EQUIVALENCE ( EXPAREDS(4),LIVETIME)
      EQUIVALENCE ( EXPAREDS(5),REALTIME)
      EQUIVALENCE ( EXPAREDS(6),TBEWIND)
      EQUIVALENCE ( EXPAREDS(7),TAUWIND)
      EQUIVALENCE ( EXPAREDS(8),TDEADLYR)
      EQUIVALENCE ( EXPAREDS(9),TACTLYR)
      EQUIVALENCE ( EXPAREDS(10),TALWIND)
      EQUIVALENCE ( EXPAREDS(11),TPYWIND)
      EQUIVALENCE ( EXPAREDS(12),TBNWIND)
      EQUIVALENCE ( EXPAREDS(13),TDIWIND)
      EQUIVALENCE ( EXPAREDS(14),THCWIND)
c
c
c-------------------------------------------------------------------------------
c Fill in constants
c-------------------------------------------------------------------------------
c
      VERSION = 1.0
c
c-------------------------------------------------------------------------------
c Begin the program by writing a program title
c-------------------------------------------------------------------------------
```

```
c
1      WRITE (6,10) Version
10     FORMAT (
     *  21X,'*************************************',/,
     *  21X,'*                                   *',/,
     *  21X,'* EMSA / MAS  File  Format  Program *',/,
     *  21X,'*      Version Number : ',1F6.2,'      *',/,
     *  21X,'*           October 1991            *',/,
     *  21X,'*                                   *',/,
     *  21X,'*************************************',//)

C
c      Begin by asking the user what (s)he wants to do
c      R=Read an EMSAMAS File
c      W=Write an EMSAMAS File
c      S=Read System Parameters File
c      T=Translate a non EMSA/MAS File Format
c      M=Modifies System Parameters
c      Q=Quit the program
c
11     Write (6,20)
20     Format (1X,//,1x,
     *  'Chooze option by typing indicated character then a <RETURN>',/,
     *  '-----------------------------------------------------------',//,
     *  10X,'R = Read  an EMSA/MAS Spectral Data File',/,
     *  10X,'W = Write an EMSA/MAS Spectral Data File',/,
     *  10X,'M = Modify System or File Parameters',/,
     *  10X,'S = Read a System Parameters File',/,
     *  10x,'T = Translate a NonEMSA/MAS Spectral Data File',/,
     *  10X,'Q = Quit this Program',//,
     *  ' Select Option: [R,W,M,S,T,Q] ?=> ')
       Read (5,30) ANS
30     Format (1A1)
C
C      Just in case make sure we are not case sensitive
C
c
c      do we want to quit the program?
c
       IF ((ANS.EQ.'Q').or.(ANS.EQ.'q')) Go to 400
c
c      we did not want to quit so now do something
c
       IF ((ANS.EQ.'R').OR.(ANS.EQ.'r')) CALL RDEMMFF
       IF ((ANS.EQ.'W').OR.(ANS.EQ.'w')) CALL WREMMFF
       IF ((ANS.EQ.'S').OR.(ANS.EQ.'s')) CALL RDSYPARM
       IF ((ANS.EQ.'M').OR.(ANS.EQ.'m')) CALL MODPARM
       IF ((ANS.EQ.'T').OR.(ANS.EQ.'t')) CALL RDNEMMFF
c
c      If ANS NE Read or Write there was an operator input error
c       force user to try again
c
       Go to 11
c
c We are all done Say Goodbye
c
400    Write (6,410)
410    Format (
     *25X,'       Quiting Program:',//,
     *25X,' EMSA/MAS File Format Committee',/,
     *25X,' ==============================',/,
```

```
     *10X,' Ray F. Egerton------University of Alberta -Chairman',/,
     *10X,' Charles E. Fiori----Nat. Inst. for Science & Technology',/,
     *10X,' John A. Hunt--------Lehigh University',/,
     *10X,' Michael S. Isaacson-Cornell University',/,
     *10X,' Earl J. Kirkland----Cornell University',/,
     *10X,' Nestor J. Zaluzec---Argonne National Laboratory',//)
      STOP
      END

C=====================================================================
        Subroutine RDEMMFF
C=====================================================================
c
c
c This subroutine  READS the EMSA/MAS Spectrum File Format.
c The results  from the file are stored locally in predefined
c arrays. These arrays can be passed directly back to the main
c program and can be used to redirect data elsewhere as needed.
c In this implementation the arrays are passed in labeled common
c blocks. They can alternatively be passed as subroutine
c arguments.
c
c REVISION HISTORY
c
c Version  Date  Initials  Reason
c-----------------------------------------------------------------------
c
c 1.0   10/01/91  NJZ    Creation of the World WARNING THIS VERSION
c                       Case Sensitive!!! Until DeBugged
c
c-----------------------------------------------------------------------
c Definitions of Arrays used in this Subroutine
c
c
c
c XADATA   = Real Array <= 4096 values containing X-Axis Data
c YADATA   = Real Array <= 4096 values containing Y-Axis Data
c EXPARSPT = Real Array <=20 values containing Expt Parameters of Spectrum
c EXPARMSC = Real Array <=20 values containing Expt Parameters of Microscope
c EXPARSAM = Real Array <=20 values containing Expt Parameters of Sample
c EXPAREDS = Real Array <=20 values containing Expt Parameters of EDS
c EXPARELS = Real Array <=20 values containing Expt Parameters of ELS
c EXPARAES = Real Array <=20 values containing Expt Parameters of AES
c EXPARWDS = Real Array <=20 values containing Expt Parameters of WDS
c EXPARPES = Real Array <=20 values containing Expt Parameters of PES
c EXPARXRF = Real Array <=20 values containing Expt Parameters of XRF
c EXPARCLS = Real Array <=20 values containing Expt Parameters of CLS
c EXPARGAM = Real Array <=20 values containing Expt Parameters of GAM
c SIGNALTY = Byte Array of 3 characters describing the signal type
c DATATYPE = Byte Array of 2 characters describing the format
c OPERMODE = Byte Array of 5 characters describing
c           the instrument operating mode
c EDSDET   = Byte Array of 6 characters describing the EDS detector type
c ELSDET   = Byte Array of 6 characters describing the ELS detector type
c XUNITS   = Byte Array <= 64 characters describing the X-Axis Data units
c YUNITS   = Byte Array <= 64 characters describing the Y-Axis Data units
c XLABEL   = Byte Array <= 64 characters describing the X-Axis Data label
c YLABEL   = Byte Array <= 64 characters describing the Y-Axis Data label
c TITLE    = Byte Array <= 64 characters with a spectrum Title
c DATE     = Byte Array of 12 characters with the date of acquisition
c           in the form DD-MMM-YYYY
```

```
c TIME     = Byte Array of 5 characters with the time of acquisition
c            in the form HH:MM
c OWNER    = Byte Array <= 64 characters with the owner/analysts name
c COMMENT  = Byte Array <= 64 Character*1 used to hold a text comment string
c
c---------------------------------------------------------------------------
c
c Current Definitions of EXPAR (Experimental Parameter Array) Values
c
c---------------------------------------------------------------------------
c  Parameters relating to the Spectrum Characteristics = EXPARSPT
c---------------------------------------------------------------------------
c
c Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
c Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),XLABEL(64),YLABEL(64),DATATYPE(2)
c
c EXPARSPT(1) = #VERSION  = File Format Version Number
c EXPARSPT(2) = #NPOINTS  = Total Number of Data Points in X&Y Data Arrays
c          1 <= NPOINTS <= 4096
c EXPARSPT(3) = #NCOLUMNS = Number of columns of data
c          1 <= NCOLUMNS <= 5
c EXPARSPT(4) = #XPERCHAN = Increment of X-axis units per channel (eV/Channel)
c            This is only useful if (x,y) paired data are not provided
c EXPARSPT(5) = #OFFSET   = Energy value of first data point in eV
c            This is only useful if (x,y) paired data are not provided
c EXPARSPT(6) = #CHOFFSET = Channel number which corresponds to zero units along
c             x-axis, this may be either a positive or negative value
C SIGNALTY(.) = #SIGNALTYPE = Type of Spectroscopy
c        EDS = Energy Dispersive Spectroscopy
c        WDS = Wavelength Dispersive Spectroscopy
c        ELS = Energy Loss Spectroscopy
c        AES = Auger Electron Spectroscopy
c        PES = Photo Electron Spectroscopy
c        XRF = X-ray Fluorescence Spectroscopy
c        CLS = Cathodoluminescence Spectroscopy
c        GAM = Gamma Ray Spectroscopy
c XUNITS(.)   = #XUNITS = Up to 64 characters describing the X-Axis Data units
c YUNITS(.)   = #YUNITS = Up to 64 characters describing the Y-Axis Data units
c XLABEL(.)   = #XLABEL = Up to 64 characters describing the X-Axis Data label
c YLABEL(.)   = #YLABEL = Up to 64 characters describing the Y-Axis Data label
C DATATYPE(.) = #DATATYPE = Type of data format
c           Y = Spectrum Y axis data only, X-axis data to be calculated
c            using XPERCHAN and OFFSET and the following formulae
c             X = OFFSET + CHANNEL*XPERCHAN
c           XY = Spectral data is in the form of XY pairs
c
c
c---------------------------------------------------------------------------
c  Microscope/Microanalysis Instrument Parameters = EXPARMSC
c---------------------------------------------------------------------------
c
c Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,
c CONVANGLE,COLLANGLE
c Character*1 OPERMODE(5)
c
c EXPARMSC(1) = #BEAMKV    = Accelerating Voltage of Instrument in kV
c EXPARMSC(2) = #EMISSION   = Gun Emission current in microAmps
c EXPARMSC(3) = #PROBECUR   = Probe current in nanoAmps
c EXPARMSC(4) = #BEAMDIAM   = Diameter of incident probe in nanometers
c EXPARMSC(5) = #MAGCAM   = Magnification or Camera Length (Mag in x, Cl in mm)
c EXPARMSC(6) = #CONVANGLE  = Convergence semi-angle of incident beam in milliRadians
```

```
c EXPARMSC(7) = #COLLANGLE  = Collection semi-angle of scattered beam in milliRad
c OPERMODE(.) = #OPERMODE   = Operating Mode of Instrument
c                       IMAGE = Imaging Mode
c                       DIFFR = Diffraction Mode
c                       SCIMG = Scanning Imaging Mode
c                       SCDIF = Scanning Diffraction Mode
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating to the Sample = EXPARSAM
c--------------------------------------------------------------------------
c
c Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
C      ZPOSITION
c
c EXPARSAM(1) = #THICKNESS = Specimen thickness in nanometers
c EXPARSAM(2) = #XTILTSTGE = Specimen stage tilt X-axis in degrees
c EXPARSAM(3) = #YTILTSTGE = Specimen stage tilt Y-axis in degrees
c EXPARSAM(4) = #XPOSITION =  X location of beam or specimen
c EXPARSAM(5) = #YPOSITION =  Y location of beam or specimen
c EXPARSAM(6) = #ZPOSITION =  Z location of beam or specimen
c
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to ELS = EXPARELS
c--------------------------------------------------------------------------
c
c Real EXPARELS(20),DWELLTIME,INTEGTIME
c Character*1 ELSDET(6)
c
c EXPARELS(1) = #DWELLTIME  = Dwell time per channel for serial data collection in msec
c EXPARELS(2) = #INTEGTIME  = Integration time per spectrum for parallel data collection
c            in milliseconds
c ELSDET(.)  = #ELSDET   = Type of ELS Detector
c            Serial = Serial ELS Detector
c            Parall = Parallel ELS Detector
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to EDS = EXPAREDS
c--------------------------------------------------------------------------
c
c Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
c Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
c Character*1 EDSDET(6)
c
c EXPAREDS(1) = #ELEVANGLE = Elevation angle of EDS,WDS detector in degrees
c EXPAREDS(2) = #AZIMANGLE = Azimuthal angle of EDS,WDS detector in degrees
c EXPAREDS(3) = #SOLIDANGLE = Collection solid angle of detector in sR
c EXPAREDS(4) = #LIVETIME  = Signal Processor Active (Live) time in seconds
c EXPAREDS(5) = #REALTIME  = Total clock time used to record the spectrum in seconds
c EXPAREDS(6) = #TBEWIND  = Thickness of Be Window on detector in cm
c EXPAREDS(7) = #TAUWIND  = Thickness of Au Window/Electrical Contact in cm
c EXPAREDS(8) = #TDEADLYR  = Thickness of Dead Layer in cm
c EXPAREDS(9) = #TACTLYR  = Thickness of Active Layer in cm
c EXPAREDS(10) = #TALWIND  = Thickness of Aluminium Window in cm
c EXPAREDS(11) = #TPYWIND  = Thickness of Pyrolene Window in cm
c EXPAREDS(12) = #TBNWIND  = Thickness of Boron-Nitride Window in cm
c EXPAREDS(13) = #TDIWIND  = Thickness of Diamond Window in cm
c EXPAREDS(14) = #THCWIND  = Thickness of HydroCarbon Window in cm
c EDSDET(.)    = #EDSDET  = Type of X-ray Detector
c        SIBEW = Si(Li) with Be Window
c        SIUTW = Si(Li) with Ultra Thin Window
```

```
c        SIWLS = Si(Li) Windowless
c        GEBEW = Ge with Be Window
c        GEUTW = Ge with Ultra Thin Window
c        GEWLS = Ge Windowless
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to WDS = EXPARWDS
c--------------------------------------------------------------------------
c
c     Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to XRF = EXPARXRF
c--------------------------------------------------------------------------
c
c     Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to AES = EXPARAES
c--------------------------------------------------------------------------
c
c     Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to PES = EXPARPES
c--------------------------------------------------------------------------
c
c     Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to CLS = EXPARCLS
c--------------------------------------------------------------------------
c
c     Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to GAM = EXPARGAM
c--------------------------------------------------------------------------
c
c     Nothing currently defined
c
c============================================================================
c        END OF DEFINITIONS
c============================================================================
c
c
c Start Code
c
c
c   Dimension all arrays, & Declare Variable Types
c   REMEMBER we have no integer parameters!!!!!
c
c
      Character*1 TITLE(64),DATE(12),TIME(5),OWNER(64),COMMENT(64)
      Real XADATA(4096),YADATA(4096)
      Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
      Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),
     *        XLABEL(64),YLABEL(64),DATATYPE(2)
      Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,CONVANGLE,
     *      COLLANGLE
      Character*1 OPERMODE(5)
```

```
      Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
     *     ZPOSITION
      Real EXPARELS(20),DWELLTIME,INTEGTIME
      Character*1 ELSDET(6)
      Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
      Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
      Character*1 EDSDET(6)
      Real EXPARWDS(20),EXPARXRF(20),EXPARAES(20),EXPARPES(20),EXPARCLS(20),
     *     EXPARGAM(20)
c-------------------------------------------------------------------------------
c Set up all Variable arrays to be passed in labeled common block for ease of
c passing parameters between different subroutines
c-------------------------------------------------------------------------------
c
      Common /XYDATA/ XADATA,YADATA
      Common /EXPPAR/ EXPARSPT,EXPARMSC,EXPARSAM,EXPARELS,EXPAREDS,EXPARWDS,EXPARAES,
     *  EXPARPES,EXPARXRF,EXPARCLS,EXPARGAM
      Common /TEXT/ TITLE,DATE,TIME,OWNER,COMMENT,
     *  SIGNALTY,XUNITS,YUNITS,XLABEL,YLABEL,DATATYPE,
     *  OPERMODE,ELSDET,EDSDET
c
      Character*1 ANS,NUM,FKEY
      Character*5 KEY
      Character*9 WORD
      EQUIVALENCE ( EXPARSPT(1),VERSION)
      EQUIVALENCE ( EXPARSPT(2),NPOINTS)
      EQUIVALENCE ( EXPARSPT(3),NCOLUMNS)
      EQUIVALENCE ( EXPARSPT(4),XPERCHAN)
      EQUIVALENCE ( EXPARSPT(5),OFFSET)
      EQUIVALENCE ( EXPARSPT(6),CHOFFSET)
      EQUIVALENCE ( EXPARMSC(1),BEAMKV)
      EQUIVALENCE ( EXPARMSC(2),EMISSION)
      EQUIVALENCE ( EXPARMSC(3),PROBECUR)
      EQUIVALENCE ( EXPARMSC(4),BEAMDIAM)
      EQUIVALENCE ( EXPARMSC(5),MAGCAM)
      EQUIVALENCE ( EXPARMSC(6),CONVANGLE)
      EQUIVALENCE ( EXPARMSC(7),COLLANGLE)
      EQUIVALENCE ( EXPARSAM(1),THICKNESS)
      EQUIVALENCE ( EXPARSAM(2),XTILTSTGE)
      EQUIVALENCE ( EXPARSAM(3),YTILTSTGE)
      EQUIVALENCE ( EXPARSAM(4),XPOSITION)
      EQUIVALENCE ( EXPARSAM(5),YPOSITION)
      EQUIVALENCE ( EXPARSAM(6),ZPOSITION)
      EQUIVALENCE ( EXPARELS(1),DWELLTIME)
      EQUIVALENCE ( EXPARELS(2),INTEGTIME)
      EQUIVALENCE ( EXPAREDS(1),ELEVANGLE)
      EQUIVALENCE ( EXPAREDS(2),AZIMANGLE)
      EQUIVALENCE ( EXPAREDS(3),SOLIDANGLE)
      EQUIVALENCE ( EXPAREDS(4),LIVETIME)
      EQUIVALENCE ( EXPAREDS(5),REALTIME)
      EQUIVALENCE ( EXPAREDS(6),TBEWIND)
      EQUIVALENCE ( EXPAREDS(7),TAUWIND)
      EQUIVALENCE ( EXPAREDS(8),TDEADLYR)
      EQUIVALENCE ( EXPAREDS(9),TACTLYR)
      EQUIVALENCE ( EXPAREDS(10),TALWIND)
      EQUIVALENCE ( EXPAREDS(11),TPYWIND)
      EQUIVALENCE ( EXPAREDS(12),TBNWIND)
      EQUIVALENCE ( EXPAREDS(13),TDIWIND)
      EQUIVALENCE ( EXPAREDS(14),THCWIND)
c
c
```

```
c      NOW define different arrays used only in this routine
c
c      XVALUE & YVALUE are temporary arrays used for reading in values
c
       REAL XVALUE(10),YVALUE(10)
c
c
c The array FNAME = Name of File to be read, Usually null terminated
c so we will initialize it with zero's using a Do loop so that successive
c operations of the routine within a single program donot cause problems
c if a short filename is used.
c
c FNAME   = Byte array of 17 characters containing the filename
c KText  = Byte array of 64 characters containing Keyword text
c
c
       Character*17 FNAME
       Character*1 KTEXT(64)
       Logical*1 TF
c
c Initialize Filename
c
c
c
c
c Begin by asking the user to supply a file name
c Note the following statement is generally compiler dependant
c Note this version limits file name to 16 characters
c
c
109    WRITE (6,110)
110    FORMAT (' Enter name of file [DEV:NAME.EXT]=> ')
       READ (5,120) FNAME
120    FORMAT(1A16)
c
c     check to see if the file exists
c
      INQUIRE (FILE=FNAME,EXIST=TF)
      IF(TF) THEN
c
c           It exists so now open the file
c
               OPEN (UNIT=99,STATUS='OLD',BLANK='NULL',
     *               ACCESS='SEQUENTIAL',FILE=FNAME)
           ELSE
c
c           It does not exist so tell the user
c
               WRITE (6,*) 'FILENAME Does Not Exit: Re-enter'
               Go to 109
           ENDIF
c
c Now lets read and write the required parameters as defined by EMSA/MAS
c
c
c First read the Format Title
c
       READ (99,125) NUM,KEY,WORD,KTEXT
125    FORMAT (1A1,1A5,1A9,64A1)
C
C Check to see if it is legal
```

