

A Generic JCAMP-DX Standard File Format, JCAMP-DX V.6.00

INTERNATIONAL UNION OF PURE
AND APPLIED CHEMISTRY



COMMITTEE ON PRINTED
AND ELECTRONIC PUBLICATIONS

Working Party on Spectroscopic Data Standards (JCAMP-DX)*

Prepared for publication by
P. Lampen^{1*}, R.J. Lancashire², R.S. McDonald³,
P.S. McIntyre⁴, D.N. Rutledge⁵, A.N. Davies⁶

¹ ISAS, *Inst. für Spektrochemie und Angewandte Spektroskopie, Bunsen-Kirchhoff-Str.11, 44139 Dortmund, Germany*

² *Department of Chemistry, University of the West Indies, Mona, Kingston 7, Jamaica*

³ *9 Woodside Dr., Burnt Hills, New York 12027, United States of America*

⁴ *School of Applied Sciences, University of Glamorgan, Pontypridd, Mid-Glamorgan, CF37 1DL, United Kingdom*

⁵ *Inst. National Agronomique, Lab. de Chimie Analytique, 16, rue Claude Bernard, 75231 Paris Cedex 05, France*

⁶ *ISAS, Inst. für Spektrochemie und Angewandte Spektroskopie, Bunsen-Kirchhoff-Str.11, 44139 Dortmund, Germany (Current address c/o CREON•LAB•CONTROL AG, Europaallee, 27-29 50226 Frechen, Germany.)*

*Author to whom comments should be sent.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 1/40

A Generic JCAMP-DX Standard File Format, JCAMP-DX V.6.00₀

P. Lampen^{1*}, R.J. Lancashire², R.S. McDonald³, P.S. McIntyre⁴, D.N. Rutledge⁵, A.N. Davies⁶

¹ *ISAS, Inst. für Spektrochemie und Angewandte Spektroskopie, Bunsen-Kirchhoff-Str.11, 44139 Dortmund, Germany*

² *Department of Chemistry, University of the West Indies, Mona, Kingston 7, Jamaica*

³ *9 Woodside Dr., Burnt Hills, New York 12027, United States of America*

⁴ *School of Applied Sciences, University of Glamorgan, Pontypridd, Mid-Glamorgan, CF37 1DL, United Kingdom*

⁵ *Inst. National Agronomique, Lab. de Chimie Analytique, 16, rue Claude Bernard, 75231 Paris Cedex 05, France*

⁶ *ISAS, Inst. für Spektrochemie und Angewandte Spektroskopie, Bunsen-Kirchhoff-Str.11, 44139 Dortmund, Germany (Current address c/o CREON•LAB•CONTROL AG, Europaallee, 27-29 50226 Frechen, Germany.)*

*Author to whom comments should be sent.

ABSTRACT

The increasing power of modern spectroscopic data systems has led to increasing use of multidimensional techniques. These lead to multidimensional data sets which need to be stored and exchanged between computer systems. This protocol will describe the generic form of the JCAMP-DX data storage format. It includes not only the means to efficiently store and exchange multidimensional data but the basic form for the production of future protocols for different spectroscopic types.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 2/40

1. SCOPE

JCAMP-DX is an evolving, open-ended, machine-independent, self-documenting file format for exchanging and archiving data from computerized laboratory instruments like spectrographs, diffractometers, and others whose output is commonly represented as spectral (profile) plots, contours, or peak tables. JCAMP-DX was originated to meet the need for exchanging infrared spectra between similar instruments of different manufacture. The present document is the result of ongoing efforts by users and manufacturers to extend JCAMP-DX to other types of instrumental data.

A major objective of JCAMP-DX is to enable routine capture of data at the source to make it available for exchange, archiving, and entry into databases.

The emphasis is on readability and flexibility rather than on size or speed for rapid search and retrieval. All data are represented as labelled fields of variable length using printable ASCII characters. A JCAMP-DX file is a text file which can be viewed, corrected, and annotated with ASCII text editors.

JCAMP-DX is compatible with all media: telephone, magnetic and optical disk, magnetic tape, and even the printed page (using a scanner).

The JCAMP-DX protocol is non-proprietary. These specifications are copyrighted by the International Union of Pure and Applied Chemistry (IUPAC) solely for the purpose of linking them with the name JCAMP-DX. The right to copy these specifications for scientific purposes is hereby granted.

Use of the name JCAMP-DX to describe data files implies that they conform to the form and style described in the relevant protocols and the information content defined for a particular DATATYPE.