```
c
      IF (NUM.NE.'#')  GO TO  991
      IF (KEY.NE.'FORMA') GO TO 991
c
c           We have a legal format title write it out
c
      WRITE (6,125) NUM,KEY,WORD,KTEXT
c
c now the version
c
      READ (99,130) NUM,KEY,WORD,VERSION
      IF (NUM.NE.'#')  GO TO 991
      IF (KEY.NE.'VERSI') GO TO 991
      WRITE (6,1301) NUM,KEY,WORD,VERSION
c
c           Compare the version number are we version 1.0?
c           if not you need a different routine
c
c
      IF (VERSION.NE.1.) GO TO 991
130    FORMAT(1A1,1A5,1A9,1G12.6)
1301   FORMAT(1A1,1A5,1A9,1G12.6)
c
c Okay we have a legal Format and Version number
c we now read in all keywords as predefined by EMSA/MAS
c Each will be checked individually for safety, rather that
c assume the data file is perfect.
c
c
c Does the data start? Read the next line
c write it out regardless of its value and
c but look for the keyword "SPECTRUM"
c
      NUMKEY=2
129    READ (99,125) NUM,KEY,WORD,KTEXT
      IF(KEY.EQ.'SPECT') THEN
            WRITE (6,125) NUM,KEY,WORD,KTEXT
            GO TO 1000
            ENDIF
c
c the key word not SPECT, hence there is more information
c lets read it.
c
c       first check to see how many keywords we have read in
c       if greater than 100 there is probably an error
c
       NUMKEY=NUMKEY+1
       IF(NUMKEY.GT.100) GO TO 991
c
c First Backspace and compare with defined Keys,
c then read the value and store it as appropriate
c
c
      BACKSPACE (99)
c
c
c
c now test each predefined keyword
c
       FKEY='F'
c
```

```
c      define a false flag key FKEY to indicate a nonEMSA/MAS
c             keyword in the input. This word will be printed
c             on the screen but not used in the program.
c
      IF(KEY.EQ.'TITLE') THEN
            READ (99,125) NUM,KEY,WORD,TITLE
            WRITE (6,125) NUM,KEY,WORD,TITLE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'DATE ') THEN
            READ (99,125) NUM,KEY,WORD,DATE
            WRITE (6,125) NUM,KEY,WORD,DATE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TIME') THEN
            READ (99,125) NUM,KEY,WORD,TIME
            WRITE (6,125) NUM,KEY,WORD,TIME
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'OWNER') THEN
            READ (99,125) NUM,KEY,WORD,OWNER
            WRITE (6,125) NUM,KEY,WORD,OWNER
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'NPOIN') THEN
            READ (99,130) NUM,KEY,WORD,NPOINTS
            WRITE (6,1301) NUM,KEY,WORD,NPOINTS
                IF(NPOINTS.GT.4096) GO TO 991
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'NCOLU') THEN
            READ (99,130) NUM,KEY,WORD,NCOLUMNS
            WRITE (6,1301) NUM,KEY,WORD,NCOLUMNS
                  IF (NCOLUMNS.GT.5) GO TO 991
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'XUNIT') THEN
            READ (99,125) NUM,KEY,WORD,XUNITS
            WRITE (6,125) NUM,KEY,WORD,XUNITS
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'YUNIT') THEN
            READ (99,125) NUM,KEY,WORD,YUNITS
            WRITE (6,125) NUM,KEY,WORD,YUNITS
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'DATAT') THEN
            READ (99,125) NUM,KEY,WORD,DATATYPE
            WRITE (6,125) NUM,KEY,WORD,DATATYPE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'XPERC') THEN
```

```
            READ (99,130) NUM,KEY,WORD,XPERCHAN
            WRITE (6,1301) NUM,KEY,WORD,XPERCHAN
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'OFFSE') THEN
            READ (99,130) NUM,KEY,WORD,OFFSET
            WRITE (6,1301) NUM,KEY,WORD,OFFSET
            FKEY='T'
            GO TO 129
            ENDIF
c
c     The above are the required keywords
c     the following are optional keywords

      IF(KEY.EQ.'CHOFF') THEN
            READ (99,130) NUM,KEY,WORD,CHOFFSET
            WRITE (6,1301) NUM,KEY,WORD,CHOFFSET
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'SIGNA') THEN
            READ (99,125) NUM,KEY,WORD,SIGNALTY
            WRITE (6,125) NUM,KEY,WORD,SIGNALTY
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'XLABE') THEN
            READ (99,125) NUM,KEY,WORD,XLABEL
            WRITE (6,125) NUM,KEY,WORD,XLABEL
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'YLABE') THEN
            READ (99,125) NUM,KEY,WORD,YLABEL
            WRITE (6,125) NUM,KEY,WORD,YLABEL
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'COMME') THEN
            READ (99,125) NUM,KEY,WORD,YLABEL
            WRITE (6,125) NUM,KEY,WORD,YLABEL
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'BEAMK') THEN
            READ (99,130) NUM,KEY,WORD,BEAMKV
            WRITE (6,1301) NUM,KEY,WORD,BEAMKV
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'EMISS') THEN
            READ (99,130) NUM,KEY,WORD,EMISSION
            WRITE(6,1301) NUM,KEY,WORD,EMISSION
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'PROBE') THEN
            READ (99,130) NUM,KEY,WORD,PROBECUR
            WRITE (6,1301) NUM,KEY,WORD,PROBECUR
            FKEY='T'
```

```
      GO TO 129
      ENDIF
IF(KEY.EQ.'BEAMD') THEN
      READ (99,130) NUM,KEY,WORD,BEAMDIAM
      WRITE (6,1301) NUM,KEY,WORD,BEAMDIAM
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'MAGCA') THEN
      READ (99,130) NUM,KEY,WORD,MAGCAM
      WRITE (6,1301) NUM,KEY,WORD,MAGCAM
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'CONVA') THEN
      READ (99,130) NUM,KEY,WORD,CONVANGLE
      WRITE (6,1301) NUM,KEY,WORD,CONVANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'COLLA') THEN
      READ (99,130) NUM,KEY,WORD,COLLANGLE
      WRITE (6,1301) NUM,KEY,WORD,COLLANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'OPERM') THEN
      READ (99,125) NUM,KEY,WORD,OPERMODE
      WRITE(6,125) NUM,KEY,WORD,OPERMODE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'THICK') THEN
      READ (99,130) NUM,KEY,WORD,THICKNESS
      WRITE(6,1301) NUM,KEY,WORD,THICKNESS
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'XTILT') THEN
      READ (99,130) NUM,KEY,WORD,XTILTSTGE
      WRITE(6,1301) NUM,KEY,WORD,XTILTSTGE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'YTILT') THEN
      READ (99,130) NUM,KEY,WORD,YTILTSTGE
      WRITE(6,1301) NUM,KEY,WORD,YTILTSTGE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'XPOSI') THEN
      READ (99,130) NUM,KEY,WORD,XPOSITION
      WRITE(6,1301) NUM,KEY,WORD,XPOSITION
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'YPOSI') THEN
      READ (99,130) NUM,KEY,WORD,YPOSITION
      WRITE(6,1301) NUM,KEY,WORD,YPOSITION
      FKEY='T'
      GO TO 129
```

```
      ENDIF
IF(KEY.EQ.'ZPOSI') THEN
      READ (99,130) NUM,KEY,WORD,ZPOSITION
      WRITE (6,1301) NUM,KEY,WORD,ZPOSITION
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'DWELL') THEN
      READ (99,130) NUM,KEY,WORD,DWELLTIME
      WRITE(6,1301) NUM,KEY,WORD,DWELLTIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'INTEG') THEN
      READ (99,130) NUM,KEY,WORD,INTEGTIME
      WRITE(6,1301) NUM,KEY,WORD,INTEGTIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'ELSDE') THEN
      READ (99,125) NUM,KEY,WORD,ELSDET
      WRITE (6,125) NUM,KEY,WORD,ELSDET
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'ELEVA') THEN
      READ (99,130) NUM,KEY,WORD,ELEVANGLE
      WRITE(6,1301) NUM,KEY,WORD,ELEVANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'AZIMA') THEN
      READ (99,130) NUM,KEY,WORD,AZIMANGLE
      WRITE(6,1301) NUM,KEY,WORD,AZIMANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'SOLID') THEN
      READ (99,130) NUM,KEY,WORD,SOLIDANGLE
      WRITE(6,1301) NUM,KEY,WORD,SOLIDANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'LIVET') THEN
      READ (99,130) NUM,KEY,WORD,LIVETIME
      WRITE(6,1301) NUM,KEY,WORD,LIVETIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'REALT') THEN
      READ (99,130) NUM,KEY,WORD,REALTIME
      WRITE (6,1301) NUM,KEY,WORD,REALTIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'TBEWI') THEN
      READ (99,130) NUM,KEY,WORD,TBEWIND
      WRITE(6,1301) NUM,KEY,WORD,TBEWIND
      FKEY='T'
      GO TO 129
      ENDIF
```

```
      IF(KEY.EQ.'TAUWI') THEN
            READ (99,130) NUM,KEY,WORD,TAUWIND
            WRITE (6,1301) NUM,KEY,WORD,TAUWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TDEAD') THEN
            READ (99,130) NUM,KEY,WORD,TDEADLYR
            WRITE (6,1301) NUM,KEY,WORD,TDEADLYR
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TACTL') THEN
            READ (99,130) NUM,KEY,WORD,TACTLYR
            WRITE (6,1301) NUM,KEY,WORD,TACTLYR
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TALWI') THEN
            READ (99,130) NUM,KEY,WORD,TALWIND
            WRITE(6,1301) NUM,KEY,WORD,TALWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TPYWI') THEN
            READ (99,130) NUM,KEY,WORD,TPYWIND
            WRITE(6,1301) NUM,KEY,WORD,TPYWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TBNWI') THEN
            READ (99,130) NUM,KEY,WORD,TBNWIND
            WRITE (6,1301) NUM,KEY,WORD,TBNWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TDIWI') THEN
            READ (99,130) NUM,KEY,WORD,TDIWIND
            WRITE(6,1301) NUM,KEY,WORD,TDIWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'THCWI') THEN
            READ (99,130) NUM,KEY,WORD,THCWIND
            WRITE (6,1301) NUM,KEY,WORD,THCWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'EDSDE') THEN
            READ (99,125) NUM,KEY,WORD,EDSDET
            WRITE(6,125) NUM,KEY,WORD,EDSDET
            FKEY='T'
            GO TO 129
            ENDIF
c
c Notice if the keyword is not defined above it is ignored!!!!
c now we will write it out anyway to show the user what it was
c
c       IF (FKEY.EQ.'F') THEN
            READ (99,125) NUM,KEY,WORD,KTEXT
            WRITE (6,1251) NUM,KEY,WORD,NUM,KEY,WORD,KTEXT
```

```
1251           FORMAT (' !!!! Warning Unknown or User keyword = ',
     *         1A1,1A5,1A9,/,1X,1A1,1A5,1A9,64A1)
c              endif
c now that we have decoded the keyword and its value
c we start over  with the loop until the keyword = SPECT
c

      GO TO 129
C
C We have found the keyword spectrum begin to read the data now.
c
1000   CONTINUE

       WRITE(6,1100)
1100    FORMAT(1X,14('='),'Reading Spectral Data',14('='))
c
c      calculate number of lines of data in the file
c
      NLINES=INT(NPOINTS/NCOLUMNS)
      NCOLS=INT(NCOLUMNS)
      NPTS=INT(NPOINTS)
c
c      Begin reading in data from file
c      Check the data type is it X,Y or Y
c
      IF(DATATYPE(1).EQ.'Y') GO TO 2000
      IF(DATATYPE(1).EQ.'X') GO TO 3000
      WRITE (6,*) 'DATATYPE ERROR'
c
c      if the datatype is not X,Y or Y then we have an error
c
      GO TO 991
c
c      this area reads Y data type
c
2000   CONTINUE
c
c
c
      DO  2010 I=1,NLINES
      READ (99,*,ERR=992) (YVALUE(J), J=1,NCOLS)
d       WRITE (6,2011) (YVALUE(J), J=1,NCOLS)
d2011    FORMAT (1X,10F15.0)
2020   FORMAT (1F15.5)
c
c      now transfer this to the X&Y data arrays
c      remember since this is Y axis data only we have to
c      calcuated the X axis values using the parameters
c      passed by the header
c
      DO 2030 J=1,NCOLS
      ICHN = (I-1)*NCOLS + J
      YADATA(ICHN)=YVALUE(J)
      XADATA(ICHN)= OFFSET + ICHN*XPERCHAN
2030   CONTINUE
2010   CONTINUE
C
C
C
c
c       now check to see that all data has been read
```

```
c       is NCOLS*NLINES=NPOINTS?
c
        ICHN=NCOLS*NLINES
        IF (ICHN.LT.NPOINTS) THEN
          READ (99,*,ERR=992) (YADATA(K+ICHN),K=1,NPOINTS-ICHN)
            DO 2040    K=1,NPOINTS-ICHN
2040        XADATA(ICHN+K)= OFFSET + (ICHN+K)*XPERCHAN
        ENDIF

c
c       all the data should now be in the arrays
c       check for end of file marker
c
       GO TO 5000
c
c       This area reads (X,Y) data pairs
c
3000    CONTINUE
C
C
        DO 3010 I=1,NLINES
        READ (99,*,ERR=992) (XVALUE(K),YVALUE(K),K=1,NCOLS)
d       WRITE (6,3021) (XVALUE(K),YVALUE(K),K=1,NCOLS)
d3021   FORMAT (1X,2(F15.5,1X))
3020    FORMAT(2F15.0)
c
c       Now transfer from the temporary array to the permanent ones
c
       DO 3030 J=1,NCOLS
       ICHN = (I-1)*NCOLS + J
       YADATA(ICHN)=YVALUE(J)
       XADATA(ICHN)=XVALUE(J)
3030    CONTINUE
3010    CONTINUE
c
c       now check to see that all data has been read
c       is NCOLS*NLINES=NPOINTS?
c
        ICHN=NCOLS*NLINES
        IF (ICHN.LT.NPOINTS) THEN
            READ (99,*,ERR=992)
     *        (XADATA(K+ICHN),YADATA(K+ICHN),K=1,NPOINTS-ICHN)
           ENDIF
c
c       Calculate new offset values to go with this data
c
      XPERCHAN=XADATA(2)-XADATA(1)
      OFFSET=XADATA(1)-XPERCHAN
c
c       all the data should now be in the arrays
c       check for end of file marker
c
5000    CONTINUE

       READ (99,125) NUM,KEY,WORD,KTEXT
       WRITE (6,125) NUM,KEY,WORD,KTEXT
       IF(KEY.NE.'ENDOF') GO TO 991
        WRITE (6,5001)
5001     FORMAT(1X,14('='),'Data  Input  Complete',14('='))
        CLOSE (UNIT=99)
c
```

```
c      we found the correct end of file ask the
c      user if they want to write the data on the screen
c       the default will be YES
c
       WRITE(6,5010)
5010   FORMAT(
     *' Do you want to print the data on your screen [Y,N]?=>')
       READ (5,130) ANS
       IF ((ANS.EQ.'N').OR.(ANS.EQ.'n')) GO TO 6000
c
c      user wants to print out the data file
c
       WRITE (6,5015)
5015   FORMAT('   X-AXIS DATA     Y-AXIS DATA'
     *        ,/,31('-'),/)
       DO 5020 I=1,INT(NPOINTS)
       WRITE (6,5030) XADATA(I),YADATA(I)
5030   FORMAT(1X,1G12.5,4X,1G12.6)
5020   CONTINUE
6000   RETURN
C
C
C
991    CONTINUE
C
C There was a problem in a defined keyword or too many keywords
c
       WRITE (6,9911) NUM,KEY,WORD,NUM,KEY,WORD,KTEXT
9911   FORMAT (//,' WARNING: Problem in Keyword =',1A1,1A5,1A9,
     *         /, ' CHECK YOUR DATA FILE AT THE FOLLOWING LINE',
     *         /,1X,1A1,1A5,1A9,64A1)
       RETURN
992    CONTINUE
C
C      There was a problem with the data
c
       WRITE (6,9922) ICHN,XADATA(ICHN),YADATA(ICHN)
9922   FORMAT (//,
     *  ' WARNING: Problem with spectral data AFTER point #= ',I5,//,
     *  ' Values of last good X&Y DATA are: ',1x,2(1G12.6,','),/,
     *  ' Check your data file',//,' SPECTRAL INPUT TERMINATED',//)
       RETURN
       END


C=======================================================================
           Subroutine WREMMFF
C=======================================================================
c
c
c This subroutine WRITES the EMSA/MAS Spectrum File Format.
c The DATA to be written to the file are stored locally in predefined
c arrays. These arrays are passed directly from the main
c program and can be used to send data to the file as needed.
c In this implementation the arrays are passed in labeled common
c blocks. They can alternatively be passed as subroutine
c arguments.
c
c REVISION HISTORY
c
c Version Date Initials Reason
```

```
c-----------------------------------------------------------------------
c
c 1.0  1/18/91 NJZ   Creation of the World WARNING THIS VERSION
c                    Case Sensitive!!! Until DeBugged
c
c-----------------------------------------------------------------------
c Definitions of Arrays used in this Subroutine
c
c
c
c XADATA  = Real Array <= 4096 values containing X-Axis Data
c YADATA  = Real Array <= 4096 values containing Y-Axis Data
c EXPARSPT = Real Array <=20 values containing Expt Parameters of Spectrum
c EXPARMSC = Real Array <=20 values containing Expt Parameters of Microscope
c EXPARSAM = Real Array <=20 values containing Expt Parameters of Sample
c EXPAREDS = Real Array <=20 values containing Expt Parameters of EDS
c EXPARELS = Real Array <=20 values containing Expt Parameters of ELS
c EXPARAES = Real Array <=20 values containing Expt Parameters of AES
c EXPARWDS = Real Array <=20 values containing Expt Parameters of WDS
c EXPARPES = Real Array <=20 values containing Expt Parameters of PES
c EXPARXRF = Real Array <=20 values containing Expt Parameters of XRF
c EXPARCLS = Real Array <=20 values containing Expt Parameters of CLS
c EXPARGAM = Real Array <=20 values containing Expt Parameters of GAM
c SIGNALTY = Byte Array of 2 characters describing the signal type
c DATATYPE = Byte Array of 3 characters describing the format
c OPERMODE = Byte Array of 5 characters describing
c          the instrument operating mode
c EDSDET  = Byte Array of 6 characters describing the EDS detector type
c ELSDET  = Byte Array of 6 characters describing the ELS detector type
c XUNITS  = Byte Array <= 64 characters describing the X-Axis Data units
c YUNITS  = Byte Array <= 64 characters describing the Y-Axis Data units
c XLABEL  = Byte Array <= 64 characters describing the X-Axis Data label
c YLABEL  = Byte Array <= 64 characters describing the Y-Axis Data label
c FFORMAT = Byte Array <= 64 characters with the File Format Title
c TITLE   = Byte Array <= 64 characters with a spectrum Title
c DATE    = Byte Array of 12 characters with the date of acquisition
c          in the form DD-MMM-YYYY
c TIME    = Byte Array of 5 characters with the time of acquisition
c          in the form HH:MM
c OWNER   = Byte Array <= 64 characters with the owner/analysts name
c COMMENT = Byte Array <= 64 Character*1 used to hold a text comment string
c
c-----------------------------------------------------------------------
c
c Current Definitions of EXPAR (Experimental Parameter Array) Values
c
c-----------------------------------------------------------------------
c Parameters relating to the Spectrum Characteristics = EXPARSPT
c-----------------------------------------------------------------------
c
c Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
c Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),XLABEL(64),YLABEL(64),DATATYPE(2)
c
c EXPARSPT(1) = c#VERSION = File Format Version Number
c EXPARSPT(2) = c#NPOINTS = Total Number of Data Points in X&Y Data Arrays
c        1 <= NPOINTS <= 4096
c EXPARSPT(3) = c#NCOLUMNS = Number of columns of data
c        1 <= NCOLUMNS <= 5
c EXPARSPT(4) = c#XPERCHAN = Increment of X-axis units per channel (eV/Channel)
c          This is only useful if (x,y) paired data are not provided
c EXPARSPT(5) = c#OFFSET  = Energy value of first data point in eV
```