Use of the name JCAMP-DX in the description of software capability implies the ability to generate and read JCAMP-DX files as defined in the relevant published protocols for a particular DATATYPE.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 3/40

2. SIGNIFICANCE AND USE

This version of JCAMP-DX provides a description of the file structure to be used to accommodate a very wide spectrum of applications and users.

The present effort is to focus the JCAMP-DX standard on definition of the generic form and style of representation of the various elements of data files, and to encourage user groups to define application specific details for their own techniques.

This publication restricts itself to technique independent generic JCAMP-DX labels, and providing mechanisms for user-groups to:

- 1) generate DATATYPE-SPECIFIC labels for their own special techniques and applications such as multi-dimensional NMR experiments, and
- 2) generate PRIVATE labels specific to their own working environment and not expected to be read outside this environment.

Inasmuch as it is very desirable for instrument data systems to be able to read and write files in a standard format directly, instrument vendors are encouraged to develop JCAMP-DX software

for the instruments which they currently support. It is feasible for vendors to do so for a tightly defined CORE of information which is described in Section 4.2.1.

The present version introduces a generalized data table form for multivariate data. The original published version of JCAMP-DX (Section 2.1) was focused mainly on bivariate (X,Y) data. The previously defined bivariate forms ##XYDATA=, and ##XYPOINTS=, are retained, together with the bivariate form of ##PEAKTABLE=. It is recommended that the remaining forms, ##PEAKTABLE=, ##ASSIGNMENTS=, and ##RADATA=, be replaced by corresponding multivariate forms.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 4/40

3. BASIC STRUCTURE OF JCAMP-DX FILES.

JCAMP-DX is a FILE specification. The basic element of JCAMP-DX files is the LABELLED-DATA-RECORD (LDR). LABELLED-DATA-RECORDS are combined into BLOCKS. A FILE may contain more than one BLOCK.

We will briefly mention here the multiblock structure which resembles a hierarchical directory structure with LINK BLOCKS functioning as directories, and DATA BLOCKS, as FILES. The BLOCK structure is described in Section 5.4. The remainder of this document will be devoted to complex JCAMP-DX files using the NTUPLES format which consist of multiple DATA PAGES.

This is what we will normally mean by the word FILE.

The logical division of a JCAMP-DX file into CORE and NOTES subsets is a measure to separate the protocol for representing tabular data (CORE) which must be parsed by computer from that which is mainly for reference by humans.

3.1. CORE

The CORE portion of a JCAMP-DX BLOCK or FILE is the irreducible minimum JCAMP-DX file. The CORE provides a focus for instrument and software vendors for converting between in-memory forms and JCAMP-DX. The CORE contains one or more data arrays, corresponding attributes for defining units, scale factors, sizes, etc., of arrays for the data, plus key application dependent attributes and identification. The main objective of the CORE is to focus on the information which must be transferred to a foreign system in order for it to process and/or plot and label the data as if it had been generated internally.

The CORE into two subsets: CORE HEADER and CORE DATA.

3.1.1. CORE HEADER

The CORE HEADER provides for overall identification of a JCAMP-DX file. It consists of:

TITLE,

JCAMP-DX VERSION NUMBER, comment on version of software writing the file

DATATYPE,

DATA CLASS,

NUMDIM, number of dimensions in a multi-dimensional data file

ORIGIN of the data,

OWNER of the data,

DATA CLASS, and NUMDIM are new LDRs which are described below.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 5/40

3.1.1.1. ##DATA CLASS= (STRING).

This LDR contains the name of the type of TABULAR DATA, i.e., XYDATA, XYPOINTS, PEAK TABLE, ASSIGNMENTS, RADATA, or NTUPLES. Coates (Cited in Section 2.3) first pointed out that the data for a number of different applications (IR and Raman spectra, GC retention times and NMR Chemical Shift) are so similar that JCAMP-DX files for all these types can be plotted or processed (smoothed, peak-picked, deconvoluted, etc.) by existing software for one of the techniques. This can clearly be extended to many other types to XY data which are represented in the XYDATA form, and probably also as XYPOINTS.

##DATA CLASS= is intended to provide an early indication to software for the purpose of determining whether or not it can process a given file. This seems important because it allows crossover of data processing routines between different techniques.

3.1.1.2. ##NUMDIM= (AFFN).

This LDR contains the number of dimensions stored in the CORE DATA. It refers to the number of independent variables in the NTUPLES structure. It should be noted that for complex data such as NMR free induced decay curves where both the real and imaginary data sets are stored this would still count as one as they are both in the same dimension.

3.1.2. CORE DATA

Consists of one or more CORE DATASETS and corresponding sets or CORE DATA ATTRIBUTES to define the dataset and its relationship to the measurement. The data is either stored in linked independent BLOCKS or in NTUPLES PAGES of data. See Section 6 and 7 for details.

3.2. NOTES.

The NOTES subset complement the CORE by providing information on the sample, equipment, method or observation, data processing, etc. Experience has shown that the needs of users span the full range from:

those who just want data in an ASCII file for input to their own software, to

those who want to be able to describe every experimental detail in the same file as the data. By and large, the NOTES consist of information describing a sample, method of preparation and observation, on instrumentation and operating conditions, and on data processing. Instrument vendors are encouraged to include as much information as is available from their native format data files.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

4. DEFINITIONS

4.1 CHARACTER SET.

The character set consists of standard printing ASCII characters (TABLE 5.1). Characters with special meanings are listed in Table 5.2. Some advantages of ASCII are outlined below.

ASCII characters can be listed without special software. Communication hardware and software is normally transparent to printable ASCII characters. Knowledge of internal hardware representation of numbers is not required in order to write software for ASCII numerical conversions.

Choice of ASCII is a conscious judgement that readability and machine independence are more important in a data exchange standard than compactness. A single, consistent ASCII representation handles both integers and floating point values. It is of variable length, allowing unlimited precision while using only as much space as necessary.

There are various ASCII devices to minimise the space required to store data, some at the expense of human readability of the data arrays, but less so than if the data were converted to a binary representation.

Using ASCII limits the number of special characters. This forces certain characters to have special meanings under special circumstances as listed in Table 5.2.

TABLE 5.1. ASCII printing characters.

```
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _  
' a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
```

TABLE 5.2. Special characters.

##	Start of data-label. Preceded by blanks or tabs only.
=	End of data-label.
\$\$	Indicates that remainder of line is a comment.
\$	First character of a private label name (##\$SHOPORDER=).
.	First character of DATATYPE-SPECIFIC-LABEL-NAME (e.g. ##.PULSESEQUENCE=).
<>	Used as quotes to allow strings to be nested.
()	Used to delimit compound data groups containing strings.
“	Can be used to delimit strings in Data-Tables for import into spread sheets via "Comma and quote" option.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

4.2. LABELLED-DATA-RECORD (LDR).

The basic element of a JCAMP-DX file is the LABELLED DATA-RECORD. An LDR consists of a flagged data-label and an associated dataset, both defined below. An LDR begins with a Data-Label-Flag (##) and ends with the next data-label-flag.

4.3. LINES.

LDRs are divided into lines of 80 or fewer characters, terminated by an end-of-line character sequence. Each LDR occupies as many lines as necessary.

4.3.1. END-OF-LINE.

End-of-line is equivalent to a blank. Experience shows that end-of-line may be any of the following: <CR>, <LF>, <CRLF>, depending on the system which generated the file. Prudent programming practice calls for reading JCAMP-DX files in such a way that all of these are detected.

4.4. DATA-LABELS.

A DATA-LABEL contains the name of a LABELLED-DATA-RECORD. Each DATA-LABEL consists of a DATA-LABEL-NAME delimited by a data-label-flag (##) and a data-label-terminator (=). ##TITLE= is a DATA-LABEL; TITLE is the corresponding DATA-LABEL-NAME. A LINE contains no more than one DATA-LABEL. When DATA-LABELS are parsed, alphabetic characters are converted to upper case, and all spaces, dashes, slashes, and underlines are discarded. (XUNITS, xunits XUNITS, and X_UNI-TS are equivalent, thus these characters can be used optionally as separators for readability. It is important when writing software to read JCAMP-DX to realise this!

4.4.1 GLOBAL-DATA-LABELS.

GLOBAL-DATA-LABELS are RESERVED in the sense that they are defined in this standard and are available for all datatype specific standards. They are listed in the GLOBAL-LABEL-DICTIONARY (Section 8). GLOBAL-DATA-LABELS consist of alphabetic and numeric characters only.

The first character is always alphabetic.

In order for JCAMP-DX to be truly generic GLOBAL-DATA-LABELS should be independent of the type of data stored. Most of the labels which were defined in JCAMP-DX v4.24 are independent of technique and will be retained in the global label dictionary.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 8/40

4.4.2. DATATYPE-SPECIFIC-LABELS.

DATATYPE-SPECIFIC-LABELS are RESERVED labels which are defined by qualified user groups for a particular DATATYPE. A DATATYPE-SPECIFIC-LABEL is distinguished by a DATATYPE-SPECIFIC-LABEL-NAME which starts with a period e.g. (##.ATR_PLATE_MATERIAL). Choice of period as distinguishing character is by analogy with the convention for data-structure names in Pascal and C. Effectively, the full LABEL-NAME is the concatenation of the DATATYPE NAME and the LABEL NAME, with a period in between, i.e., IR_SPECTRUM.ATR_PLATE_MATERIAL.