```
c          This is only useful if (x,y) paired data are not provided
c EXPARSPT(6) = c#CHOFFSET = Channel number which corresponds to zero units along
c           x-axis, this may be either a positive or negative value
C SIGNALTY(.) = c#SIGNALTYPE = Type of Spectroscopy
c       EDS = Energy Dispersive Spectroscopy
c       WDS = Wavelength Dispersive Spectroscopy
c       ELS = Energy Loss Spectroscopy
c       AES = Auger Electron Spectroscopy
c       PES = Photo Electron Spectroscopy
c       XRF = X-ray Fluorescence Spectroscopy
c       CLS = Cathodoluminescence Spectroscopy
c       GAM = Gamma Ray Spectroscopy
c XUNITS(.)  = c#XUNITS = Up to 64 characters describing the X-Axis Data units
c YUNITS(.)  = c#YUNITS = Up to 64 characters describing the Y-Axis Data units
c XLABEL(.)  = c#XLABEL = Up to 64 characters describing the X-Axis Data label
c YLABEL(.)  = c#YLABEL = Up to 64 characters describing the Y-Axis Data label
C DATATYPE(.) = c#DATATYPE = Type of data format
c          Y = Spectrum Y axis data only, X-axis data to be calculated
c           using XPERCHAN and OFFSET and the following formulae
c            X = OFFSET + CHANNEL*XPERCHAN
c          XY = Spectral data is in the form of XY pairs
c
c
c--------------------------------------------------------------------------
c Microscope/Microanalysis Instrument Parameters = EXPARMSC
c--------------------------------------------------------------------------
c
c Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,
c CONVANGLE,COLLANGLE
c Character*1 OPERMODE(5)
c
c EXPARMSC(1) = c#BEAMKV   = Accelerating Voltage of Instrument in kV
c EXPARMSC(2) = c#EMISSION  = Gun Emission current in microAmps
c EXPARMSC(3) = c#PROBECUR  = Probe current in nanoAmps
c EXPARMSC(4) = c#BEAMDIAM  = Diameter of incident probe in nanometers
c EXPARMSC(5) = c#MAGCAM  = Magnification or Camera Length (Mag in x, Cl in mm)
c EXPARMSC(6) = c#CONVANGLE = Convergence semi-angle of incident beam in milliRadians
c EXPARMSC(7) = c#COLLANGLE = Collection semi-angle of scattered beam in milliRad
c OPERMODE(.) = c#OPERMODE  = Operating Mode of Instrument
c                       IMAGE = Imaging Mode
c                       DIFFR = Diffraction Mode
c                       SCIMG = Scanning Imaging Mode
c                       SCDIF = Scanning Diffraction Mode
c
c--------------------------------------------------------------------------
c Experimental Parameters relating to the Sample = EXPARSAM
c--------------------------------------------------------------------------
c
c Real EXPARSAM(20),THICKNESS,XTILTSTAGE,YTILTSTAGE,XPOSITION,YPOSITION,
C     ZPOSITION
c
c EXPARSAM(1) = c#THICKNESS = Specimen thickness in nanometers
c EXPARSAM(2) = c#XTILTSTAGE = Specimen stage tilt X-axis in degrees
c EXPARSAM(3) = c#YTILTSTAGE = Specimen stage tilt Y-axis in degrees
c EXPARSAM(4) = c#XPOSITION = X location of beam or specimen
c EXPARSAM(5) = c#YPOSITION = Y location of beam or specimen
c EXPARSAM(6) = c#ZPOSITION = Z location of beam or specimen
c
c
c--------------------------------------------------------------------------
c Experimental Parameters relating mainly to ELS = EXPARELS
```

```
c-----------------------------------------------------------------------
c
c Real EXPARELS(20),DWELLTIME,INTEGTIME
c Character*1 ELSDET(6)
c
c EXPARELS(1) = c#DWELLTIME = Dwell time per channel for serial data collection in msec
c EXPARELS(2) = c#INTEGTIME = Integration time per spectrum for parallel data collection
c           in milliseconds
c ELSDET(.) = c#ELSDET  = Type of ELS Detector
c           Serial = Serial ELS Detector
c           Parall = Parallel ELS Detector
c
c-----------------------------------------------------------------------
c Experimental Parameters relating mainly to EDS = EXPAREDS
c-----------------------------------------------------------------------
c
c Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
c Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
c Character*1 EDSDET(6)
c
c EXPAREDS(1) = c#ELEVANGLE = Elevation angle of EDS,WDS detector in degrees
c EXPAREDS(2) = c#AZIMANGLE = Azimuthal angle of EDS,WDS detector in degrees
c EXPAREDS(3) = c#SOLIDANGLE = Collection solid angle of detector in sR
c EXPAREDS(4) = c#LIVETIME = Signal Processor Active (Live) time in seconds
c EXPAREDS(5) = c#REALTIME = Total clock time used to record the spectrum in seconds
c EXPAREDS(6) = c#TBEWIND = Thickness of Be Window on detector in cm
c EXPAREDS(7) = c#TAUWIND = Thickness of Au Window/Electrical Contact in cm
c EXPAREDS(8) = c#TDEADLYR = Thickness of Dead Layer in cm
c EXPAREDS(9) = c#TACTLYR = Thickness of Active Layer in cm
c EXPAREDS(10) = c#TALWIND = Thickness of Aluminium Window in cm
c EXPAREDS(11) = c#TPYWIND = Thickness of Pyrolene Window in cm
c EXPAREDS(12) = c#TBNWIND = Thickness of Boron-Nitride Window in cm
c EXPAREDS(13) = c#TDIWIND = Thickness of Diamond Window in cm
c EXPAREDS(14) = c#THCWIND = Thickness of HydroCarbon Window in cm
c EDSDET(.)   = c#EDSDET = Type of X-ray Detector
c       SIBEW = Si(Li) with Be Window
c       SIUTW = Si(Li) with Ultra Thin Window
c       SIWLS = Si(Li) Windowless
c       GEBEW = Ge with Be Window
c       GEUTW = Ge with Ultra Thin Window
c       GEWLS = Ge Windowless
c
c-----------------------------------------------------------------------
c Experimental Parameters relating mainly to WDS = EXPARWDS
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c Experimental Parameters relating mainly to XRF = EXPARXRF
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c Experimental Parameters relating mainly to AES = EXPARAES
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
```

```
c Experimental Parameters relating mainly to PES = EXPARPES
c-------------------------------------------------------------------------
c
c     Nothing currently defined
c
c-------------------------------------------------------------------------
c Experimental Parameters relating mainly to CLS = EXPARCLS
c-------------------------------------------------------------------------
c
c     Nothing currently defined
c
c-------------------------------------------------------------------------
c Experimental Parameters relating mainly to GAM = EXPARGAM
c-------------------------------------------------------------------------
c
c     Nothing currently defined
c
c=========================================================================
c       END OF DEFINITIONS
c=========================================================================
c
c
c Start Code
c
c
c  Dimension all arrays, & Declare Variable Types
c  REMEMBER we have no integer parameters!!!!!
c
c
            Character*1 TITLE(64),DATE(12),TIME(5),OWNER(64),COMMENT(64)
            Real XADATA(4096),YADATA(4096)
            Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
            Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),
     *      XLABEL(64),YLABEL(64),DATATYPE(2)
            Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,CONVANGLE,
     *     COLLANGLE
            Character*1 OPERMODE(5)
            Real EXPARSAM(20),THICKNESS,XTILTSTAGE,YTILTSTAGE,XPOSITION,YPOSITION,
     *     ZPOSITION
            Real EXPARELS(20),DWELLTIME,INTEGTIME
            Character*1 ELSDET(6)
            Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
            Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
            Character*1 EDSDET(6)
            Real EXPARWDS(20),EXPARXRF(20),EXPARAES(20),EXPARPES(20),EXPARCLS(20),
     *      EXPARGAM(20)
c-------------------------------------------------------------------------------
c Set up all Variable arrays to be passed in labeled common block for ease of
c passing parameters between different subroutines
c-------------------------------------------------------------------------------
c
            Common /XYDATA/ XADATA,YADATA
            Common /EXPPAR/ EXPARSPT,EXPARMSC,EXPARSAM,EXPARELS,EXPAREDS,EXPARWDS,
     *  EXPARAES,EXPARPES,EXPARXRF,EXPARCLS,EXPARGAM
            Common /TEXT/ TITLE,DATE,TIME,OWNER,COMMENT,
     *  SIGNALTY,XUNITS,YUNITS,XLABEL,YLABEL,DATATYPE,
     *  OPERMODE,ELSDET,EDSDET
c
            Character*1 ANS,NUM,FKEY
            Character*3 STYPE
            Logical *1 TF
```

```
            Character*5 KEY
            Character*9 WORD
           EQUIVALENCE ( EXPARSPT(1),VERSION)
           EQUIVALENCE ( EXPARSPT(2),NPOINTS)
           EQUIVALENCE ( EXPARSPT(3),NCOLUMNS)
           EQUIVALENCE ( EXPARSPT(4),XPERCHAN)
           EQUIVALENCE ( EXPARSPT(5),OFFSET)
           EQUIVALENCE ( EXPARSPT(6),CHOFFSET)
           EQUIVALENCE ( EXPARMSC(1),BEAMKV)
           EQUIVALENCE ( EXPARMSC(2),EMISSION)
           EQUIVALENCE ( EXPARMSC(3),PROBECUR)
           EQUIVALENCE ( EXPARMSC(4),BEAMDIAM)
           EQUIVALENCE ( EXPARMSC(5),MAGCAM)
           EQUIVALENCE ( EXPARMSC(6),CONVANGLE)
           EQUIVALENCE ( EXPARMSC(7),COLLANGLE)
           EQUIVALENCE ( EXPARSAM(1),THICKNESS)
           EQUIVALENCE ( EXPARSAM(2),XTILTSTAGE)
           EQUIVALENCE ( EXPARSAM(3),YTILTSTAGE)
           EQUIVALENCE ( EXPARSAM(4),XPOSITION)
           EQUIVALENCE ( EXPARSAM(5),YPOSITION)
           EQUIVALENCE ( EXPARSAM(6),ZPOSITION)
           EQUIVALENCE ( EXPARELS(1),DWELLTIME)
           EQUIVALENCE ( EXPARELS(2),INTEGTIME)
           EQUIVALENCE ( EXPAREDS(1),ELEVANGLE)
           EQUIVALENCE ( EXPAREDS(2),AZIMANGLE)
           EQUIVALENCE ( EXPAREDS(3),SOLIDANGLE)
           EQUIVALENCE ( EXPAREDS(4),LIVETIME)
           EQUIVALENCE ( EXPAREDS(5),REALTIME)
           EQUIVALENCE ( EXPAREDS(6),TBEWIND)
           EQUIVALENCE ( EXPAREDS(7),TAUWIND)
           EQUIVALENCE ( EXPAREDS(8),TDEADLYR)
           EQUIVALENCE ( EXPAREDS(9),TACTLYR)
           EQUIVALENCE ( EXPAREDS(10),TALWIND)
           EQUIVALENCE ( EXPAREDS(11),TPYWIND)
           EQUIVALENCE ( EXPAREDS(12),TBNWIND)
           EQUIVALENCE ( EXPAREDS(13),TDIWIND)
           EQUIVALENCE ( EXPAREDS(14),THCWIND)
c
c
c    NOW define different arrays used only in this routine
c
c    XVALUE & YVALUE are temporary arrays used for reading in values
c    NCOL is a temp parameter for reading in values
c
           REAL XVALUE(10),YVALUE(10)
           REAL NCOL
c
c
c The array FNAME = Name of File to be read, Usually null terminated
c so we will initialize it with zero's using a Do loop so that successive
c operations of the routine within a single program donot cause problems
c if a short filename is used.
c
c FNAME  = Byte array of 17 characters containing the filename
c KText = Byte array of 64 characters containing Keyword text
c
c
           Character*17 FNAME
           Character*1 KTEXT(64)
C
C      Initialize some more arrays only used in this routine
```

```
c
          Integer*2  IYEAR,IMON,IDAY,IHOUR,IMIN,ISEC,IHSEC
          Character*3 Month (12)
          Data Month /'Jan','Feb','Mar','Apr','May','Jun','Jul','Aug',
     *     'Sep','Oct','Nov','Dec'/
c
c Initialize Filename
c
c
c
c
c Begin by asking the user to supply a file name
c Note the following statement is generally compiler dependant
c Note this version limits file name to 16 characters
c
c
100     WRITE (6,101)
101   FORMAT (' Enter name of OUTPUT file [DEV:FILENAME.EXT]=> ')
          READ (5,102) FNAME
102   FORMAT (1A16)
C
C     EXIT IF BLANK FILENAME IS PROVIDED BY THE USER
C
          IF (FNAME.EQ.' ') RETURN
c
c Does the file already exist?
c TF is a logical parameter True => files exists
c                    False => files dne
c
c
c     check that the file exists
c
          INQUIRE (FILE=FNAME,EXIST=TF)
          IF (TF) THEN
c
c         Replacing a file!!! Warn user.
c
                    Write (6,*)
     *    'WARNING: Replacing Existing File Continue? [Y,N]: '
                    Read (6,123)  ANS
                    IF (ANS.EQ.'y') ANS='Y'
                    IF (ANS.NE.'Y') GO TO 100
                    ENDIF
c
c Now get some statistics and check for possible errors
c
120   FORMAT (1A45)
c
c     List the current file parameters for data output
c     and allow an update if desired.
c     But first check on a few settings to make sure they
c     are okay.
c
          IF (NPOINTS.LE.0) THEN
                    WRITE (6,*) 'ERROR: Number of points in the data set is ZERO'
                    RETURN
                    ENDIF
          IF (NCOLUMNS.EQ.0) NCOLUMNS=1.
          IF (NCOLUMNS.GT.5) NCOLUMNS=5.
          IF ((DATATYPE(1).NE.'Y').AND.(DATATYPE(1).NE.'X')) THEN
                    DATATYPE(1)='Y'
```

```
                DATATYPE(2)=' '
                ENDIF
c
c    IF DATAYPE IS XY THEN ONLY ALLOW MAXIMUM OF 3 COLUMNS!!
C
          IF ((DATATYPE(1).EQ.'X').AND.(NCOLUMNS.GT.3)) NCOLUMNS=3.
          WRITE (6,103) Datatype,Npoints,Ncolumns,Npoints/Ncolumns
103    FORMAT (24('='),/,' Current File Parameters',/,24('='),/,
   *       ' DataType = ',2A1,/,
   *       ' Points   = ',1f7.2,/,
   *       ' Columns  = ',1f7.2,/,
   *       ' Lines    = ',1f7.2,/)
c
c    Check to see if user wants a last minute change in
c    output format (x,y) or ncolumns
c
1221  Write (6,122) Datatype
122   FORMAT (' Enter the DataType Y or XY [Default = ',2a1,'] ')
          Read (6,123) ANS
123   FORMAT (1A1)
          IF (ANS.NE.' ') THEN
                IF ((ANS.EQ.'X').OR.(ANS.EQ.'x')) THEN
                      DATATYPE(1)='X'
                      DATATYPE(2)='Y'
                      GO TO 1231
                      ENDIF
                IF ((ANS.EQ.'Y').OR.(ANS.EQ.'y')) THEN
                      DATATYPE(1)='Y'
                      DATATYPE(2)=' '
                      GO TO 1231
                      ENDIF
c
c        Not X or Y then we must have an input error reenter data
c
                GO TO 1221
                ENDIF
1231  CONTINUE
c
c    make sure that Ncolumns is compatible with XY or Y Formats
c
c    IF DATAYPE IS XY THEN ONLY ALLOW MAXIMUM OF 3 COLUMNS!!
C
          IF ((DATATYPE(1).EQ.'X').AND.(NCOLUMNS.GT.3)) NCOLUMNS=3.

105   IF (DATATYPE(1).EQ.'Y') Write (6,121) Ncolumns
          IF (DATATYPE(1).EQ.'X') Write (6,1211) Ncolumns
121   FORMAT (' Enter the number of COLUMNS 1-5 ',
    * 'in the File [Default=',1f2.0,'] ')
1211   FORMAT (' Enter the number of COLUMNS 1-3 ',
    * 'in the File [Default=',1f2.0,'] ')
          READ(6, 104) NCOL
104    FORMAT (1F5.0)
          IF (NCOL.NE.0) THEN
                IF (NCOL.GT.5) GO TO 105
                IF ((NCOL.GT.3).AND.(DATATYPE(1).EQ.'X')) GO TO 105
                NCOLUMNS=NCOL
                ENDIF
C
C GET THE DATE AND TIME
C
          CALL GETDAT (IYEAR,IMON,IDAY)
```

```
          CALL GETTIM (IHOUR,IMIN,ISEC,IHSEC)
c
c         Now that everything is okay open the file
c
          OPEN (UNIT=98,BLANK='NULL',ACCESS='SEQUENTIAL',FILE=FNAME)
c
c    Ask the user if full header or partial header is desired
c
4001  WRITE (6,4000)
4000  FORMAT (
     * ' Select the type of Information Header for your file:',//,
     * 10X,' B = Basic EMSA/MAS Header Information Only',/,
     * 10x,' F = Full  EMSA/MAS Header',/,
     * 10x,' P = Partial Headers based upon Current Signal Type',/,
     * 10x,'     [Includes Basic,Specimen & Instrument Headers]',/,
     * 10x,' ==> ')
          READ (6,123) ANS
          IF (ANS.EQ.' ') THEN
            GO TO 4001
            ELSE
              IF (ANS.EQ.'p') ANS = 'P'
              IF (ANS.EQ.'b') ANS = 'B'
              IF (ANS.EQ.'f') ANS = 'F'
              IF (((ANS.EQ.'F').OR.(ANS.EQ.'B')).OR.(ANS.EQ.'P')) THEN
                              GO TO 4002
                              ELSE
                              GO TO 4001
                              ENDIF
            ENDIF
4002  CONTINUE
c=====================================================================
c
c Now lets write the required (BASIC) parameters as defined by EMSA/MAS
c
c=====================================================================
c
c First WRITE the Format TYPE
c
          CALL TWRITE ('#FORMAT      : ',
     *          'EMSA/MAS SPECTRAL DATA STANDARD',31,'Y')
125   FORMAT (1A46)
126   FORMAT (1A15,64A1)
1261  FORMAT (1X,1A15,64A1)
c
c now the version
c
          CALL VWRITE ('#VERSION     : ',VERSION,'Y')
130   FORMAT(1A15,1G12.5)
1301  FORMAT (1X,1A15,1G12.6)
c
c now the Title BUT give user option to change it.
c
            CALL TMODIFY(  '#TITLE       : ',TITLE,64)
            CALL TWRITE ( '#TITLE       : ',TITLE,64,'Y')
c
c now the Date
c
          WRITE (98,1262) '#DATE        : ',IDAY,'-',MONTH(IMON),'-',IYEAR
          WRITE (6,1263)  '#DATE        : ',IDAY,'-',MONTH(IMON),'-',IYEAR
1262  FORMAT(1A15,I2.2,1A1,1A3,1A1,I4)
1263  FORMAT(1X,1A15,I2.2,1A1,1A3,1A1,I4)
```

```
c
c
c now the Time
c
            WRITE (98,1271) '#TIME        : ',IHOUR,':',IMIN,':',ISEC
            WRITE (6,1272)  '#TIME        : ',IHOUR,':',IMIN,':',ISEC
1271        FORMAT(1A15,I2.2,1A1,I2.2,1A1,I2.2)
1272        FORMAT(1X,1A15,I2.2,1A1,I2.2,1A1,I2.2)
C
C
c now the Owner
c
            CALL TWRITE ( '#OWNER       : ',OWNER,64,'Y')

c
c now the Npoints
c
            CALL VWRITE ( '#NPOINTS     : ',NPOINTS,'Y')
c
c now the NColumns
c
            CALL VWRITE (  '#NCOLUMNS    : ',NCOLUMNS,'Y')
c
c now the XUnits
c
            CALL TWRITE ( '#XUNITS      : ',XUNITS,64,'Y')
c
c now the YUnits
c
            CALL TWRITE ( '#YUNITS      : ',YUNITS,64,'Y')
c
c now the DATATYPE
c
            CALL TWRITE ( '#DATATYPE    : ',DATATYPE,2,'Y')
c
c now the XPERCHAN
c
            CALL VWRITE ( '#XPERCHAN    : ',XPERCHAN,'Y')
c
c now the OFFSET
c
             CALL VWRITE ( '#OFFSET      : ',OFFSET,'Y')
c
c We now have the minimum data required
c begin checking individual key words
c based upon user selected option of FULL or Partial Headers
c as well as the Current Signal Type.
c
c Check the next keyword
c write it out if of its value exists and
c
c
            IF (ANS.EQ.'B') WRITE (6,*) 'Basic Header Requested'
            IF (ANS.EQ.'F') WRITE (6,*) 'Full Header Requested'
            IF (ANS.EQ.'P') WRITE (6,*) 'Partial Header Requested'
c now test each predefined keyword
c
c     concatenate the signaltype characters into a
c     single character constant for ease of use below
c
            STYPE =SIGNALTY(1)//SIGNALTY(2)//SIGNALTY(3)
```

```
c      write (6,*) 'signaltype = ',stype
c
c now for modification of each predefined keyword
c
          IF ((ANS.EQ.'F').OR.(ANS.EQ.'P')) THEN
                    CALL VWRITE ( '#CHOFFSET    : ',CHOFFSET,'Y')
                    CALL TWRITE ( '#SIGNALTYPE  : ',SIGNALTY,3,'Y')
                    CALL TWRITE ( '#XLABEL      : ',XLABEL,64,'Y')
                    CALL TWRITE ( '#YLABEL      : ',YLABEL,64,'Y')
                    CALL VWRITE ( '#BEAMKV   -kV: ',BEAMKV,'Y')
                    CALL VWRITE ( '#EMISSION -uA: ',EMISSION,'Y')
                    CALL VWRITE ( '#PROBECUR -nA: ',PROBECUR,'Y')
                    CALL VWRITE ( '#BEAMDIAM -nm: ',BEAMDIAM,'Y')
                    CALL VWRITE ( '#MAGCAM      : ',MAGCAM,'Y')
                    CALL VWRITE ( '#CONVANGLE-mR: ',CONVANGLE,'Y')
                    CALL VWRITE ( '#COLLANGLE-mR: ',COLLANGLE,'Y')
                    CALL TWRITE ( '#OPERMODE    : ',OPERMODE,5,'Y')
                    CALL VWRITE ( '#THICKNESS-nm: ',THICKNESS,'Y')
                    CALL VWRITE ( '#XTILTSTAGE  : ',XTILTSTGE,'Y')
                    CALL VWRITE ( '#YTILTSTAGE  : ',YTILTSTGE,'Y')
                    CALL VWRITE ( '#XPOSITION   : ',XPOSITION,'Y')
                    CALL VWRITE ( '#YPOSITION   : ',YPOSITION,'Y')
                    CALL VWRITE ( '#ZPOSITION   : ',ZPOSITION,'Y')
          ENDIF
          IF ((ANS.EQ.'F').OR.((ANS.EQ.'P').AND.(STYPE.EQ.'ELS'))) THEN
C
C     ELS HEADER INFORMATION
C
                    CALL TWRITE ('#ELSDET      : ',ELSDET,6,'Y')
                    CALL VWRITE ('#DWELLTIME-ms: ',DWELLTIME,'Y')
                    CALL VWRITE ('#INTEGTIME-ms: ',INTEGTIME,'Y')
          ENDIF
C
C     NOW CHECK FOR EDS
C
          IF ((ANS.EQ.'F').OR.((ANS.EQ.'P').AND.(STYPE.EQ.'EDS'))) THEN
C
C     EDS INFORMATION
C
                    CALL TWRITE ('#EDSDET      : ',EDSDET,5,'Y')
                    CALL VWRITE ('#ELEVANGLE-dg: ',ELEVANGLE,'Y')
                    CALL VWRITE ('#AZIMANGLE-dg: ',AZIMANGLE,'Y')
                    CALL VWRITE ('#SOLIDANGL-sR: ',SOLIDANGLE,'Y')
                    CALL VWRITE ('#LIVETIME  -s: ',LIVETIME,'Y')
                    CALL VWRITE ('#REALTIME  -s: ',REALTIME,'Y')
                    CALL VWRITE ('#TBEWIND  -cm: ',TBEWIND,'Y')
                    CALL VWRITE ('#TAUWIND  -cm: ',TAUWIND,'Y')
                    CALL VWRITE ('#TDEADLYR -cm: ',TDEADLYR,'Y')
                    CALL VWRITE ('#TACTLYR  -cm: ',TACTLYR,'Y')
                    CALL VWRITE ('#TALWIND  -cm: ',TALWIND,'Y')
                    CALL VWRITE ('#TPYWIND  -cm: ',TPYWIND,'Y')
                    CALL VWRITE ('#TBNWIND  -cm: ',TBNWIND,'Y')
                    CALL VWRITE ('#TDIWIND  -cm: ',TDIWIND,'Y')
                    CALL VWRITE ('#THCWIND  -cm: ',THCWIND,'Y')
          ENDIF
C
C     END OF EMSA/MAS DEFINITIONS
C
          CALL TWRITE ('#SPECTRUM    : ',
     *               'Spectral Data Starts Here',25,'Y')
c
```

```
1000  CONTINUE
c=====================================================================
c
c   Now write the data
c
c
c     calculate number of lines of data in the file
c
            NLINES=INT(NPOINTS/NCOLUMNS)
            NCOLS=INT(NCOLUMNS)
            NPTS=INT(NPOINTS)
C
C     Write info on the screen
            WRITE(6,1100)
1100  FORMAT(1X,14('='),' Writing Spectral Data File ',14('='))
c
c     Begin writing in data to file
c     Check the data type is it X,Y or Y
c
            IF(DATATYPE(1).EQ.'Y') GO TO 2000
            IF(DATATYPE(1).EQ.'X') GO TO 3000
c
c     if the datatype is not X,Y or Y then we have an error
c
            GO TO 991
c
c     this area writes Y data type
c
2000  CONTINUE
c
c
c
            DO 2010 I=1,NLINES
            DO 2011 J=1,NCOLS
            ICHN = (I-1)*NCOLS + J
2011    YVALUE(J)=YADATA(ICHN)
            WRITE(98,2020,ERR=992) (YVALUE(J), J=1,NCOLS)
2020    FORMAT (1X,5(G12.6,:,','))
2010    CONTINUE
C
C       CHECK TO MAKE SURE WE HAVE ALL THE DATA WRITTEN
C       THIS MAY HAPPEN IF NPOINTS/NCOLUMNS IS NOT AN INTEGER
C       A SIMPLE CHECK IS TO COMPARE ICHN WITH NPOINTS IF
C       THEY ARE EQUAL THEN EVERYTHING IS OKAY, OTHERWISE
C       THERE ARE A FEW DATA POINTS LEFT.
C
            IF(ICHN.LT.NPOINTS) THEN
                    DO 2030 I=1,NPOINTS-ICHN
                    K=ICHN+I
                    YVALUE(I)=YADATA(K)
2030            CONTINUE
                    WRITE(98,2020) (YVALUE(J), J=1,NPOINTS-ICHN)
                    ENDIF
c
c     all the data should now be in the arrays
c     check for end of file marker
c
            GO TO 5000
c
c     This area writes (X,Y) data pairs
c
```

```
3000  CONTINUE
C
C
            DO 3010 I=1,NLINES
            DO 3030 J=1,NCOLS
            ICHN = (I-1)*NCOLS + J
            YVALUE(J)=YADATA(ICHN)
            XVALUE(J)=XADATA(ICHN)
3030     CONTINUE
            WRITE(98,3020,ERR=992) (XVALUE(K),YVALUE(K),K=1,NCOLS)
3020  FORMAT (1X,10(G12.6,:,','))
3010   CONTINUE
C
C       CHECK TO MAKE SURE WE HAVE ALL THE DATA WRITTEN
C       THIS MAY HAPPEN IF NPOINTS/NCOLUMNS IS NOT AN INTEGER
C       A SIMPLE CHECK IS TO COMPARE ICHN WITH NPOINTS IF
C       THEY ARE EQUAL THEN EVERYTHING IS OKAY, OTHERWISE
C       THERE ARE A FEW DATA POINTS LEFT.
C
            IF(ICHN.LT.NPOINTS) THEN
                    DO 3040 I=1,NPOINTS-ICHN
                    K=ICHN+I
                    YVALUE(I)=YADATA(K)
                    XVALUE(I)=XADATA(K)
3040        CONTINUE
            WRITE(98,3020,ERR=992) (XVALUE(K),YVALUE(K),K=1,NPOINTS-ICHN)
            ENDIF

c
c     all the data should now be in the arrays
c     Write end of file marker
c
5000  CONTINUE

            CALL TWRITE ( '#ENDOFDATA   : ',' ',1,'Y')
            WRITE (6,5001)
5001   FORMAT(1X,14('='),'Output Complete to Data File',14('='))
            CLOSE (UNIT=98)
c
c     we found the correct end of file ask the
c     user if they want to write the data on the screen
c      the default will be YES
c
            WRITE(6,5010)
5010  FORMAT(
     *' Do you want to print the data on your screen [Y,N]?=>')
            READ (5,5011) ANS
5011  FORMAT (1A1)
            IF ((ANS.EQ.'N').OR.(ANS.EQ.'n')) GO TO 6000
c
c     user wants to print out the data file
c
            WRITE (6,5015)
5015  FORMAT('  X-AXIS DATA    Y-AXIS DATA'
     *       ,/,31('-'),/)
            DO 5020 I=1,NPTS
            WRITE (6,5030) XADATA(I),YADATA(I)
5030  FORMAT(1X,1G12.5,4X,1G12.6)
5020  CONTINUE
6000  RETURN
C
```

```
C
C
991    CONTINUE
C
C There was a problem in a defined keyword
c
           WRITE (6,9911) KEY,WORD,NUM,KEY,WORD,KTEXT
9911   FORMAT (//,' WARNING: Problem in Keyword =',1A5,1A9,
     *        /, ' CHECK YOUR DATA FILE AT THE FOLLOWING LINE',
     *        /,1A1,1A5,1A9,64A1)
           RETURN
992    CONTINUE
C
C      There was a problem with the data
c
           WRITE (6,9922) ICHN,XADATA(ICHN),YADATA(ICHN)
9922   FORMAT (//,
     * ' WARNING: Problem with spectral data AFTER point c#= ',I5,//,
     * ' Values of last good X&Y DATA are: ',2(1G12.5,','),/,
     * ' Check your data file',//,' SPECTRAL INPUT TERMINATED',//)
           RETURN
           END



C
C=======================================================================
       Subroutine MODPARM
C=======================================================================
c
c
c This subroutine MODIFIES the default settings for the current
c instrumental conditions and parameters defined in the EMSA/MAS format
c Data may be stored temporary (within the program) or permanent (as a file)
c This file may be customized by the user to reflect his/her operating
c conditions. The routine also allows individuals to have multiple
c files which reflect different operating modes to save reentering data.
c The results  from the file are stored locally in predefined
c arrays. These arrays can be passed directly back to the main
c program and can be used to redirect data elsewhere as needed.
c In this implementation the arrays are passed in labeled common
c blocks. They can alternatively be passed as subroutine
c arguments.
c
c REVISION HISTORY
c
c Version  Date  Initials  Reason
c-----------------------------------------------------------------------
c
c 1.0   10/01/91  NJZ    Creation of the World WARNING THIS VERSION
c                      Case Sensitive!!! Until DeBugged
c
c-----------------------------------------------------------------------
c Definitions of Arrays used in this Subroutine
c
c
c
c EXPARSPT = Real Array <=20 values containing Expt Parameters of Spectrum
c EXPARMSC = Real Array <=20 values containing Expt Parameters of Microscope
c EXPARSAM = Real Array <=20 values containing Expt Parameters of Sample
c EXPAREDS = Real Array <=20 values containing Expt Parameters of EDS
```

```
c EXPARELS = Real Array <=20 values containing Expt Parameters of ELS
c EXPARAES = Real Array <=20 values containing Expt Parameters of AES
c EXPARWDS = Real Array <=20 values containing Expt Parameters of WDS
c EXPARPES = Real Array <=20 values containing Expt Parameters of PES
c EXPARXRF = Real Array <=20 values containing Expt Parameters of XRF
c EXPARCLS = Real Array <=20 values containing Expt Parameters of CLS
c EXPARGAM = Real Array <=20 values containing Expt Parameters of GAM
c SIGNALTY = Byte Array of 3 characters describing the signal type
c DATATYPE = Byte Array of 3 characters describing the format
c OPERMODE = Byte Array of 5 characters describing
c           the instrument operating mode
c EDSDET   = Byte Array of 6 characters describing the EDS detector type
c ELSDET   = Byte Array of 6 characters describing the ELS detector type
c XUNITS   = Byte Array <= 64 characters describing the X-Axis Data units
c YUNITS   = Byte Array <= 64 characters describing the Y-Axis Data units
c XLABEL   = Byte Array <= 64 characters describing the X-Axis Data label
c YLABEL   = Byte Array <= 64 characters describing the Y-Axis Data label
c TITLE    = Byte Array <= 64 characters with a spectrum Title
c DATE     = Byte Array of 12 characters with the date of acquisition
c           in the form DD-MMM-YYYY
c TIME     = Byte Array of 5 characters with the time of acquisition
c           in the form HH:MM
c OWNER    = Byte Array <= 64 characters with the owner/analysts name
c COMMENT  = Byte Array <= 64 Character*1 used to hold a text comment string
c
c--------------------------------------------------------------------------
c
c Current Definitions of EXPAR (Experimental Parameter Array) Values
c
c--------------------------------------------------------------------------
c  Parameters relating to the Spectrum Characteristics = EXPARSPT
c--------------------------------------------------------------------------
c
c Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
c Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),XLABEL(64),YLABEL(64),DATATYPE(2)
c
c EXPARSPT(1) = #VERSION  = File Format Version Number
c EXPARSPT(2) = #NPOINTS  = Total Number of Data Points in X&Y Data Arrays
c         1 <= NPOINTS <= 4096
c EXPARSPT(3) = #NCOLUMNS = Number of columns of data
c         1 <= NCOLUMNS <= 5
c EXPARSPT(4) = #XPERCHAN = Increment of X-axis units per channel (eV/Channel)
c           This is only useful if (x,y) paired data are not provided
c EXPARSPT(5) = #OFFSET   = Energy value of first data point in eV
c           This is only useful if (x,y) paired data are not provided
c EXPARSPT(6) = #CHOFFSET = Channel number which corresponds to zero units along
c            x-axis, this may be either a positive or negative value
C SIGNALTY(.) = #SIGNALTYPE = Type of Spectroscopy
c        EDS = Energy Dispersive Spectroscopy
c        WDS = Wavelength Dispersive Spectroscopy
c        ELS = Energy Loss Spectroscopy
c        AES = Auger Electron Spectroscopy
c        PES = Photo Electron Spectroscopy
c        XRF = X-ray Fluorescence Spectroscopy
c        CLS = Cathodoluminescence Spectroscopy
c        GAM = Gamma Ray Spectroscopy
c XUNITS(.)   = #XUNITS = Up to 64 characters describing the X-Axis Data units
c YUNITS(.)   = #YUNITS = Up to 64 characters describing the Y-Axis Data units
c XLABEL(.)   = #XLABEL = Up to 64 characters describing the X-Axis Data label
c YLABEL(.)   = #YLABEL = Up to 64 characters describing the Y-Axis Data label
C DATATYPE(.) = #DATATYPE = Type of data format
```

```
c          Y = Spectrum Y axis data only, X-axis data to be calculated
c           using XPERCHAN and OFFSET and the following formulae
c             X = OFFSET + CHANNEL*XPERCHAN
c          XY = Spectral data is in the form of XY pairs
c
c
c--------------------------------------------------------------------------
c  Microscope/Microanalysis Instrument Parameters = EXPARMSC
c--------------------------------------------------------------------------
c
c Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,
c CONVANGLE,COLLANGLE
c Character*1 OPERMODE(5)
c
c EXPARMSC(1) = #BEAMKV    = Accelerating Voltage of Instrument in kV
c EXPARMSC(2) = #EMISSION   = Gun Emission current in microAmps
c EXPARMSC(3) = #PROBECUR   = Probe current in nanoAmps
c EXPARMSC(4) = #BEAMDIAM   = Diameter of incident probe in nanometers
c EXPARMSC(5) = #MAGCAM   = Magnification or Camera Length (Mag in x, Cl in mm)
c EXPARMSC(6) = #CONVANGLE  = Convergence semi-angle of incident beam in milliRadians
c EXPARMSC(7) = #COLLANGLE  = Collection semi-angle of scattered beam in milliRad
c OPERMODE(.) = #OPERMODE   = Operating Mode of Instrument
c                       IMAGE = Imaging Mode
c                       DIFFR = Diffraction Mode
c                       SCIMG = Scanning Imaging Mode
c                       SCDIF = Scanning Diffraction Mode
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating to the Sample = EXPARSAM
c--------------------------------------------------------------------------
c
c Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
C      ZPOSITION
c
c EXPARSAM(1) = #THICKNESS = Specimen thickness in nanometers
c EXPARSAM(2) = #XTILTSTGE = Specimen stage tilt X-axis in degrees
c EXPARSAM(3) = #YTILTSTGE = Specimen stage tilt Y-axis in degrees
c EXPARSAM(4) = #XPOSITION =  X location of beam or specimen
c EXPARSAM(5) = #YPOSITION =  Y location of beam or specimen
c EXPARSAM(6) = #ZPOSITION =  Z location of beam or specimen
c
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to ELS = EXPARELS
c--------------------------------------------------------------------------
c
c Real EXPARELS(20),DWELLTIME,INTEGTIME
c Character*1 ELSDET(6)
c
c EXPARELS(1) = #DWELLTIME  = Dwell time per channel for serial data collection in msec
c EXPARELS(2) = #INTEGTIME  = Integration time per spectrum for parallel data collection
c            in milliseconds
c ELSDET(.)  = #ELSDET   = Type of ELS Detector
c            Serial = Serial ELS Detector
c            Parall = Parallel ELS Detector
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to EDS = EXPAREDS
c--------------------------------------------------------------------------
c
c Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
```

```
c Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
c Character*1 EDSDET(6)
c
c EXPAREDS(1) = #ELEVANGLE = Elevation angle of EDS,WDS detector in degrees
c EXPAREDS(2) = #AZIMANGLE = Azimuthal angle of EDS,WDS detector in degrees
c EXPAREDS(3) = #SOLIDANGLE = Collection solid angle of detector in sR
c EXPAREDS(4) = #LIVETIME  = Signal Processor Active (Live) time in seconds
c EXPAREDS(5) = #REALTIME  = Total clock time used to record the spectrum in seconds
c EXPAREDS(6) = #TBEWIND  = Thickness of Be Window on detector in cm
c EXPAREDS(7) = #TAUWIND  = Thickness of Au Window/Electrical Contact in cm
c EXPAREDS(8) = #TDEADLYR  = Thickness of Dead Layer in cm
c EXPAREDS(9) = #TACTLYR  = Thickness of Active Layer in cm
c EXPAREDS(10) = #TALWIND  = Thickness of Aluminium Window in cm
c EXPAREDS(11) = #TPYWIND  = Thickness of Pyrolene Window in cm
c EXPAREDS(12) = #TBNWIND  = Thickness of Boron-Nitride Window in cm
c EXPAREDS(13) = #TDIWIND  = Thickness of Diamond Window in cm
c EXPAREDS(14) = #THCWIND  = Thickness of HydroCarbon Window in cm
c EDSDET(.)    = #EDSDET  = Type of X-ray Detector
c         SIBEW = Si(Li) with Be Window
c         SIUTW = Si(Li) with Ultra Thin Window
c         SIWLS = Si(Li) Windowless
c         GEBEW = Ge with Be Window
c         GEUTW = Ge with Ultra Thin Window
c         GEWLS = Ge Windowless
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to WDS = EXPARWDS
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to XRF = EXPARXRF
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to AES = EXPARAES
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to PES = EXPARPES
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to CLS = EXPARCLS
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to GAM = EXPARGAM
c--------------------------------------------------------------------------
c
c      Nothing currently defined
c
```