4.4.3. PRIVATE-DATA-LABELS.

PRIVATE DATA LABELS (also known as USER-DEFINED-LABELS) are local to a particular instrument, location, or use. They can be generated by users for such fields operator's name, shoporder, company dependent variables like laboratory information management system data, etc., as needed for private use. The first character of a PRIVATE-DATA-LABEL-NAME is \$ to distinguish it from other LABEL-NAMES. e.g. ##\$LAB_NUMBER=

4.5. DATASETS.

The actual data portion of a LABELLED-DATA-RECORD is called a DATASET. Usually a DATASET starts on the same line as the DATA-LABEL, and in many cases is completed on the same line, i.e., a physical property of a sample and associated units. Other DATASETS might consist of a description of a sample, a procedure for measuring a property of a sample, a list of literature citations, a table of thousands of data points in a complex spectrum, or even spectra of fractions from a chromatography run.

Dataset structures are of varying dimensionality: Scalars, Vectors, Pages, (Slabs, Hyperslabs, etc.). We will only be dealing with Scalar, Vector, and Page datasets below. A Scalar dataset (FIRSTX, LASTX, XUNITS, etc.) is the equivalent of a point; a Vector dataset i.e. a linear array (VAR_NAME, SYMBOL, FIRST, LAST, see Section 9.2); a PAGE dataset is an array of arrays (See Section 9.3.3); etc., where the arrays can be inhomogeneous i.e., with string and numeric elements intermixed in some cases. Scalar components of datasets are described below.

4.5.1. TEXT

TEXT datasets contain descriptive information for humans, not normally intended to be parsed by computer, i.e. title, sample description, etc. normally, text datasets are scalars.

4.5.2. STRING

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

STRING datasets contain alphanumeric fields intended to be parsed by computer and read by a human. The form of each string field is specified for each use. A number of STRING datasets are defined for individual LDRs, Like LongDate, String (##LONG_DATE=), CasNameString (##CASNAME=), and many more (See Section **).

4.5.3. AFFN (ASCII FREE FORMAT NUMERIC)

AFFN is similar to free-form input or BASIC. A field which starts with a +, -, decimal point, or digit is treated as a numeric field and converted to the internal form of the target computer. E and e are the only allowed alphabetic characters. A numeric field is terminated by E, e, comma, or blank. If E is followed immediately by either + or - and a two or more digit number, it specifies the power of 10 by which the initial field (mantissa) must be multiplied.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 10/40

5. ASDF (ASCII SQUEEZED DIFFERENCE FORM).

5.1. Tabular Data

ASDF is a special compressed tabular form which makes use of the following features:

convert decimals to integers via scale FACTORS,

replacing leading digits by pseudo-digits which represent a delimiter, sign and the first digit itself (see TABLE 5.5.1.4), expressing data as differences between adjacent points, and

combining duplicate adjacent ordinates. Subsequent digit or a multidigit number are standard ASCII digits. It is not necessary to specify ASDF forms separately because the first character of each numerical value yields this information.

5.2. Data Compression

Compression and decompression via ASDF is faster than for AFFN because less characters are processed. It is also faster than most other forms of data compression. A single software procedure can be written in most high level languages to decode any and all combinations of the ASDF forms described below, together with AFFN. See Tables VI and VII for examples and translation table for ASDF.

1.	ASCII digits	0	1	2	3	4	5	6	7	8	9
2.	Positive SOZ Digits	@	A	B	C	D	E	F	G	H	I
3.	Negative SOZ Digits		a	b	c	d	e	f	g	h	i
4.	Positive DIF Digits	%	J	K	L	M	N	O	P	Q	R
5.	Negative DIF Digits		j	k	l	m	n	o	p	q	r
6.	DUP Digits		S	T	U	V	W	X	Y	Z	s

5.3. AFFN.

AFFN is actually an ASDF form. ASDF processors should always accept AFFN as well as ASDF so that it is not necessary to flag a data table to indicate that it has been compressed.

e.g. 1000 2000 2001 2002 2003 2003 2003

5.4 PACKED Form (PAC)

Adjacent AFFN values are separated by +, -' or blank.

e.g. +1000+2000+2001+2002+2003+2003+2003

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 11/40

5.5 SQUEEZED Form (SQZ)

Delimiter, leading digit and sign are replaced by a pseudo-digit from Table 5, lines 2,3.

e.g. A000B000B001B002B003B003B003

5.6 DIFFERENCE Form (DIF)

Delimiter, leading digit, and sign or the difference between adjacent values are represented by a pseudo-digit from Table 5, lines 4,5.

e.g. A000J000JJJ%%

5.7. Check-Points.

The start of each line after the first is a CHECK-POINT. Checks are of two types; X-sequence checks, and Y-value checks.