```
c=======================================================================
c       END OF DEFINITIONS
c=======================================================================
c
c
c Start Code
c
c
c   Dimension all arrays, & Declare Variable Types
c   REMEMBER we have no integer parameters!!!!!
c
c
      Character*1 TITLE(64),DATE(12),TIME(5),OWNER(64),COMMENT(64)
      Real XADATA(4096),YADATA(4096)
      Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
      Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),
     *       XLABEL(64),YLABEL(64),DATATYPE(2)
      Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,CONVANGLE,
     *      COLLANGLE
      Character*1 OPERMODE(5)
      Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
     *      ZPOSITION
      Real EXPARELS(20),DWELLTIME,INTEGTIME
      Character*1 ELSDET(6)
      Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
      Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
      Character*1 EDSDET(6)
      Real EXPARWDS(20),EXPARXRF(20),EXPARAES(20),EXPARPES(20),EXPARCLS(20),
     *       EXPARGAM(20)
c-------------------------------------------------------------------------------
c Set up all Variable arrays to be passed in labeled common block for ease of
c passing parameters between different subroutines
c-------------------------------------------------------------------------------
c
      Common /XYDATA/ XADATA,YADATA
      Common /EXPPAR/ EXPARSPT,EXPARMSC,EXPARSAM,EXPARELS,EXPAREDS,EXPARWDS,EXPARAES,
     *   EXPARPES,EXPARXRF,EXPARCLS,EXPARGAM
      Common /TEXT/ TITLE,DATE,TIME,OWNER,COMMENT,
     *   SIGNALTY,XUNITS,YUNITS,XLABEL,YLABEL,DATATYPE,
     *   OPERMODE,ELSDET,EDSDET
c
      Character*1 ANS,NUM,FKEY
      Character*3 STYPE
      Character*5 KEY
      Character*9 WORD
     EQUIVALENCE ( EXPARSPT(1),VERSION)
     EQUIVALENCE ( EXPARSPT(2),NPOINTS)
     EQUIVALENCE ( EXPARSPT(3),NCOLUMNS)
     EQUIVALENCE ( EXPARSPT(4),XPERCHAN)
     EQUIVALENCE ( EXPARSPT(5),OFFSET)
     EQUIVALENCE ( EXPARSPT(6),CHOFFSET)
     EQUIVALENCE ( EXPARMSC(1),BEAMKV)
     EQUIVALENCE ( EXPARMSC(2),EMISSION)
     EQUIVALENCE ( EXPARMSC(3),PROBECUR)
     EQUIVALENCE ( EXPARMSC(4),BEAMDIAM)
     EQUIVALENCE ( EXPARMSC(5),MAGCAM)
     EQUIVALENCE ( EXPARMSC(6),CONVANGLE)
     EQUIVALENCE ( EXPARMSC(7),COLLANGLE)
     EQUIVALENCE ( EXPARSAM(1),THICKNESS)
     EQUIVALENCE ( EXPARSAM(2),XTILTSTGE)
     EQUIVALENCE ( EXPARSAM(3),YTILTSTGE)
```

```
      EQUIVALENCE ( EXPARSAM(4),XPOSITION)
      EQUIVALENCE ( EXPARSAM(5),YPOSITION)
      EQUIVALENCE ( EXPARSAM(6),ZPOSITION)
      EQUIVALENCE ( EXPARELS(1),DWELLTIME)
      EQUIVALENCE ( EXPARELS(2),INTEGTIME)
      EQUIVALENCE ( EXPAREDS(1),ELEVANGLE)
      EQUIVALENCE ( EXPAREDS(2),AZIMANGLE)
      EQUIVALENCE ( EXPAREDS(3),SOLIDANGLE)
      EQUIVALENCE ( EXPAREDS(4),LIVETIME)
      EQUIVALENCE ( EXPAREDS(5),REALTIME)
      EQUIVALENCE ( EXPAREDS(6),TBEWIND)
      EQUIVALENCE ( EXPAREDS(7),TAUWIND)
      EQUIVALENCE ( EXPAREDS(8),TDEADLYR)
      EQUIVALENCE ( EXPAREDS(9),TACTLYR)
      EQUIVALENCE ( EXPAREDS(10),TALWIND)
      EQUIVALENCE ( EXPAREDS(11),TPYWIND)
      EQUIVALENCE ( EXPAREDS(12),TBNWIND)
      EQUIVALENCE ( EXPAREDS(13),TDIWIND)
      EQUIVALENCE ( EXPAREDS(14),THCWIND)
c
c
c      NOW define different arrays used only in this routine
c
c
c The array FNAME = Name of File to be read, Usually null terminated
c so we will initialize it with zero's using a Do loop so that successive
c operations of the routine within a single program donot cause problems
c if a short filename is used.
c
c FNAME   = Byte array of 17 characters containing the filename
c KText  = Byte array of 64 characters containing Keyword text
c
c
      Character*17 FNAME,DNAME
      Character*1 KTEXT(64),PERM
      Logical*1 TF
c
c     Permanent or Temporary?
c
100   Write (6,*) 'Should the changes be Stored as a File? [Y,N]'
     READ (6,101) PERM
101    FORMAT (1A1)
     IF (PERM.EQ.' ') GO TO 100
     IF ((PERM.EQ.'N').OR.(PERM.EQ.'n')) THEN
     WRITE (6,*) 'Changes are not Stored'
     GO TO 200
     ELSE
     Write (6,*) 'Changes will be Stored'
     PERM='Y'
c
c     User wants to store changes onto a file
c
c Initialize Default Filename
c
c
     DNAME='SYSPARM.DAT'
c
c
c
109       WRITE (6,110)
110    FORMAT (
```

```
    *   ' Enter name of a System Parameters file [SYSPARM.DAT]=> ')
     READ (5,120) FNAME
120    FORMAT(1A16)
         IF (FNAME.EQ.' ') THEN
              FNAME=DNAME
              ENDIF
c
c     check that the file exists
c
     INQUIRE (FILE=FNAME,EXIST=TF)
     IF (TF) THEN
c
c          Replacing a file!!! Warn user.
c
          Write (6,*)
    *      'WARNING: Replacing Existing File Continue? [Y,N]: '
          Read (6,101)  ANS
          IF (ANS.EQ.'y') ANS='Y'
          IF (ANS.NE.'Y') GO TO 109
          ENDIF
c
c          Now open the file
c
          OPEN (UNIT=98,BLANK='NULL',ACCESS='SEQUENTIAL',FILE=FNAME)
c
c Now lets write the defined parameters as defined by EMSA/MAS
c
c
c
     WRITE (6,1200) FNAME
1200  FORMAT (' Current System Parameters in File : ',1a16,/)
200   WRITE (6,1201)
1201  FORMAT(
    *  ' ============================================================',/,
    *  ' You may also change System Parameters outside this program',/,
    *  '    by using the TEXT EDITOR supplied with your computer.',//,
    *  '  TO CHANGE A PARAMETER TYPE IN THE NEW VALUE AT THE PROMPT',/,
    *  '  TO LEAVE  A PARAMETER UNCHANGED ENTER A <RETURN> INSTEAD',/,
    *  '    Note: Only data independent parameters can be edited.',/,
    *  ' ============================================================',/)
c
c     The system parameters file should be customized by the user
c     to reflect their standard conditions. The user should
c     use a simple screen editor to update the default file
c     provided. We will read all the parameters in the file
c     and if they correspond to a defined EMSA/MAS keyword then
c     store the value. If they are not standard then they will
c     be simply typed on the screen. Some of these parameters
c     will logically be updated when a EMSA/MAS file is written
c     to disk. These are indicated by a * when typed on the screen.
c
c     Note: the System Parameters file may not have all the
c     predefined EMSA/MAS keywords, but we must check for all
c     of them regardless.
c=====================================================================
c
c     Ask the user if full header or partial header is desired
c
4001  WRITE (6,4000)
4000  FORMAT (
    * ' Select the type of Information to be Edited:',//,
```

```
     * 10X,' B = Basic EMSA/MAS Header Information Only',/,
     * 10x,' F = Full  EMSA/MAS Header',/,
     * 10x,' P = Partial Headers based upon Current Signal Type',/,
     * 10x,'     [Includes Basic,Specimen & Instrument Headers]',//,
     * 10x,' ==> ')
      READ (6,101) ANS
      IF (ANS.EQ.' ') THEN
        GO TO 4001
        ELSE
          IF (ANS.EQ.'p') ANS = 'P'
          IF (ANS.EQ.'b') ANS = 'B'
          IF (ANS.EQ.'f') ANS = 'F'
          IF (((ANS.EQ.'F').OR.(ANS.EQ.'B')).OR.(ANS.EQ.'P')) THEN
                GO TO 4002
                ELSE
                GO TO 4001
                ENDIF
        ENDIF
4002  CONTINUE

c
c Now lets write the required parameters as defined by EMSA/MAS
c
c=====================================================================
c
c First WRITE the Format USER CAN'T MODIFY THIS
c
      CALL TWRITE ('#FORMAT      : ',
     *  'EMSA/MAS SPECTRAL DATA STANDARD',31,PERM)
c125   FORMAT (1A46)
c126   FORMAT (1A15,64A1)
c1261  FORMAT (1X,1A15,64A1)
c
c now the version  USER CAN'T MODIFY THIS
c
      CALL VWRITE ('#VERSION     : ',VERSION,PERM)
c130   FORMAT(1A15,1G15.2)
c1301  FORMAT (1X,1A15,1G15.2)
c
c now the Title BUT give user option to change it.
c
      CALL TMODIFY('#TITLE       : ',TITLE,64)
      CALL TWRITE ('#TITLE       : ',TITLE,64,PERM)
c
c now the Date  USER CAN'T MODIFY THIS AS THE PROGRAM DOES IT
c
      CALL TWRITE ('#DATE        : ', DATE,12,PERM)
c
c
c now the Time  USER CAN'T MODIFY THIS
c
      CALL TWRITE ('#TIME        : ',TIME,5,PERM)

C
C
c now the Owner but give the option to change it.
c
      CALL TMODIFY('#OWNER       : ',OWNER,64)
      CALL TWRITE ('#OWNER       : ',OWNER,64,PERM)

c
```

```
c now the Npoints USER CAN'T MODIFY THIS BECAUSE IT IS DATA DEPENDANT
c
     CALL VWRITE ('#NPOINTS     : ',NPOINTS,PERM)
c
c now the NColumns give the option to change it.
c
     CALL VMODIFY('#NCOLUMNS    : ',NCOLUMNS)
     CALL VWRITE ('#NCOLUMNS    : ',NCOLUMNS,PERM)
c
c now the XUnits
c
     CALL TMODIFY ('#XUNITS      : ',XUNITS,64)
     CALL TWRITE  ('#XUNITS      : ',XUNITS,64,PERM)
c
c now the YUnits
c
     CALL TMODIFY('#YUNITS      : ',YUNITS,64)
     CALL TWRITE ('#YUNITS      : ',YUNITS,64,PERM)
c
c now the DATATYPE
c
     CALL TMODIFY('#DATATYPE    : ',DATATYPE,2)
     CALL TWRITE ('#DATATYPE    : ',DATATYPE,2,PERM)
c
c now the XPERCHAN USER CAN'T MODIFY THIS BECAUSE IT IS DATA DEPENDANT
c
     CALL VWRITE ('#XPERCHAN    : ',XPERCHAN,PERM)
c
c now the OFFSET USER CAN'T MODIFY THIS BECAUSE IT IS DATA DEPENDANT
c
      CALL VWRITE ('#OFFSET      : ',OFFSET,PERM)
c
c We now have the minimum data required
c begin checking individual key words
c
c Check the next keyword
c write it out if of its value exists and
c
c
c We now have the minimum data required
c begin checking individual key words
c based upon user selected option of FULL or Partial Headers
c as well as the Current Signal Type.
c
c Check the next keyword
c write it out if of its value exists and
c
c
     IF (ANS.EQ.'B') WRITE (6,*) 'Basic Header Requested'
     IF (ANS.EQ.'F') WRITE (6,*) 'Full Header Requested'
     IF (ANS.EQ.'P') WRITE (6,*) 'Partial Header Requested'
c
c     concatenate the signaltype characters into a
c     single character constant for ease of use below
c
     STYPE =SIGNALTY(1)//SIGNALTY(2)//SIGNALTY(3)
c      write (6,*) 'signaltype = ',stype
c
c now for modification of each predefined keyword
c
     IF ((ANS.EQ.'F').OR.(ANS.EQ.'P')) THEN
```

```
c
c     this next set of parameters is either instrument
c     or sample related therfore if F or P we will write it
c     if not then exit routine (many lines down at ELSE)
c
c     can't modify this   CALL VMODIFY ( '#CHOFFSET    : ',CHOFFSET)
          CALL VWRITE ( '#CHOFFSET    : ',CHOFFSET,PERM)
          CALL TMODIFY ('#SIGNALTYPE  : ',SIGNALTY,3)
          CALL TWRITE ( '#SIGNALTYPE  : ',SIGNALTY,3,PERM)
          CALL TMODIFY( '#XLABEL      : ',XLABEL,64)
          CALL TWRITE ( '#XLABEL      : ',XLABEL,64,PERM)
          CALL TMODIFY( '#YLABEL      : ',YLABEL,64)
          CALL TWRITE ( '#YLABEL      : ',YLABEL,64,PERM)
          CALL VMODIFY( '#BEAMKV   -kV: ',BEAMKV)
          CALL VWRITE ( '#BEAMKV   -kV: ',BEAMKV,PERM)
          CALL VMODIFY( '#EMISSION -uA: ',EMISSION)
          CALL VWRITE ( '#EMISSION -uA: ',EMISSION,PERM)
          CALL VMODIFY( '#PROBECUR -nA: ',PROBECUR)
          CALL VWRITE ( '#PROBECUR -nA: ',PROBECUR,PERM)
          CALL VMODIFY( '#BEAMDIAM -nm: ',BEAMDIAM)
          CALL VWRITE ( '#BEAMDIAM -nm: ',BEAMDIAM,PERM)
          CALL VMODIFY( '#MAGCAM      : ',MAGCAM)
          CALL VWRITE ( '#MAGCAM      : ',MAGCAM,PERM)
          CALL VMODIFY( '#CONVANGLE-mR: ',CONVANGLE)
          CALL VWRITE ( '#CONVANGLE-mR: ',CONVANGLE,PERM)
          CALL VMODIFY( '#COLLANGLE-mR: ',COLLANGLE)
          CALL VWRITE ( '#COLLANGLE-mR: ',COLLANGLE,PERM)
          CALL TMODIFY( '#OPERMODE    : ',OPERMODE,5)
          CALL TWRITE ( '#OPERMODE    : ',OPERMODE,5,PERM)
          CALL VMODIFY( '#THICKNESS-nm: ',THICKNESS)
          CALL VWRITE ( '#THICKNESS-nm: ',THICKNESS,PERM)
          CALL VMODIFY( '#XTILTSTAGE  : ',XTILTSTGE)
          CALL VWRITE ( '#XTILTSTAGE  : ',XTILTSTGE,PERM)
          CALL VMODIFY( '#YTILTSTAGE  : ',YTILTSTGE)
          CALL VWRITE ( '#YTILTSTAGE  : ',YTILTSTGE,PERM)
          CALL VMODIFY( '#XPOSITION   : ',XPOSITION)
          CALL VWRITE ( '#XPOSITION   : ',XPOSITION,PERM)
          CALL VMODIFY( '#YPOSITION   : ',YPOSITION)
          CALL VWRITE ( '#YPOSITION   : ',YPOSITION,PERM)
          CALL VMODIFY( '#ZPOSITION   : ',ZPOSITION)
          CALL VWRITE ( '#ZPOSITION   : ',ZPOSITION,PERM)
      ELSE
C
C           IF FALSE THEN EXIT MODIFICATION ROUTINE
C
          GO TO 9000
      ENDIF
C
C
C     The following parameters are EELS related
c
c     check for full or partial
c
      IF ((ANS.EQ.'F').OR.
     *       ((ANS.EQ.'P').AND.(STYPE.EQ.'ELS'))) THEN
          CALL TMODIFY('#ELSDET       : ',ELSDET,6)
          CALL TWRITE ('#ELSDET       : ',ELSDET,6,PERM)
c
c           Dwell time & Integ time are data related User cannot change
c
          CALL VWRITE ( '#DWELLTIME-ms: ',DWELLTIME,PERM)
```

```
          CALL VWRITE ('#INTEGTIME-ms: ',INTEGTIME,PERM)
      ENDIF
c
c     The following parameters are XEDS related
c
c
      IF ((ANS.EQ.'F').OR.
     *       ((ANS.EQ.'P').AND.(STYPE.EQ.'EDS'))) THEN
c
c     okay now act on EDS signal parameters
c
          CALL TMODIFY('#EDSDET      : ',EDSDET,5)
          CALL TWRITE ('#EDSDET      : ',EDSDET,5,PERM)
          CALL VMODIFY('#ELEVANGEL-dg: ',ELEVANGLE)
          CALL VWRITE ('#ELEVANGLE-dg: ',ELEVANGLE,PERM)
          CALL VMODIFY('#AZIMANGLE-dg: ',AZIMANGLE)
          CALL VWRITE ('#AZIMANGLE-dg: ',AZIMANGLE,PERM)
          CALL VMODIFY('#SOLIDANGL-sR: ',SOLIDANGLE)
          CALL VWRITE ('#SOLIDANGL-sR: ',SOLIDANGLE,PERM)
C
C          Live and Real Time are Data related and not changable by user
c
          CALL VWRITE ('#LIVETIME  -s: ',LIVETIME,PERM)
          CALL VWRITE ('#REALTIME  -s: ',REALTIME,PERM)
          CALL VMODIFY('#TBEWIND  -cm: ',TBEWIND)
          CALL VWRITE ('#TBEWIND  -cm: ',TBEWIND,PERM)
          CALL VMODIFY('#TAUWIND  -cm: ',TAUWIND)
          CALL VWRITE ('#TAUWIND  -cm: ',TAUWIND,PERM)
          CALL VMODIFY('#TDEADLYR -cm: ',TDEADLYR)
          CALL VWRITE ('#TDEADLYR -cm: ',TDEADLYR,PERM)
          CALL VMODIFY('#TACTLYR  -cm: ',TACTLYR)
          CALL VWRITE ('#TACTLYR  -cm: ',TACTLYR,PERM)
          CALL VMODIFY('#TALWIND  -cm: ',TALWIND)
          CALL VWRITE ('#TALWIND  -cm: ',TALWIND,PERM)
          CALL VMODIFY('#TPYWIND  -cm: ',TPYWIND)
          CALL VWRITE ('#TPYWIND  -cm: ',TPYWIND,PERM)
          CALL VMODIFY('#TBNWIND  -cm: ',TBNWIND)
          CALL VWRITE ('#TBNWIND  -cm: ',TBNWIND,PERM)
          CALL VMODIFY('#TDIWIND  -cm: ',TDIWIND)
          CALL VWRITE ('#TDIWIND  -cm: ',TDIWIND,PERM)
          CALL VMODIFY('#THCWIND  -cm: ',THCWIND)
          CALL VWRITE ('#THCWIND  -cm: ',THCWIND,PERM)
      ENDIF
c
c          No Other Parameters currently defined by EMSA/MAS
c
9000  CALL TWRITE ('#SPECTRUM    : ',
     *               'Spectral Data Starts Here',25,PERM)
c
c=====================================================================
c
      WRITE(6,1100)
1100  FORMAT(1X,14('='),'End of Parameter File',14('='),//)
      If (PERM.EQ.'Y') close (unit=98)
6000  RETURN
      END
           SUBROUTINE RDNEMMFF
c=====================================================================
c
c
c
```

```
c REVISION HISTORY
c
c Version  Date  Initials  Reason
c-------------------------------------------------------------------------
c
c 1.0   10/01/91  NJZ    Creation of the World
c
c
c-------------------------------------------------------------------------
c
c     This subroutine access files stored in
c     other CUSTOM data formats such as that of commerical manufacturers
c     This routine should be modified to add translators from
c     other file formats. In this example the EMMPDL file format
c     is translated into the EMSA/MAS Format.
c
c     Data should be passed using the predefined common blocks
c     defined in the main program
c
c
c-------------------------------------------------------------------------
c Definitions of Arrays used in this Subroutine
c
c
c
c XADATA   = Real Array <= 4096 values containing X-Axis Data
c YADATA   = Real Array <= 4096 values containing Y-Axis Data
c EXPARSPT = Real Array <=20 values containing Expt Parameters of Spectrum
c EXPARMSC = Real Array <=20 values containing Expt Parameters of Microscope
c EXPARSAM = Real Array <=20 values containing Expt Parameters of Sample
c EXPAREDS = Real Array <=20 values containing Expt Parameters of EDS
c EXPARELS = Real Array <=20 values containing Expt Parameters of ELS
c EXPARAES = Real Array <=20 values containing Expt Parameters of AES
c EXPARWDS = Real Array <=20 values containing Expt Parameters of WDS
c EXPARPES = Real Array <=20 values containing Expt Parameters of PES
c EXPARXRF = Real Array <=20 values containing Expt Parameters of XRF
c EXPARCLS = Real Array <=20 values containing Expt Parameters of CLS
c EXPARGAM = Real Array <=20 values containing Expt Parameters of GAM
c SIGNALTY = Byte Array of 3 characters describing the signal type
c DATATYPE = Byte Array of 2 characters describing the format
c OPERMODE = Byte Array of 5 characters describing
c           the instrument operating mode
c EDSDET   = Byte Array of 6 characters describing the EDS detector type
c ELSDET   = Byte Array of 6 characters describing the ELS detector type
c XUNITS   = Byte Array <= 64 characters describing the X-Axis Data units
c YUNITS   = Byte Array <= 64 characters describing the Y-Axis Data units
c XLABEL   = Byte Array <= 64 characters describing the X-Axis Data label
c YLABEL   = Byte Array <= 64 characters describing the Y-Axis Data label
c TITLE    = Byte Array <= 64 characters with a spectrum Title
c DATE     = Byte Array of 12 characters with the date of acquisition
c           in the form DD-MMM-YYYY
c TIME     = Byte Array of 5 characters with the time of acquisition
c           in the form HH:MM
c OWNER    = Byte Array <= 64 characters with the owner/analysts name
c COMMENT  = Byte Array <= 64 Character*1 used to hold a text comment string
c
c-------------------------------------------------------------------------
c
c Current Definitions of EXPAR (Experimental Parameter Array) Values
c
c-------------------------------------------------------------------------
```