5.7.1. X-Sequence Check

At the beginning of each line, X-sequence is checked in all modes to make sure that a whole line is not omitted or duplicated.

5.7.2 Y-Value Check

In DIF mode, a Y-value check is performed at the beginning of each line to guard against a single error which would invalidate all following values. When, and only when, the last ordinate of a line is in DIF form, the abscissa does not advance before the next line.

The first ordinate of the next line (after the leading abscissa value) is always an actual value, equal to the last calculated ordinate of the previous line.

The last line of a block of DIF data contains only abscissa and ordinate for the Y-value check of the last ordinate.

5.8. DUPLICATE SUPPRESSION (DUP)

This is a special form which is valid only when adjacent values represent the same variable. In this case, two or more adjacent values are identical, all but the first are replaced by a duplicate-count whose initial digit is a pseudo-digit from Table 5, line 6. Duplicate count is the number of successive identical table values, including the first. It can be used with all ASDF forms. Count for four identical values is 4 i.e., 50 50 50 50 becomes 50V in DUP form.

5.9. DIFDUP form

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 12/40

When duplicate suppression is combined with difference form, the duplicate count is obtained by counting identical differences. The above example, 50 50 50 50, becomes 50%% in DIF form, and 50%U in DIFDUP form.

e.g. A000J000UJ%%

5.10. Multiple Dependent Variables

The JCAMP-DX v4.24 protocol did not define how DIF mode was to be applied when a data table contains multiple ordinate variables such as the real and imaginary components of a complex variable. In this case, the real components are stored independently from the imaginary in separate PAGES. See example ???

5.11. DATA TABLES.

Tabular data are the real substance of JCAMP-DX data files. They are discussed in detail in a separate Sections 6 and 7.

5.12. BLOCKS.

A JCAMP-DX BLOCK is a collection of LDRs, starting with ##TITLE= and ending with ##END=. There are two distinct types of BLOCKS; LINK BLOCKS and DATA BLOCKS which may form a multi-block structure which resembles a hierarchical directory structure, with LINK BLOCKS functioning more or less like directories, and DATA BLOCKS as files within the LINK BLOCK.

5.12.1. DATA BLOCKS

DATA BLOCKS contain only data; they may not contain other BLOCKS. It should be possible to extract each DATA BLOCK using appropriate software to a separate file.

5.12.2. LINK BLOCKS

LINK BLOCKS may contain either type of BLOCK.

5.13. FILES.

JCAMP-DX FILES may be either simple or complex. No matter whether simple or complex, a FILE starts with ##TITLE= and ends with the matching ##END=. The file name extension or a JCAMP-DX FILE is .DX. For file systems which do not use file name extensions, a file name should end with .DX or DX is feasible. The extension .JDX has also found widespread use and is also listed as a file extension for JCAMP-DX files in the Chemical Multipurpose Mail Extension tables (Chemical MIME) see ref ???(JCICS?).

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 13/40

A FILE which consists of a single DATA BLOCK is a simple JCAMP-DX FILE. One which contains multiple BLOCKS is compound JCAMP-DX FILE.

There are various reasons for using compound JCAMP-DX files containing multiple BLOCKS, but the main reason is that it is convenient to store related data in the same file. One current multiblock application combines a DATA BLOCK containing spectroscopic data with DATA BLOCKS containing chemical structural information in JCAMP-CS form within a LINK BLOCK. (JCAMP-CS is an adaptation of SMD to JCAMP-DX style). Another multiblock application involves combining NIR files for a set of standard reference materials for a given application into a single file. Still another example of a multiblock file for GCIR data was given in TABLE XI of JCAMP-DX v4.24. This application would now use the NTUPLE data structure described below which allows a complete GCIR or GCMS run in a single BLOCK, including both TIC and spectra.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 14/40

6. EXISTING TABULAR DATA FORMS

6.1 DATA FORMS ???

The JCAMP-DX v4.24 protocol described five forms of tabular data. The description was oriented mainly toward the bivariate forms. In particular, the parameter labels were provided for X and Y for all of the five forms that were described: ##XYDATA=, ##XYPOINTS=, ##PEAKTABLE=, ##ASSIGNMENTS=, and ##RADATA=. The latter is 4_tuple form for interferograms which require two variables for the interferogram which is measured experimentally and two more for the spectrum which is obtained by transforming the interferogram.