```
c  Parameters relating to the Spectrum Characteristics = EXPARSPT
c--------------------------------------------------------------------------
c
c Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
c Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),XLABEL(64),YLABEL(64),DATATYPE(2)
c
c EXPARSPT(1) = #VERSION  = File Format Version Number
c EXPARSPT(2) = #NPOINTS  = Total Number of Data Points in X&Y Data Arrays
c         1 <= NPOINTS <= 4096
c EXPARSPT(3) = #NCOLUMNS = Number of columns of data
c         1 <= NCOLUMNS <= 5
c EXPARSPT(4) = #XPERCHAN = Increment of X-axis units per channel (eV/Channel)
c           This is only useful if (x,y) paired data are not provided
c EXPARSPT(5) = #OFFSET   = Energy value of first data point in eV
c           This is only useful if (x,y) paired data are not provided
c EXPARSPT(6) = #CHOFFSET = Channel number which corresponds to zero units along
c            x-axis, this may be either a positive or negative value
C SIGNALTY(.) = #SIGNALTYPE = Type of Spectroscopy
c       EDS = Energy Dispersive Spectroscopy
c       WDS = Wavelength Dispersive Spectroscopy
c       ELS = Energy Loss Spectroscopy
c       AES = Auger Electron Spectroscopy
c       PES = Photo Electron Spectroscopy
c       XRF = X-ray Fluorescence Spectroscopy
c       CLS = Cathodoluminescence Spectroscopy
c       GAM = Gamma Ray Spectroscopy
c XUNITS(.)   = #XUNITS = Up to 64 characters describing the X-Axis Data units
c YUNITS(.)   = #YUNITS = Up to 64 characters describing the Y-Axis Data units
c XLABEL(.)   = #XLABEL = Up to 64 characters describing the X-Axis Data label
c YLABEL(.)   = #YLABEL = Up to 64 characters describing the Y-Axis Data label
C DATATYPE(.) = #DATATYPE = Type of data format
c          Y = Spectrum Y axis data only, X-axis data to be calculated
c           using XPERCHAN and OFFSET and the following formulae
c            X = OFFSET + CHANNEL*XPERCHAN
c          XY = Spectral data is in the form of XY pairs
c
c
c--------------------------------------------------------------------------
c  Microscope/Microanalysis Instrument Parameters = EXPARMSC
c--------------------------------------------------------------------------
c
c Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,
c CONVANGLE,COLLANGLE
c Character*1 OPERMODE(5)
c
c EXPARMSC(1) = #BEAMKV    = Accelerating Voltage of Instrument in kV
c EXPARMSC(2) = #EMISSION   = Gun Emission current in microAmps
c EXPARMSC(3) = #PROBECUR   = Probe current in nanoAmps
c EXPARMSC(4) = #BEAMDIAM   = Diameter of incident probe in nanometers
c EXPARMSC(5) = #MAGCAM   = Magnification or Camera Length (Mag in x, Cl in mm)
c EXPARMSC(6) = #CONVANGLE  = Convergence semi-angle of incident beam in milliRadians
c EXPARMSC(7) = #COLLANGLE  = Collection semi-angle of scattered beam in milliRad
c OPERMODE(.) = #OPERMODE   = Operating Mode of Instrument
c                       IMAGE = Imaging Mode
c                       DIFFR = Diffraction Mode
c                       SCIMG = Scanning Imaging Mode
c                       SCDIF = Scanning Diffraction Mode
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating to the Sample = EXPARSAM
c--------------------------------------------------------------------------
```

```
c
c Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
C      ZPOSITION
c
c EXPARSAM(1) = #THICKNESS = Specimen thickness in nanometers
c EXPARSAM(2) = #XTILTSTGE = Specimen stage tilt X-axis in degrees
c EXPARSAM(3) = #YTILTSTGE = Specimen stage tilt Y-axis in degrees
c EXPARSAM(4) = #XPOSITION =  X location of beam or specimen
c EXPARSAM(5) = #YPOSITION =  Y location of beam or specimen
c EXPARSAM(6) = #ZPOSITION =  Z location of beam or specimen
c
c
c-------------------------------------------------------------------------
c  Experimental Parameters relating mainly to ELS = EXPARELS
c-------------------------------------------------------------------------
c
c Real EXPARELS(20),DWELLTIME,INTEGTIME
c Character*1 ELSDET(6)
c
c EXPARELS(1) = #DWELLTIME  = Dwell time per channel for serial data collection in msec
c EXPARELS(2) = #INTEGTIME  = Integration time per spectrum for parallel data collection
c            in milliseconds
c ELSDET(.)  = #ELSDET   = Type of ELS Detector
c            Serial = Serial ELS Detector
c            Parall = Parallel ELS Detector
c
c-------------------------------------------------------------------------
c  Experimental Parameters relating mainly to EDS = EXPAREDS
c-------------------------------------------------------------------------
c
c Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
c Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
c Character*1 EDSDET(6)
c
c EXPAREDS(1) = #ELEVANGLE = Elevation angle of EDS,WDS detector in degrees
c EXPAREDS(2) = #AZIMANGLE = Azimuthal angle of EDS,WDS detector in degrees
c EXPAREDS(3) = #SOLIDANGLE = Collection solid angle of detector in sR
c EXPAREDS(4) = #LIVETIME  = Signal Processor Active (Live) time in seconds
c EXPAREDS(5) = #REALTIME  = Total clock time used to record the spectrum in seconds
c EXPAREDS(6) = #TBEWIND  = Thickness of Be Window on detector in cm
c EXPAREDS(7) = #TAUWIND  = Thickness of Au Window/Electrical Contact in cm
c EXPAREDS(8) = #TDEADLYR  = Thickness of Dead Layer in cm
c EXPAREDS(9) = #TACTLYR  = Thickness of Active Layer in cm
c EXPAREDS(10) = #TALWIND  = Thickness of Aluminium Window in cm
c EXPAREDS(11) = #TPYWIND  = Thickness of Pyrolene Window in cm
c EXPAREDS(12) = #TBNWIND  = Thickness of Boron-Nitride Window in cm
c EXPAREDS(13) = #TDIWIND  = Thickness of Diamond Window in cm
c EXPAREDS(14) = #THCWIND  = Thickness of HydroCarbon Window in cm
c EDSDET(.)    = #EDSDET  = Type of X-ray Detector
c        SIBEW = Si(Li) with Be Window
c        SIUTW = Si(Li) with Ultra Thin Window
c        SIWLS = Si(Li) Windowless
c        GEBEW = Ge with Be Window
c        GEUTW = Ge with Ultra Thin Window
c        GEWLS = Ge Windowless
c
c-------------------------------------------------------------------------
c  Experimental Parameters relating mainly to WDS = EXPARWDS
c-------------------------------------------------------------------------
c
c     Nothing currently defined
```

```
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to XRF = EXPARXRF
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to AES = EXPARAES
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to PES = EXPARPES
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to CLS = EXPARCLS
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to GAM = EXPARGAM
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c=======================================================================
c        END OF DEFINITIONS
c=======================================================================
c
c
c Start Code
c
c
c   Dimension all arrays, & Declare Variable Types
c   REMEMBER we have no integer parameters!!!!!
c
c
      Character*1 TITLE(64),DATE(12),TIME(5),OWNER(64),COMMENT(64)
      Real XADATA(4096),YADATA(4096)
      Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
      Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),
     *        XLABEL(64),YLABEL(64),DATATYPE(2)
      Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,CONVANGLE,
     *       COLLANGLE
      Character*1 OPERMODE(5)
      Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
     *       ZPOSITION
      Real EXPARELS(20),DWELLTIME,INTEGTIME
      Character*1 ELSDET(6)
      Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
      Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
      Character*1 EDSDET(6)
      Real EXPARWDS(20),EXPARXRF(20),EXPARAES(20),EXPARPES(20),EXPARCLS(20),
     *       EXPARGAM(20)
c-----------------------------------------------------------------------------
```

```
c Set up all Variable arrays to be passed in labeled common block for ease of
c passing parameters between different subroutines
c-------------------------------------------------------------------------------
c
      Common /XYDATA/ XADATA,YADATA
      Common /EXPPAR/ EXPARSPT,EXPARMSC,EXPARSAM,EXPARELS,EXPAREDS,EXPARWDS,EXPARAES,
     *    EXPARPES,EXPARXRF,EXPARCLS,EXPARGAM
      Common /TEXT/ TITLE,DATE,TIME,OWNER,COMMENT,
     *    SIGNALTY,XUNITS,YUNITS,XLABEL,YLABEL,DATATYPE,
     *    OPERMODE,ELSDET,EDSDET
c
      Character*1 NUM,FKEY
      Character*5 KEY
      Character*4 WORD
      Character*2 ANS
      EQUIVALENCE ( EXPARSPT(1),VERSION)
      EQUIVALENCE ( EXPARSPT(2),NPOINTS)
      EQUIVALENCE ( EXPARSPT(3),NCOLUMNS)
      EQUIVALENCE ( EXPARSPT(4),XPERCHAN)
      EQUIVALENCE ( EXPARSPT(5),OFFSET)
      EQUIVALENCE ( EXPARSPT(6),CHOFFSET)
      EQUIVALENCE ( EXPARMSC(1),BEAMKV)
      EQUIVALENCE ( EXPARMSC(2),EMISSION)
      EQUIVALENCE ( EXPARMSC(3),PROBECUR)
      EQUIVALENCE ( EXPARMSC(4),BEAMDIAM)
      EQUIVALENCE ( EXPARMSC(5),MAGCAM)
      EQUIVALENCE ( EXPARMSC(6),CONVANGLE)
      EQUIVALENCE ( EXPARMSC(7),COLLANGLE)
      EQUIVALENCE ( EXPARSAM(1),THICKNESS)
      EQUIVALENCE ( EXPARSAM(2),XTILTSTGE)
      EQUIVALENCE ( EXPARSAM(3),YTILTSTGE)
      EQUIVALENCE ( EXPARSAM(4),XPOSITION)
      EQUIVALENCE ( EXPARSAM(5),YPOSITION)
      EQUIVALENCE ( EXPARSAM(6),ZPOSITION)
      EQUIVALENCE ( EXPARELS(1),DWELLTIME)
      EQUIVALENCE ( EXPARELS(2),INTEGTIME)
      EQUIVALENCE ( EXPAREDS(1),ELEVANGLE)
      EQUIVALENCE ( EXPAREDS(2),AZIMANGLE)
      EQUIVALENCE ( EXPAREDS(3),SOLIDANGLE)
      EQUIVALENCE ( EXPAREDS(4),LIVETIME)
      EQUIVALENCE ( EXPAREDS(5),REALTIME)
      EQUIVALENCE ( EXPAREDS(6),TBEWIND)
      EQUIVALENCE ( EXPAREDS(7),TAUWIND)
      EQUIVALENCE ( EXPAREDS(8),TDEADLYR)
      EQUIVALENCE ( EXPAREDS(9),TACTLYR)
      EQUIVALENCE ( EXPAREDS(10),TALWIND)
      EQUIVALENCE ( EXPAREDS(11),TPYWIND)
      EQUIVALENCE ( EXPAREDS(12),TBNWIND)
      EQUIVALENCE ( EXPAREDS(13),TDIWIND)
      EQUIVALENCE ( EXPAREDS(14),THCWIND)
c
c
c     NOW define different arrays used only in this routine
c
c-------------------------------------------------------------------
c     This code does nothing at the present time but write
c     a message on the screen
c-------------------------------------------------------------------
c
     Write (6,*) 'Select Format to be Translated:'
     Write (6,100)
```

```
100   FORMAT (
     * 10X,' EMMPDL = EM',/,
     * 10X,'*EDAX   = ED',/,
     * 10x,'*GATAN  = GA',/,
     * 10X,'*LINK   = LI',/,
     * 10X,'*KEVEX  = KE',/,
     * 10x,'*NORAN  = NO',/,
     * 10x,'*PGT    = PG',/,
     * 10X,'*TRACOR = TR',/,
     * 10X,'* = Translator not presently available.',/,
     * 10X,'========> ')
      READ (6,101) ANS
101   FORMAT (1A2)
      IF ((ANS.EQ.'EM').or.(ANS.EQ.'em')) THEN
           CALL RDEMMPDL
           GO TO 200
           ENDIF
      IF (ANS.EQ.'ED') THEN
c            CALL RDEDAX
           GO TO 201
           ENDIF
      IF (ANS.EQ.'GA') THEN
c            CALL RDGATAN
           GO TO 201
           ENDIF
      IF (ANS.EQ.'LI') THEN
c            CALL RDLINK
           GO TO 201
           ENDIF
      IF (ANS.EQ.'KE') THEN
c            CALL RDKEVEX
           GO TO 201
           ENDIF
      IF (ANS.EQ.'NO') THEN
c            CALL RDNORAN
           GO TO 201
           ENDIF
      IF (ANS.EQ.'PG') THEN
c            CALL RDPGT
           GO TO 201
           ENDIF
      IF (ANS.EQ.'TR') THEN
c            CALL RDTRACOR
           GO TO 201
           ENDIF
C
C     IF WE GOT THIS FAR THERE WAS NO FORMAT SELECTED
C
201   WRITE (6,*) 'Unknown or Not Implemented Format Selected'
200   CONTINUE
      RETURN
      END
C
C
C=====================================================================
      Subroutine RDEMMPDL
c=====================================================================
c
c     This subroutine READS an EMMPDL version 1.1 spectral file
c
c     It should act as a model for other routines to be developed
```

```
c     additional Read routines can be adapted from the
c     NTRANS program in the EMMPDL and will be added to this code
c     by spring 1992
c
c
c
c
c REVISION HISTORY
c
c Version  Date  Initials  Reason
c-------------------------------------------------------------------------
c
c 1.0   10/01/91  NJZ    Creation of the World
c
c
c-------------------------------------------------------------------------
c
c=========================================================================
c         See main routine for all array definitions
c=========================================================================
c
c
c Start Code
c
c
c   Dimension all arrays, & Declare Variable Types
c   REMEMBER we have no integer parameters!!!!!
c
c
       Character*1 TITLE(64),DATE(12),TIME(5),OWNER(64),COMMENT(64)
       Real XADATA(4096),YADATA(4096)
       Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
       Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),
     *         XLABEL(64),YLABEL(64),DATATYPE(2)
       Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,CONVANGLE,
     *       COLLANGLE
       Character*1 OPERMODE(5)
       Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
     *       ZPOSITION
       Real EXPARELS(20),DWELLTIME,INTEGTIME
       Character*1 ELSDET(6)
       Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
       Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
       Character*1 EDSDET(6)
       Real EXPARWDS(20),EXPARXRF(20),EXPARAES(20),EXPARPES(20),EXPARCLS(20),
     *         EXPARGAM(20)
c-------------------------------------------------------------------------------
c Set up all Variable arrays to be passed in labeled common block for ease of
c passing parameters between different subroutines
c-------------------------------------------------------------------------------
c
       Common /XYDATA/ XADATA,YADATA
       Common /EXPPAR/ EXPARSPT,EXPARMSC,EXPARSAM,EXPARELS,EXPAREDS,EXPARWDS,EXPARAES,
     *    EXPARPES,EXPARXRF,EXPARCLS,EXPARGAM
       Common /TEXT/ TITLE,DATE,TIME,OWNER,COMMENT,
     *    SIGNALTY,XUNITS,YUNITS,XLABEL,YLABEL,DATATYPE,
     *    OPERMODE,ELSDET,EDSDET
C
C
C
      Character*1 NUM,FKEY
```

```
      Character*5 KEY
      Character*4 WORD
      Character*2 ANS
      EQUIVALENCE ( EXPARSPT(1),VERSION)
      EQUIVALENCE ( EXPARSPT(2),NPOINTS)
      EQUIVALENCE ( EXPARSPT(3),NCOLUMNS)
      EQUIVALENCE ( EXPARSPT(4),XPERCHAN)
      EQUIVALENCE ( EXPARSPT(5),OFFSET)
      EQUIVALENCE ( EXPARSPT(6),CHOFFSET)
      EQUIVALENCE ( EXPARMSC(1),BEAMKV)
      EQUIVALENCE ( EXPARMSC(2),EMISSION)
      EQUIVALENCE ( EXPARMSC(3),PROBECUR)
      EQUIVALENCE ( EXPARMSC(4),BEAMDIAM)
      EQUIVALENCE ( EXPARMSC(5),MAGCAM)
      EQUIVALENCE ( EXPARMSC(6),CONVANGLE)
      EQUIVALENCE ( EXPARMSC(7),COLLANGLE)
      EQUIVALENCE ( EXPARSAM(1),THICKNESS)
      EQUIVALENCE ( EXPARSAM(2),XTILTSTGE)
      EQUIVALENCE ( EXPARSAM(3),YTILTSTGE)
      EQUIVALENCE ( EXPARSAM(4),XPOSITION)
      EQUIVALENCE ( EXPARSAM(5),YPOSITION)
      EQUIVALENCE ( EXPARSAM(6),ZPOSITION)
      EQUIVALENCE ( EXPARELS(1),DWELLTIME)
      EQUIVALENCE ( EXPARELS(2),INTEGTIME)
      EQUIVALENCE ( EXPAREDS(1),ELEVANGLE)
      EQUIVALENCE ( EXPAREDS(2),AZIMANGLE)
      EQUIVALENCE ( EXPAREDS(3),SOLIDANGLE)
      EQUIVALENCE ( EXPAREDS(4),LIVETIME)
      EQUIVALENCE ( EXPAREDS(5),REALTIME)
      EQUIVALENCE ( EXPAREDS(6),TBEWIND)
      EQUIVALENCE ( EXPAREDS(7),TAUWIND)
      EQUIVALENCE ( EXPAREDS(8),TDEADLYR)
      EQUIVALENCE ( EXPAREDS(9),TACTLYR)
      EQUIVALENCE ( EXPAREDS(10),TALWIND)
      EQUIVALENCE ( EXPAREDS(11),TPYWIND)
      EQUIVALENCE ( EXPAREDS(12),TBNWIND)
      EQUIVALENCE ( EXPAREDS(13),TDIWIND)
      EQUIVALENCE ( EXPAREDS(14),THCWIND)
c
c
c     NOW define different arrays used only in this routine
c
c---------------------------------------------------------------------
c
c
c     NOW define different arrays used only in this routine
c
c     XVALUE & YVALUE are temporary arrays used for reading in values
c
      REAL XVALUE(10),YVALUE(10)
c
c
c The array FNAME = Name of File to be read, Usually null terminated
c so we will initialize it with zero's using a Do loop so that successive
c operations of the routine within a single program donot cause problems
c if a short filename is used.
c
c FNAME   = Byte array of 17 characters containing the filename
c KText  = Byte array of 64 characters containing Keyword text
c
c
```

```
      Character*17 FNAME
      Character*1 KTEXT(64)
      Logical*1 TF
c
c Initialize Filename
c
c
c
c
c Begin by asking the user to supply a file name
c Note the following statement is generally compiler dependant
c Note this version limits file name to 16 characters
c
c
109    WRITE (6,110)
110    FORMAT (' Enter name of file [DEV:NAME.EXT]=> ')
      READ (5,120) FNAME
120    FORMAT(1A16)
c
c     check to see if the file exists
c
     INQUIRE (FILE=FNAME,EXIST=TF)
     IF(TF) THEN
c
c          It exists so now open the file
c
               OPEN (UNIT=99,STATUS='OLD',BLANK='NULL',
     *              ACCESS='SEQUENTIAL',FILE=FNAME)
          ELSE
c
c          It does not exist so tell the user
c
               WRITE (6,*) 'FILENAME Does Not Exit: Re-enter'
               Go to 109
          ENDIF
c
c     Begin reading the input file by inputing any parameters from
c     the file header which match the definitions of EMSA/MAS
c
c     Now lets read  the header defined by the EMMPDL
c     Note: some of the EMMPDL keywords are not defined in the
c     EMSA/MAS headers, also some have different names for the
c     than EMSA/MAS, correct this in the read routine and store
c     the parameters as their corresponding EMSA/MAS counterparts.
c
c
c First read the Spectrum Title
c
      READ (99,125) NUM,KEY,WORD,KTEXT
125    FORMAT (1A1,1A5,1A4,64A1)
C
C Check to see if it is legal
c
      IF (NUM.NE.'#')  GO TO  991
      IF (KEY.NE.'Title') GO TO 991
c
c          We have a legal format title write it out
c
      WRITE (6,125) NUM,KEY,WORD,KTEXT
c
c now the emmpdl version
```

```
c
      READ (99,130) NUM,KEY,WORD,VER
      IF (NUM.NE.'#')  GO TO 991
      IF (KEY.NE.'Versi') GO TO 991
      WRITE (6,1301) NUM,KEY,WORD,VER
c
c      Compare the version number are we EMMPDL version 1.1?
c      if not you need a different routine
c
c
      IF (VER.NE.1.1) GO TO 991
130    FORMAT(1A1,1A5,1A4,1G12.6)
1301   FORMAT(1A1,1A5,1A4,1G12.6)
c
c Okay we have a legal Version number
c we now read in all keywords as predefined by EMMPDL
c Each will be checked individually for safety, rather that
c assume the data file is perfect.
c
c
c Does the data start? Read the next line
c write it out regardless of its value and
c but look for the keyword "SPECTRUM"
c
      NUMKEY=2
129    READ (99,125) NUM,KEY,WORD,KTEXT
      IF(KEY.EQ.'SPECT') THEN
            WRITE (6,125) NUM,KEY,WORD,KTEXT
            GO TO 1000
            ENDIF
c
c the key word not SPECT, hence there is more information
c lets read it.
c
c       first check to see how many keywords we have read in
c       if greater than 20 there is an error
c
       NUMKEY=NUMKEY+1
       IF(NUMKEY.GT.20) GO TO 991
c
c First Backspace and compare with defined Keys,
c then read the value and store it as appropriate
c
c
      BACKSPACE (99)
c
c
c
c now test each predefined keyword
c
       FKEY='F'
c
      IF(KEY.EQ.'NPTS-') THEN
            READ (99,130) NUM,KEY,WORD,NPOINTS
            WRITE (6,1301) NUM,KEY,WORD,NPOINTS
                IF(NPOINTS.GT.4096) GO TO 991
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'NCOL-') THEN
            READ (99,130) NUM,KEY,WORD,NCOLUMNS
```

```
            WRITE (6,1301) NUM,KEY,WORD,NCOLUMNS
                 IF (NCOLUMNS.GT.5) GO TO 991
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'EVCH-') THEN
            READ (99,130) NUM,KEY,WORD,XPERCHAN
            WRITE (6,1301) NUM,KEY,WORD,XPERCHAN
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'OFFS-') THEN
            READ (99,130) NUM,KEY,WORD,OFFSET
            WRITE (6,1301) NUM,KEY,WORD,OFFSET
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'VOLT-') THEN
            READ (99,130) NUM,KEY,WORD,BEAMKV
            WRITE (6,1301) NUM,KEY,WORD,BEAMKV
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'ALPH-') THEN
            READ (99,130) NUM,KEY,WORD,CONVANGLE
            WRITE (6,1301) NUM,KEY,WORD,CONVANGLE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'BETA-') THEN
            READ (99,130) NUM,KEY,WORD,COLLANGLE
            WRITE (6,1301) NUM,KEY,WORD,COLLANGLE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'THCK-') THEN
            READ (99,130) NUM,KEY,WORD,THICKNESS
            WRITE(6,1301) NUM,KEY,WORD,THICKNESS
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'BCUR-') THEN
            READ (99,130) NUM,KEY,WORD,PROBECUR
            WRITE (6,1301) NUM,KEY,WORD,PROBECUR
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'BDIA-') THEN
            READ (99,130) NUM,KEY,WORD,BEAMDIAM
            WRITE (6,1301) NUM,KEY,WORD,BEAMDIAM
            FKEY='T'
            GO TO 129
            ENDIF
c
c Notice if the keyword is not defined above it is ignored!!!!
c now we will write it out anyway to show the user what it was
c
c       IF (FKEY.EQ.'F') THEN
            READ (99,125) NUM,KEY,WORD,KTEXT
            WRITE (6,1251) NUM,KEY,WORD,NUM,KEY,WORD,KTEXT
1251          FORMAT (' !!!! Warning Unknown or User keyword = ',
```

```
    *         1A1,1A5,1A9,/,1X,1A1,1A5,1A9,64A1)
c              endif
c now that we have decoded the keyword and its value
c we start over  with the loop until the keyword = SPECT
c

      GO TO 129
C
C We have found the keyword spectrum begin to read the data now.
c
1000   CONTINUE

       WRITE(6,1100)
1100    FORMAT(1X,14('='),'Reading Spectral Data',14('='))
C
C     EMMPDL FILES ARE ALWAYS 'Y' datatype SO SET THIS PARAMETER
c
      DATATYPE(1) = 'Y'
      DATATYPE(2) = ' '
c
c      calculate number of lines of data in the file
c
      NLINES=INT(NPOINTS/NCOLUMNS)
      NCOLS=INT(NCOLUMNS)
      NPTS=INT(NPOINTS)
c
c      Begin reading in data from file
c
2000   CONTINUE
c
c
c
      DO  2010 I=1,NLINES
      READ (99,*,ERR=992) (YVALUE(J), J=1,NCOLS)
d       WRITE (6,2011) (YVALUE(J), J=1,NCOLS)
d2011    FORMAT (1X,10F15.0)
2020   FORMAT (1F15.5)
c
c      now transfer this to the X&Y data arrays
c      remember since this is Y axis data only we have to
c      calcuated the X axis values using the parameters
c      passed by the header
c
      DO 2030 J=1,NCOLS
      ICHN = (I-1)*NCOLS + J
      YADATA(ICHN)=YVALUE(J)
      XADATA(ICHN)= OFFSET + ICHN*XPERCHAN
2030   CONTINUE
2010   CONTINUE
C
C
C
c
c       now check to see that all data has been read
c       is NCOLS*NLINES=NPOINTS?
c
       ICHN=NCOLS*NLINES
       IF (ICHN.LT.NPOINTS) THEN
         READ (99,*,ERR=992) (YADATA(K+ICHN),K=1,NPOINTS-ICHN)
           DO 2040    K=1,NPOINTS-ICHN
2040        XADATA(ICHN+K)= OFFSET + (ICHN+K)*XPERCHAN
```

```
       ENDIF

c
c      all the data should now be in the arrays
c      check for end of file marker
c

      READ (99,125) NUM,KEY,WORD,KTEXT
      WRITE (6,125) NUM,KEY,WORD,KTEXT
      IF(KEY.NE.'ENDOF') GO TO 991
       WRITE (6,5001)
5001    FORMAT(1X,14('='),'Data  Input  Complete',14('='))
       CLOSE (UNIT=99)
c
c      we found the correct end of file ask the
c      user if they want to write the data on the screen
c       the default will be YES
c
      WRITE(6,5010)
5010   FORMAT(
     *' Do you want to print the data on your screen [Y,N]?=>')
       READ (5,130) ANS
       IF ((ANS.EQ.'N').OR.(ANS.EQ.'n')) GO TO 6000
c
c      user wants to print out the data file
c
      WRITE (6,5015)
5015   FORMAT('   X-AXIS DATA     Y-AXIS DATA'
     *        ,/,31('-'),/)
      DO 5020 I=1,INT(NPOINTS)
      WRITE (6,5030) XADATA(I),YADATA(I)
5030   FORMAT(1X,1G12.5,4X,1G12.6)
5020   CONTINUE
6000   RETURN
c
c
c
991     CONTINUE
C
C There was a problem in a defined keyword or too many keywords
c
       WRITE (6,9911) NUM,KEY,WORD,NUM,KEY,WORD,KTEXT
9911    FORMAT (//,' WARNING: Problem in Keyword =',1A1,1A5,1A9,
     *         /, ' CHECK YOUR DATA FILE AT THE FOLLOWING LINE',
     *         /,1X,1A1,1A5,1A9,64A1)
       RETURN
992     CONTINUE
C
C       There was a problem with the data
c
       WRITE (6,9922) ICHN,XADATA(ICHN),YADATA(ICHN)
9922    FORMAT (//,
     *  ' WARNING: Problem with spectral data AFTER point #= ',I5,//,
     *  ' Values of last good X&Y DATA are: ',1x,2(1G12.6,','),/,
     *  ' Check your data file',//,' SPECTRAL INPUT TERMINATED',//)
       RETURN
       END


C
C=======================================================================
```

```
      Subroutine RDSYPARM
C=======================================================================
c
c
c This subroutine  READS a default file called
c SYSPARM.DAT which contains default settings for the current
c instrumental conditions and parameters defined in the EMSA/MAS format
c This file may be customized by the user to reflect his/her operating
c conditions. The routine also allows individuals to have multiple
c files which reflect different operating modes to save reentering data.
c The results  from the file are stored locally in predefined
c arrays. These arrays can be passed directly back to the main
c program and can be used to redirect data elsewhere as needed.
c In this implementation the arrays are passed in labeled common
c blocks. They can alternatively be passed as subroutine
c arguments.
c
c REVISION HISTORY
c
c Version  Date  Initials  Reason
c-----------------------------------------------------------------------
c
c 1.0   10/01/91  NJZ    Creation of the World WARNING THIS VERSION
c                     Case Sensitive!!! Until DeBugged
c
c-----------------------------------------------------------------------
c Definitions of Arrays used in this Subroutine
c
c
c
c EXPARSPT = Real Array <=20 values containing Expt Parameters of Spectrum
c EXPARMSC = Real Array <=20 values containing Expt Parameters of Microscope
c EXPARSAM = Real Array <=20 values containing Expt Parameters of Sample
c EXPAREDS = Real Array <=20 values containing Expt Parameters of EDS
c EXPARELS = Real Array <=20 values containing Expt Parameters of ELS
c EXPARAES = Real Array <=20 values containing Expt Parameters of AES
c EXPARWDS = Real Array <=20 values containing Expt Parameters of WDS
c EXPARPES = Real Array <=20 values containing Expt Parameters of PES
c EXPARXRF = Real Array <=20 values containing Expt Parameters of XRF
c EXPARCLS = Real Array <=20 values containing Expt Parameters of CLS
c EXPARGAM = Real Array <=20 values containing Expt Parameters of GAM
c SIGNALTY = Byte Array of 3 characters describing the signal type
c DATATYPE = Byte Array of 3 characters describing the format
c OPERMODE = Byte Array of 5 characters describing
c           the instrument operating mode
c EDSDET   = Byte Array of 5 characters describing the EDS detector type
c ELSDET   = Byte Array of 5 characters describing the ELS detector type
c XUNITS   = Byte Array <= 64 characters describing the X-Axis Data units
c YUNITS   = Byte Array <= 64 characters describing the Y-Axis Data units
c XLABEL   = Byte Array <= 64 characters describing the X-Axis Data label
c YLABEL   = Byte Array <= 64 characters describing the Y-Axis Data label
c TITLE    = Byte Array <= 64 characters with a spectrum Title
c DATE     = Byte Array of 12 characters with the date of acquisition
c           in the form DD-MMM-YYYY
c TIME     = Byte Array of 5 characters with the time of acquisition
c           in the form HH:MM
c OWNER    = Byte Array <= 64 characters with the owner/analysts name
c COMMENT  = Byte Array <= 64 Character*1 used to hold a text comment string
c
c-----------------------------------------------------------------------
c
```

```
c Current Definitions of EXPAR (Experimental Parameter Array) Values
c
c--------------------------------------------------------------------------
c  Parameters relating to the Spectrum Characteristics = EXPARSPT
c--------------------------------------------------------------------------
c
c Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
c Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),XLABEL(64),YLABEL(64),DATATYPE(2)
c
c EXPARSPT(1) = #VERSION  = File Format Version Number
c EXPARSPT(2) = #NPOINTS  = Total Number of Data Points in X&Y Data Arrays
c         1 <= NPOINTS <= 4096
c EXPARSPT(3) = #NCOLUMNS = Number of columns of data
c         1 <= NCOLUMNS <= 5
c EXPARSPT(4) = #XPERCHAN = Increment of X-axis units per channel (eV/Channel)
c           This is only useful if (x,y) paired data are not provided
c EXPARSPT(5) = #OFFSET   = Energy value of first data point in eV
c           This is only useful if (x,y) paired data are not provided
c EXPARSPT(6) = #CHOFFSET = Channel number which corresponds to zero units along
c            x-axis, this may be either a positive or negative value
C SIGNALTY(.) = #SIGNALTYPE = Type of Spectroscopy
c       EDS = Energy Dispersive Spectroscopy
c       WDS = Wavelength Dispersive Spectroscopy
c       ELS = Energy Loss Spectroscopy
c       AES = Auger Electron Spectroscopy
c       PES = Photo Electron Spectroscopy
c       XRF = X-ray Fluorescence Spectroscopy
c       CLS = Cathodoluminescence Spectroscopy
c       GAM = Gamma Ray Spectroscopy
c XUNITS(.)   = #XUNITS = Up to 64 characters describing the X-Axis Data units
c YUNITS(.)   = #YUNITS = Up to 64 characters describing the Y-Axis Data units
c XLABEL(.)   = #XLABEL = Up to 64 characters describing the X-Axis Data label
c YLABEL(.)   = #YLABEL = Up to 64 characters describing the Y-Axis Data label
C DATATYPE(.) = #DATATYPE = Type of data format
c          Y = Spectrum Y axis data only, X-axis data to be calculated
c           using XPERCHAN and OFFSET and the following formulae
c            X = OFFSET + CHANNEL*XPERCHAN
c          XY = Spectral data is in the form of XY pairs
c
c
c--------------------------------------------------------------------------
c  Microscope/Microanalysis Instrument Parameters = EXPARMSC
c--------------------------------------------------------------------------
c
c Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,
c CONVANGLE,COLLANGLE
c Character*1 OPERMODE(5)
c
c EXPARMSC(1) = #BEAMKV    = Accelerating Voltage of Instrument in kV
c EXPARMSC(2) = #EMISSION   = Gun Emission current in microAmps
c EXPARMSC(3) = #PROBECUR   = Probe current in nanoAmps
c EXPARMSC(4) = #BEAMDIAM   = Diameter of incident probe in nanometers
c EXPARMSC(5) = #MAGCAM   = Magnification or Camera Length (Mag in x, Cl in mm)
c EXPARMSC(6) = #CONVANGLE  = Convergence semi-angle of incident beam in milliRadians
c EXPARMSC(7) = #COLLANGLE  = Collection semi-angle of scattered beam in milliRad
c OPERMODE(.) = #OPERMODE   = Operating Mode of Instrument
c                       IMAGE = Imaging Mode
c                       DIFFR = Diffraction Mode
c                       SCIMG = Scanning Imaging Mode
c                       SCDIF = Scanning Diffraction Mode
c
```

```
c--------------------------------------------------------------------------
c  Experimental Parameters relating to the Sample = EXPARSAM
c--------------------------------------------------------------------------
c
c Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
C      ZPOSITION
c
c EXPARSAM(1) = #THICKNESS = Specimen thickness in nanometers
c EXPARSAM(2) = #XTILTSTGE = Specimen stage tilt X-axis in degrees
c EXPARSAM(3) = #YTILTSTGE = Specimen stage tilt Y-axis in degrees
c EXPARSAM(4) = #XPOSITION =  X location of beam or specimen
c EXPARSAM(5) = #YPOSITION =  Y location of beam or specimen
c EXPARSAM(6) = #ZPOSITION =  Z location of beam or specimen
c
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to ELS = EXPARELS
c--------------------------------------------------------------------------
c
c Real EXPARELS(20),DWELLTIME,INTEGTIME
c Character*1 ELSDET(6)
c
c EXPARELS(1) = #DWELLTIME  = Dwell time per channel for serial data collection in msec
c EXPARELS(2) = #INTEGTIME  = Integration time per spectrum for parallel data collection
c            in milliseconds
c ELSDET(.)  = #ELSDET   = Type of ELS Detector
c            Serial = Serial ELS Detector
c            Parall = Parallel ELS Detector
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to EDS = EXPAREDS
c--------------------------------------------------------------------------
c
c Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
c Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
c Character*1 EDSDET(5)
c
c EXPAREDS(1) = #ELEVANGLE = Elevation angle of EDS,WDS detector in degrees
c EXPAREDS(2) = #AZIMANGLE = Azimuthal angle of EDS,WDS detector in degrees
c EXPAREDS(3) = #SOLIDANGLE = Collection solid angle of detector in sR
c EXPAREDS(4) = #LIVETIME  = Signal Processor Active (Live) time in seconds
c EXPAREDS(5) = #REALTIME  = Total clock time used to record the spectrum in seconds
c EXPAREDS(6) = #TBEWIND  = Thickness of Be Window on detector in cm
c EXPAREDS(7) = #TAUWIND  = Thickness of Au Window/Electrical Contact in cm
c EXPAREDS(8) = #TDEADLYR  = Thickness of Dead Layer in cm
c EXPAREDS(9) = #TACTLYR  = Thickness of Active Layer in cm
c EXPAREDS(10) = #TALWIND  = Thickness of Aluminium Window in cm
c EXPAREDS(11) = #TPYWIND  = Thickness of Pyrolene Window in cm
c EXPAREDS(12) = #TBNWIND  = Thickness of Boron-Nitride Window in cm
c EXPAREDS(13) = #TDIWIND  = Thickness of Diamond Window in cm
c EXPAREDS(14) = #THCWIND  = Thickness of HydroCarbon Window in cm
c EDSDET(.)    = #EDSDET  = Type of X-ray Detector
c        SIBEW = Si(Li) with Be Window
c        SIUTW = Si(Li) with Ultra Thin Window
c        SIWLS = Si(Li) Windowless
c        GEBEW = Ge with Be Window
c        GEUTW = Ge with Ultra Thin Window
c        GEWLS = Ge Windowless
c
c--------------------------------------------------------------------------
c  Experimental Parameters relating mainly to WDS = EXPARWDS
```

```
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to XRF = EXPARXRF
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to AES = EXPARAES
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to PES = EXPARPES
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to CLS = EXPARCLS
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c-----------------------------------------------------------------------
c  Experimental Parameters relating mainly to GAM = EXPARGAM
c-----------------------------------------------------------------------
c
c     Nothing currently defined
c
c=======================================================================
c        END OF DEFINITIONS
c=======================================================================
c
c
c Start Code
c
c
c   Dimension all arrays, & Declare Variable Types
c   REMEMBER we have no integer parameters!!!!!
c
c
      Character*1 TITLE(64),DATE(12),TIME(5),OWNER(64),COMMENT(64)
      Real XADATA(4096),YADATA(4096)
      Real EXPARSPT(20),VERSION,NPOINTS,NCOLUMNS,XPERCHAN,OFFSET,CHOFFSET
      Character*1 SIGNALTY(3),XUNITS(64),YUNITS(64),
     *        XLABEL(64),YLABEL(64),DATATYPE(2)
      Real EXPARMSC(20),BEAMKV,EMISSION,PROBECUR,BEAMDIA,MAGCAM,CONVANGLE,
     *      COLLANGLE
      Character*1 OPERMODE(5)
      Real EXPARSAM(20),THICKNESS,XTILTSTGE,YTILTSTGE,XPOSITION,YPOSITION,
     *      ZPOSITION
      Real EXPARELS(20),DWELLTIME,INTEGTIME
      Character*1 ELSDET(6)
      Real EXPAREDS(20),ELEVANGLE,AZIMANGLE,SOLIDANGLE,LIVETIME,REALTIME
      Real TBEWIND,TAUWIND,TDEADLYR,TACTLYR,TALWIND,TPYWIND,TDIWIND,THCWIND
      Character*1 EDSDET(5)
```