6.2 THE NEED FOR A NEW DATA FORM ???

Since JCAMP-DX v4.24 the need has arisen for forms for more complex data, and a general n-tuple form has been worked out. It has the advantage that it does not require a new set of labels nor units, factors, first and last values etc., for each new variable. This N-TUPLE form is described in Section 7. It is so general that it could be used for all of the five forms which were described previously. However, some of the previous forms are in wide use, and it does not seem workable to discard them in favor of the more general form. Hence, those for which a full set of parameters was provided previously will be retained. These forms are ##XYDATA=, ###XYPOINTS=, ##RADATA=, and the form of ##PEAKTABLE= which is limited to peak position and peak intensity. The form of ##PEAKTABLE= which includes peak width and ##ASSIGNMENTS= will be dropped (unless there is a strong outcry). It is recommended that the ##RADATA= be converted to the new multivariate form, especially because it will accommodate both sample and reference interferograms in the same BLOCK or FILE.

6.3 ???

The new section which describes ##XYDATA=, ##XYPOINTS=, i.e., ##PEAKTABLE=, ##ASSIGNMENTS=, and ##RADATA= has yet to be written. For the present, refer to V4.24 Sections 6.2-6.4, and 8.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 15/40

7. MULTIVARIATE DATA TABLES - NTUPLES

The basic idea for adapting the NTUPLES data structure for use in JCAMP-DX was suggested by James Hegedus of VG Instruments.

The NTUPLES data structure described below is a computer and human readable form, designed to represent the full range of data from bivariate (X,Y) to multivariate forms consisting of any number of independent or dependent variables. It is sufficiently general for most common types of tabular numerical data and related string data.

The following discussion of the NTUPLES form is expressed in terms associated with data as output by laboratory instruments, but this form will accommodate many other types of tabular data, even those which appear in handbooks. The NTUPLES data structure is intended to be simple when the data itself can be represented in a simple table. Complexity creeps in when the structure of the data itself is complex.

The NTUPLES structure represents a significant extension to JCAMP-DX v4.24 which was focused primarily on bivariate data.

It can accommodate all of the data forms described in V4.24: ##XYDATA=, ##XYPOINTS=, ##PEAKTABLE=, ##ASSIGNMENTS=, and ##RADATA=. However, these previously defined forms will be retained for data compatibility.

All new applications should use the ##NTUPLES= form.

7.1 NTUPLES DATA STRUCTURE

The NTUPLES data structure consists of the NTUPLES label, an ATTRIBUTE TABLE, which specifies VARIABLES and ATTRIBUTES, and one or more DATA PAGES (also called simply PAGES, each of which contains a DATA TABLE and ATTRIBUTES which are local to the PAGE itself. ATTRIBUTES are such items as variable names, units, scale factors, type of representation (numeric or string), first and last values, maximum, minimum, etc.

7.1.1. NTUPLES ATTRIBUTE TABLE

The NTUPLES ATTRIBUTE TABLE is a two dimensional table with ROWs representing NTUPLES ATTRIBUTES, and COLUMNs representing different VARIABLES. Each ROW in the table starts with a JCAMP-DX LABEL which names the class of the ATTRIBUTES listed in that ROW i.e. ##UNITS= or ##VAR_NAME=. The predefined ATTRIBUTES will tend to be generic and can be used for a wide variety of applications. This reduces the number of Labels which must be coded into software, or entered into the global LABEL dictionary.

7.1.2. NTUPLES DATA PAGE

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 16/40

A DATA PAGE consists of a PAGE LABEL (##PAGE=), a PAGE ATTRIBUTE LIST for ATTRIBUTES local to the PAGE (T1=0.1144165E-03 for example defining the second dimension in an 2D-NMR), and a DATA TABLE.

Previously defined GLOBAL and NTUPLES ATTRIBUTES are temporarily overridden by PAGE ATTRIBUTES. This can be used for example to redefine ##FIRST= and ##LAST= values that are only relevant to that particular PAGE.

The contents of the DATA TABLE itself are organized into LINES whose structure is defined by a VARIABLE LIST (See Section 7.3.3.1) followed by the style of plot (PROFILE, XYDATA, PEAKS, CONTOUR) when appropriate.

7.1.3. NTUPLE MULTIPLE PAGES

The NTUPLES structure may contain multiple PAGES for data with two or more INDEPENDENT VARIABLES, for techniques like GCIR, GCMS, 2D-NMR, nD-NMR, 2D fluorescence data, etc. To write multidimensional data files using JCAMP-DX v4.24 would require a separate BLOCK including rewriting the entire header information each time to store the data found in each PAGE using NTUPLES.

7.1.4. MULTI NTUPLE STRUCTURES

Multiple NTUPLES structures are allowed in a JCAMP-DX BLOCK or FILE for the purpose of accommodating heterogeneous DATA TABLES, such as the combination of total ion current with corresponding GCIR or GCMS scans. In other applications, calibration data can be tabulated in one NTUPLES structure, while raw data is tabulated in a second NTUPLES structure in the same BLOCK or FILE.

7.1.5. ADAPTING NTUPLES FOR INDIVIDUAL DATATYPES

The NTUPLES structure is flexible allowing expansion where necessary for individual DATATYPE SPECIFIC applications. One example is the use of the NTUPLES structure to store NMR chemical shift referencing data for each of the independent variables (see example ???)

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 17/40

TABLE 7.1 BNF Description of NTUPLES LDR

Conventions:

- LDR is an abbreviation for Labelled data Record
- () alternate terms
- :
- separates alternate terms
- [] optional terms
- { } terms which may occur zero or more times
- "" terminal symbols (symbols which appear literally in LDRs).
i.e. ##UNITS=.
- <> application specific attribute name

TABLE 7.1.1. NTUPLES GROSS STRUCTURE:

NtuplesStructure=	NtuplesLdr+NtuplesAttributeTable+PageDataTable+EncNtuplesLdr
NtuplesAttributeTable=	VarNamLdr + [SymbolLdr] + [VarTypeLdr]+[VarFormLdr] + [VarDimLdr] + [UnitsLdr] + [FirstLdr] + [LastLdr] + [MinLdr] + [MaxLdr] + [FactorLdr] + { ApplicationAttributeLdr(s)}
PageDataTable=	PageLdr + [PageVariableLdr(s)] + DataTableLdr

TABLE 7.1.2. NTUPLES LDRs:

NtuplesLdr=	##NTUPLES=" + <application name>
VarNameLdr=	##VAR_NAME=" + {<variable name>}
SymbolLdr=	##SYMBOL=" + {<symbol>} <i>can this appear zero times?</i>
VarTypLdr =	##VAR_TYPE="+{"ABSCISSA"} "ORDINATE." "INDEPENDENT" "DEPENDENT" "PAGE" "CALCULATED" dummy)}
VarFormLdr=	##VAR_FORM=" + {"AFFN" "ASDF" <string> dummy}
VarDimLdr=	##VAR_DIM=" + {"AFFN" dummy}
UnitsLdr=	##UNITS=" + {<units> dummy}
FirstLdr=	##FIRST=" + {<AFFN> <STRING> dummy}
LastLdr=	##LAST=" + {<AFFN> <STRING> dummy}
MinLdr=	##MIN=" + {<AFFN> dummy}
MaxLdr=	##MAX=" + {<AFFN> dummy}
FactorLdr=	##FACTOR=" + {<AFFN> dummy}
PageLdr=	##PAGE=" + {Symvar(s)} <i>why Symvars?</i>
SymVar=	<Symbol> "=" <Value*> * Value can be numeric or string
PageVariableLdr=	##" + <PageVariableLabelName> + "=" + {<AFFN> <STRING> <TEXT>}
DataTableLdr=	##DATA TABLE=" + <Variable List>+ ("PROFILE" "XYDATA" "PEAKS" "CONTOUR" <empty>) <i>is this complete</i>
EndNtuplesLdr=	##END NTUPLES"+[<Application Name>]

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

TABLE 7.2 NTUPLES SCHEMATIC

See TABLE 7.1 for Conventions.

Datatype and Application specific LDRs which are not associated with specific variable appear before the NTUPLES label if they apply to all PAGES. For example, the NMR attribute **##OBSERVED NUCLEUS=** would appear before **##NTUPLES=???**(*possible wrong example if ##NUCLEUS= is moved into the table!*). In a different example, **##CASNAME=** would appear before the NTUPLES label only if all Data Tables apply to the same pure compound. See comment under **##PAGE=** on use of **##CASNAME=** when each PAGE contains data of a different compound.

##NTUPLES=	<Data Name>	
##VAR_NAME=	<NAME ₁ >, , <NAME _n >
##SYMBOL=	<SYMBOL ₁ >, , <SYMBOL _n >
##VAR_TYPE=	<TYPE ₁ >, , <TYPE _n >
##VAR_FORM=	<FORM ₁ >, , <FORM _n >
##VAR_DIM=	<DIM ₁ >, , <DIM _n >
##UNITS=	<UNITS ₁ >, , <UNITS _n >
##FIRST=	<FIRST ₁ >, , <FIRST _n >
##LAST=	<LAST ₁ >, , <LAST _n >
##MIN=	<MIN ₁ >, , <MIN _n >
##MAY=	<MAX ₁ >, , <MAX _n >
##FACTOR=	<FACTOR ₁ >, , <FACTOR _n >

Datatype- and application-specific header attributes which apply only to 1 or more PAGE VARIABLES then follow. These attributes are valid for all subsequent PAGES until **##END NTUPLES=** unless temporarily overridden by a PAGE ATTRIBUTE.