```
      Real EXPARWDS(20),EXPARXRF(20),EXPARAES(20),EXPARPES(20),EXPARCLS(20),
     *       EXPARGAM(20)
c-------------------------------------------------------------------------------
c Set up all Variable arrays to be passed in labeled common block for ease of
c passing parameters between different subroutines
c-------------------------------------------------------------------------------
c
      Common /XYDATA/ XADATA,YADATA
      Common /EXPPAR/ EXPARSPT,EXPARMSC,EXPARSAM,EXPARELS,EXPAREDS,EXPARWDS,EXPARAES,
     *    EXPARPES,EXPARXRF,EXPARCLS,EXPARGAM
      Common /TEXT/ TITLE,DATE,TIME,OWNER,COMMENT,
     *    SIGNALTY,XUNITS,YUNITS,XLABEL,YLABEL,DATATYPE,
     *    OPERMODE,ELSDET,EDSDET
c
      Character*1 ANS,NUM,FKEY
      Character*5 KEY
      Character*9 WORD
      EQUIVALENCE ( EXPARSPT(1),VERSION)
      EQUIVALENCE ( EXPARSPT(2),NPOINTS)
      EQUIVALENCE ( EXPARSPT(3),NCOLUMNS)
      EQUIVALENCE ( EXPARSPT(4),XPERCHAN)
      EQUIVALENCE ( EXPARSPT(5),OFFSET)
      EQUIVALENCE ( EXPARSPT(6),CHOFFSET)
      EQUIVALENCE ( EXPARMSC(1),BEAMKV)
      EQUIVALENCE ( EXPARMSC(2),EMISSION)
      EQUIVALENCE ( EXPARMSC(3),PROBECUR)
      EQUIVALENCE ( EXPARMSC(4),BEAMDIAM)
      EQUIVALENCE ( EXPARMSC(5),MAGCAM)
      EQUIVALENCE ( EXPARMSC(6),CONVANGLE)
      EQUIVALENCE ( EXPARMSC(7),COLLANGLE)
      EQUIVALENCE ( EXPARSAM(1),THICKNESS)
      EQUIVALENCE ( EXPARSAM(2),XTILTSTGE)
      EQUIVALENCE ( EXPARSAM(3),YTILTSTGE)
      EQUIVALENCE ( EXPARSAM(4),XPOSITION)
      EQUIVALENCE ( EXPARSAM(5),YPOSITION)
      EQUIVALENCE ( EXPARSAM(6),ZPOSITION)
      EQUIVALENCE ( EXPARELS(1),DWELLTIME)
      EQUIVALENCE ( EXPARELS(2),INTEGTIME)
      EQUIVALENCE ( EXPAREDS(1),ELEVANGLE)
      EQUIVALENCE ( EXPAREDS(2),AZIMANGLE)
      EQUIVALENCE ( EXPAREDS(3),SOLIDANGLE)
      EQUIVALENCE ( EXPAREDS(4),LIVETIME)
      EQUIVALENCE ( EXPAREDS(5),REALTIME)
      EQUIVALENCE ( EXPAREDS(6),TBEWIND)
      EQUIVALENCE ( EXPAREDS(7),TAUWIND)
      EQUIVALENCE ( EXPAREDS(8),TDEADLYR)
      EQUIVALENCE ( EXPAREDS(9),TACTLYR)
      EQUIVALENCE ( EXPAREDS(10),TALWIND)
      EQUIVALENCE ( EXPAREDS(11),TPYWIND)
      EQUIVALENCE ( EXPAREDS(12),TBNWIND)
      EQUIVALENCE ( EXPAREDS(13),TDIWIND)
      EQUIVALENCE ( EXPAREDS(14),THCWIND)
c
c
c     NOW define different arrays used only in this routine
c
c
c The array FNAME = Name of File to be read, Usually null terminated
c so we will initialize it with zero's using a Do loop so that successive
c operations of the routine within a single program donot cause problems
c if a short filename is used.
```

```
c
c FNAME   = Byte array of 17 characters containing the filename
c KText  = Byte array of 64 characters containing Keyword text
c
c
      Character*17 FNAME,DNAME
      Character*1 KTEXT(64)
      Logical*1 TF
c
c Initialize Default Filename
c
c
     DNAME='SYSPARM.DAT'
c
c
c
109       WRITE (6,110)
110    FORMAT (
    *   ' Enter name of a System Parameters file [SYSPARM.DAT]=> ')
      READ (5,120) FNAME
120    FORMAT(1A16)
          IF (FNAME.EQ.' ') THEN
               FNAME=DNAME
               ENDIF
c
c     check that the file exists
c
     INQUIRE (FILE=FNAME,EXIST=TF)
     IF (TF) THEN
c
c          Now open the file
c
          OPEN (UNIT=99,STATUS='OLD',BLANK='NULL',
    *          ACCESS='SEQUENTIAL',FILE=FNAME)
          ELSE
               WRITE (6,*) 'FILENAME Does Not Exist: Re-enter'
               GO TO 109
          ENDIF

c
c Now lets read and write the defined parameters as defined by EMSA/MAS
c
c
c
     WRITE (6,1200) FNAME
1200  FORMAT (' Current System Parameters in File : ',1a16,/
    *  ' ======================================================',/,
    *  ' The System Parameters marked  with a (*) are updated',/,
    *  '    when you WRITE an EMSA/MAS Spectral Data File.',//,
    *  ' Use the MODIFY Parameters Option of the Main Menu to',/,
    *  ' change data independent parameters or Quit this program',/,
    *  ' and use the TEXT EDITOR supplied with your computer.',/,
    *  ' ======================================================',/)
c
c First read the Format Title
c
       READ (99,125) NUM,KEY,WORD,KTEXT
125     FORMAT (1A1,1A5,1A9,64A1)
1251    FORMAT (1x,' ',1A5,1A9,64A1)
1252    FORMAT (1x,'*',1A5,1A9,64A1)
1253    FORMAT (1X,'>',1A5,1A9,64A1)
```

```
C
C Check to see if it is legal
c
      IF (NUM.NE.'#')  GO TO  991
      IF (KEY.NE.'FORMA') GO TO 991
c
c            We have a legal format title write it out
c
      WRITE (6,1251) KEY,WORD,KTEXT
c
c now the version
c
      READ (99,130) NUM,KEY,WORD,VERSION
      IF (NUM.NE.'#')  GO TO 991
      IF (KEY.NE.'VERSI') GO TO 991
      WRITE (6,1301) KEY,WORD,VERSION
c
c
c
130    FORMAT(1A1,1A5,1A9,1F20.4)
1301   FORMAT(1x,' ',1A5,1A9,1G15.3)
1302   FORMAT(1x,'*',1A5,1A9,1G15.3)
1303   FORMAT(1X,'>',1A5,1A9,1G15.3)
c
c     The system parameters file should be customized by the user
c     to reflect their standard conditions. The user should
c     use a simple screen editor to update the default file
c     provided. We will read all the parameters in the file
c     and if they correspond to a defined EMSA/MAS keyword then
c     store the value. If they are not standard then they will
c     be simply typed on the screen. Some of these parameters
c     will logically be updated when a EMSA/MAS file is written
c     to disk. These are indicated by a * when typed on the screen.
c
c     Note: the System Parameters file may not have all the
c     predefined EMSA/MAS keywords, but we must check for all
c     of them regardless.
c
      NUMKEY=2
129    READ (99,125,end=1000) NUM,KEY,WORD,KTEXT
      IF(KEY.EQ.'SPECT') THEN
            WRITE (6,1251) KEY,WORD,KTEXT
            GO TO 1000
            ENDIF
c
c the key word not SPECT, hence there is more information
c lets read it.
c
c       first check to see how many keywords we have read in
c       if greater than 100 there is probably an error
c
       NUMKEY=NUMKEY+1
       IF(NUMKEY.GT.100) GO TO 991
c
c First Backspace and compare with defined Keys,
c then read the value and store it as appropriate
c
c
      BACKSPACE (99)
c
c
```

```
c
c now test each predefined keyword
c
      FKEY='F'
c
c       define a false flag key FKEY to indicate a nonEMSA/MAS
c               keyword in the input. This word will be printed
c               on the screen but not used in the program.
c
      IF(KEY.EQ.'TITLE') THEN
            READ (99,125) NUM,KEY,WORD,TITLE
            WRITE (6,1252) KEY,WORD,TITLE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'DATE ') THEN
            READ (99,125) NUM,KEY,WORD,DATE
            WRITE (6,1252) KEY,WORD,DATE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TIME') THEN
            READ (99,125) NUM,KEY,WORD,TIME
            WRITE (6,1252) KEY,WORD,TIME
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'OWNER') THEN
            READ (99,125) NUM,KEY,WORD,OWNER
            WRITE (6,1251) KEY,WORD,OWNER
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'NPOIN') THEN
            READ (99,130) NUM,KEY,WORD,NPOINTS
            WRITE (6,1302) KEY,WORD,NPOINTS
                IF(NPOINTS.GT.4096) GO TO 991
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'NCOLU') THEN
            READ (99,130) NUM,KEY,WORD,NCOLUMNS
            WRITE (6,1302) KEY,WORD,NCOLUMNS
                  IF (NCOLUMNS.GT.5) GO TO 991
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'XUNIT') THEN
            READ (99,125) NUM,KEY,WORD,XUNITS
            WRITE (6,1251) KEY,WORD,XUNITS
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'YUNIT') THEN
            READ (99,125) NUM,KEY,WORD,YUNITS
            WRITE (6,1251) KEY,WORD,YUNITS
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'DATAT') THEN
            READ (99,125) NUM,KEY,WORD,DATATYPE
```

```
            WRITE (6,1252) KEY,WORD,DATATYPE
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'XPERC') THEN
            READ (99,130) NUM,KEY,WORD,XPERCHAN
            WRITE (6,1302) KEY,WORD,XPERCHAN
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'OFFSE') THEN
            READ (99,130) NUM,KEY,WORD,OFFSET
            WRITE (6,1302) KEY,WORD,OFFSET
            FKEY='T'
            GO TO 129
            ENDIF
c
c     The above are the required keywords
c     the following are optional keywords

      IF(KEY.EQ.'CHOFF') THEN
            READ (99,130) NUM,KEY,WORD,CHOFFSET
            WRITE (6,1302) KEY,WORD,CHOFFSET
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'SIGNA') THEN
            READ (99,125) NUM,KEY,WORD,SIGNALTY
            WRITE (6,1251) KEY,WORD,SIGNALTY
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'XLABE') THEN
            READ (99,125) NUM,KEY,WORD,XLABEL
            WRITE (6,1251) KEY,WORD,XLABEL
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'YLABE') THEN
            READ (99,125) NUM,KEY,WORD,YLABEL
            WRITE (6,1251) KEY,WORD,YLABEL
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'COMME') THEN
            READ (99,125) NUM,KEY,WORD,COMMENT
            WRITE (6,1251) KEY,WORD,COMMENT
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'BEAMK') THEN
            READ (99,130) NUM,KEY,WORD,BEAMKV
            WRITE (6,1301) KEY,WORD,BEAMKV
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'EMISS') THEN
            READ (99,130) NUM,KEY,WORD,EMISSION
            WRITE(6,1301) KEY,WORD,EMISSION
            FKEY='T'
            GO TO 129
```

```
      ENDIF
IF(KEY.EQ.'PROBE') THEN
      READ (99,130) NUM,KEY,WORD,PROBECUR
      WRITE (6,1301) KEY,WORD,PROBECUR
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'BEAMD') THEN
      READ (99,130) NUM,KEY,WORD,BEAMDIAM
      WRITE (6,1301) KEY,WORD,BEAMDIAM
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'MAGCA') THEN
      READ (99,130) NUM,KEY,WORD,MAGCAM
      WRITE (6,1301) KEY,WORD,MAGCAM
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'CONVA') THEN
      READ (99,130) NUM,KEY,WORD,CONVANGLE
      WRITE (6,1301) KEY,WORD,CONVANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'COLLA') THEN
      READ (99,130) NUM,KEY,WORD,COLLANGLE
      WRITE (6,1301) KEY,WORD,COLLANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'OPERM') THEN
      READ (99,125) NUM,KEY,WORD,OPERMODE
      WRITE(6,1251) KEY,WORD,OPERMODE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'THICK') THEN
      READ (99,130) NUM,KEY,WORD,THICKNESS
      WRITE(6,1301) KEY,WORD,THICKNESS
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'XTILT') THEN
      READ (99,130) NUM,KEY,WORD,XTILTSTGE
      WRITE(6,1301) KEY,WORD,XTILTSTGE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'YTILT') THEN
      READ (99,130) NUM,KEY,WORD,YTILTSTGE
      WRITE(6,1301) KEY,WORD,YTILTSTGE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'XPOSI') THEN
      READ (99,130) NUM,KEY,WORD,XPOSITION
      WRITE(6,1301) KEY,WORD,XPOSITION
      FKEY='T'
      GO TO 129
      ENDIF
```

```
IF(KEY.EQ.'YPOSI') THEN
      READ (99,130) NUM,KEY,WORD,YPOSITION
      WRITE(6,1301) KEY,WORD,YPOSITION
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'ZPOSI') THEN
      READ (99,130) NUM,KEY,WORD,ZPOSITION
      WRITE (6,1301) KEY,WORD,ZPOSITION
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'DWELL') THEN
      READ (99,130) NUM,KEY,WORD,DWELLTIME
      WRITE(6,1302) KEY,WORD,DWELLTIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'INTEG') THEN
      READ (99,130) NUM,KEY,WORD,INTEGTIME
      WRITE(6,1302) KEY,WORD,INTEGTIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'ELSDE') THEN
      READ (99,125) NUM,KEY,WORD,ELSDET
      WRITE (6,1251) KEY,WORD,ELSDET
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'ELEVA') THEN
      READ (99,130) NUM,KEY,WORD,ELEVANGLE
      WRITE(6,1301) KEY,WORD,ELEVANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'AZIMA') THEN
      READ (99,130) NUM,KEY,WORD,AZIMANGLE
      WRITE(6,1301) KEY,WORD,AZIMANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'SOLID') THEN
      READ (99,130) NUM,KEY,WORD,SOLIDANGLE
      WRITE(6,1301) KEY,WORD,SOLIDANGLE
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'LIVET') THEN
      READ (99,130) NUM,KEY,WORD,LIVETIME
      WRITE(6,1302) KEY,WORD,LIVETIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'REALT') THEN
      READ (99,130) NUM,KEY,WORD,REALTIME
      WRITE (6,1302) KEY,WORD,REALTIME
      FKEY='T'
      GO TO 129
      ENDIF
IF(KEY.EQ.'TBEWI') THEN
```

```
            READ (99,130) NUM,KEY,WORD,TBEWIND
            WRITE(6,1301) KEY,WORD,TBEWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TAUWI') THEN
            READ (99,130) NUM,KEY,WORD,TAUWIND
            WRITE (6,1301) KEY,WORD,TAUWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TDEAD') THEN
            READ (99,130) NUM,KEY,WORD,TDEADLYR
            WRITE (6,1301) KEY,WORD,TDEADLYR
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TACTL') THEN
            READ (99,130) NUM,KEY,WORD,TACTLYR
            WRITE (6,1301) KEY,WORD,TACTLYR
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TALWI') THEN
            READ (99,130) NUM,KEY,WORD,TALWIND
            WRITE(6,1301) KEY,WORD,TALWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TPYWI') THEN
            READ (99,130) NUM,KEY,WORD,TPYWIND
            WRITE(6,1301) KEY,WORD,TPYWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TBNWI') THEN
            READ (99,130) NUM,KEY,WORD,TBNWIND
            WRITE (6,1301) KEY,WORD,TBNWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'TDIWI') THEN
            READ (99,130) NUM,KEY,WORD,TDIWIND
            WRITE(6,1301) KEY,WORD,TDIWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'THCWI') THEN
            READ (99,130) NUM,KEY,WORD,THCWIND
            WRITE (6,1301) KEY,WORD,THCWIND
            FKEY='T'
            GO TO 129
            ENDIF
      IF(KEY.EQ.'EDSDE') THEN
            READ (99,125) NUM,KEY,WORD,EDSDET
            WRITE(6,1251) KEY,WORD,EDSDET
            FKEY='T'
            GO TO 129
            ENDIF
c
c Notice if the keyword is not defined above it is ignored!!!!
```

```
c now we will write it out anyway to show the user what it was
c
c       IF (FKEY.EQ.'F') THEN
             READ (99,125) NUM,KEY,WORD,KTEXT
             WRITE (6,2253) NUM,KEY,WORD,NUM,KEY,WORD,KTEXT
2253           FORMAT (' !!!! Warning Unknown or User keyword = ',
     *         1A1,1A5,1A9,/,1X,1A1,1A5,1A9,64A1)
c             endif
c now that we have decoded the keyword and its value
c we start over  with the loop until the keyword = SPECT
c

      GO TO 129
C
C We have found the end of the parameter file.
c
1000   CONTINUE

       WRITE(6,1100)
1100    FORMAT(1X,14('='),'End of Parameter File',14('='),//)
6000  close (unit=99)
     RETURN
C
C
C
991     CONTINUE
C
C There was a problem in a defined keyword or too many keywords
c
       WRITE (6,9911) NUM,KEY,WORD,NUM,KEY,WORD,KTEXT
9911    FORMAT (//,' WARNING: Problem in Keyword =',1A1,1A5,1A9,
     *         /, ' CHECK YOUR DATA FILE AT THE FOLLOWING LINE',
     *         /,1X,1A1,1A5,1A9,64A1)
       RETURN
       END

c=======================================================================
     Subroutine TModify(A,B,N)
c=======================================================================
c     Revision History
c     ----------------
c     Version Initials Date   Reason
c     1.0      NJZ    10/91  Creation
c-----------------------------------------------------------------------
c     This subroutine is used to modify the value of the text parameter
c     B whose descriptor is A. N  is the number of characters in B
c=======================================================================
c
c     Start Code
c
      Character*1 ANS
      Character*15 A
      Character*1 B(64),TEMP(64)
c
c     Begin by writing things out on the screen
100   WRITE (6,101) A,(B(J), J=1,N)
101   FORMAT(1A15,64(A1:))
c
c     Now test if we should change B
      WRITE (6,120)
120   FORMAT (' New Text    : ')
```

```
      READ (6,130) (TEMP(J),J=1,N)
130   FORMAT (64(A1:))
c
c     Check the answer and do something
c     a blank line means leave the item alone
c     anything else means change it.
c
      IF (TEMP(1).EQ.' ')
     *    THEN
              GO TO 140
          ELSE
              DO 150 I=1,N
150             B(I)=TEMP(I)
c               WRITE (6,*)  A,(B(J),J=1,N)
          ENDIF
c
c     we are all done, let  another routine write to the data file.
c
140   Return
      End
c=======================================================================
      Subroutine VModify(A,B)
c=======================================================================
c     Revision History
c     ----------------
c     Version Initials Date   Reason
c     1.0      NJZ    10/91  Creation
c-----------------------------------------------------------------------
c     This subroutine is used to modify the value of the real
c     value parameter B whose descriptor is A.
c=======================================================================
c
c     Start Code
c
      Character*1 ANS
      Character*15 A
      REAL B,TEMP
c
c     Begin by writing things out on the screen
100   WRITE (6,101) A,B
101   FORMAT(1A15,1G12.6)
c
c     Now test if we should change B
119   WRITE (6,120)
120   FORMAT (' New Value   : ')
      READ (6,130,ERR=119) TEMP
130   FORMAT (1G15.0)
c
c     Check the answer and do something
c     a blank line means leave the item alone
c     anything else means change it.
c
      IF (TEMP.EQ.0)
     *    THEN
              GO TO 140
          ELSE
              B=TEMP
c               WRITE (6,101)  A,B
          ENDIF
c
c     we are all done, let another routine write to the data file.
```

```
c
140   Return
      End
c
c
c
c     Additional subroutines used by EMMFF
c
c
c
c=======================================================================
      SUBROUTINE TWRITE (A,B,N,FILE)
C=======================================================================
c
c     This simple routine replaces multiple typing of 2 write statements
c     for text characters output to the screen and to a file
c
c
      CHARACTER*15 A
      CHARACTER*1 B(64),TEMP,FILE
      INTEGER*2 N
c
c       If file = Y then write to the file
c       otherwise only to the screen

      IF (FILE.EQ.'Y') WRITE (98,126) A,(B(J),J=1,N)
      WRITE (6,126) A,(B(J),J=1,N)
126   FORMAT (1A15,64(A1:))
C1261  FORMAT (1A15,64(A1))
      RETURN
      END
c
c=======================================================================
      SUBROUTINE VWRITE (A,B,FILE)
C=======================================================================
c
c     This simple routine replaces multiple typing of 2 write statements
c     for 1 text & 1 numeric value on output to the screen and to a file
c
c
      CHARACTER*15 A
      CHARACTER*1 TEMP,FILE
      REAL B
c
c       If file = Y then write to a file
c
      IF (FILE.EQ.'Y') WRITE (98,126) A,B
      WRITE (6,126) A,B
126   FORMAT (1A15,1G12.6)
C1261  FORMAT (1A15,1G15.7)
      RETURN
      END
```