##PAGE= {SymVal}

Each SymVal is of the form: SYMBOL = VALUE.

One SymVal is required for each INDEPENDENT variable in the current PAGE. i.e., GCMS scan ID, or co-ordinates of a point under study, or any other variable which defines the current PAGE.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 19/40

##NPOINTS= <No of points>

Optional if same as ##VAR_DIM=

##NPOINTS= is a typical example of a variable which may be different for each page which represents a peak table *–(I think we should stick to ##VAR_DIM= here for NTUPLES and not suddenly bring in NPOINTS!!!!)*

##DATA TABLE= <VARIABLE LIST>, <PLOT TYPE>

Data table sequence is defined by Variable List in previous line, using symbols defined by ##SYMBOL= and data forms defined by ##VARFORM=.

-
- (DATA)
-

7.2 NTUPLES ATTRIBUTES.

The following section describes the general form of each NTUPLES ATTRIBUTE. Application specific details, such as optional or required, limits, defaults, etc., are to be specified in individual protocols for each DATATYPE SPECIFIC applications such as nD-NMR.

In general all other ATTRIBUTE LDRs are optional. They may be specified as "required" for specific applications. ##NTUPLES= itself is required only when the table has a special name.???

7.2.1. GENERAL CONVENTIONS FOR ATTRIBUTE TABLES:

7.2.1.1. ROWS

Each ROW represents a different ATTRIBUTE CLASS and gives values of the ATTRIBUTE under each VARIABLE when appropriate.

7.2.1.3. LIMIT ON NUMBER OF VARIABLES

There is no formal limit to the number of VARIABLES or ATTRIBUTE CLASSES.

7.2.1.4. DUMMY VARIABLES

Each row must include an actual value or a dummy entry for each VARIABLE.

For some ATTRIBUTES, it does not make sense to provide an entry for each column. In that case, an extra comma i.e. entered to show a dummy entry (indicated by 'dummy' in Table 7.1).

For example, a scale FACTOR need not be provided for a string variable which names chemical fragments corresponding to peaks in a mass spectrum peak table. Dummy entries are for spacing purposes.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

7.2.1.5 CONTINUATION OVER LINE BREAKS

As with any LDR in JCAMP-DX a ROW may be continued on a new line after any comma and is only terminated by the next data label flag (##). This may be especially advantageous for ##VAR_NAME= because it allows as much as one line for description of each VARIABLE. In most cases this is adequate to label a plot axis.

7.2.1.6 NUMERICAL VALUES

Numeric entries are in ASCII free format numeric (AFFN) form which allows ALL print forms of integers and real numbers.

7.2.1.7 STRING VALUES

String entries may be surrounded by double quotes (") to allow NTUPLES Attribute tables to be imported into spread sheets via the "Comma and Quote Delimited" option. Most spreadsheets ignore unquoted littorals like JCAMP-DX labels which will therefore not appear in a spread sheet. This allows attribute columns to be aligned with the corresponding data columns. Spread sheets do not handle continuation lines, so that the number of variables for spread sheet applications is limited because of the 80 character line length limit for JCAMP-DX tiles. (*to be deleted???*).

7.2.2 ##NTUPLES=

##NTUPLES= (NAME STRING). Is always the first label in an NTUPLES data structure. The name is that of the data form which the structure represents (TIC, IR SPECTRUM INTERFEROGRAM, FID, NMR SPECTRUM, nD NMR FID, MS PEAK TABLE, etc.).

e.g. ##NTUPLES= nD NMR FID

7.2.3 ##VAR_NAME=

##VAR_NAME= This REQUIRED LDR contains common names of all VARIABLEs for the current dataset. This LDR always appears before the ATTRIBUTES described below because it provides column headings for them. Variable names must be kept short in order for them to fit on a single line of no more than 80 characters. Otherwise, entries can be as long as one line each, enough to provide adequate labels for plot axes. The order of listing of VARIABLES is as follows:

7.2.3.1 ABSCISSA.

The independent variables which serves as ABSCISSA for each PAGE DATA TABLE are always listed in the first columns of ##VAR_NAME=, even in the case in which a constantly incremented abscissa is not explicitly given in the DATA Table.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 21/40

7.2.3.2 ORDINATE.

The ABSCISSA is followed by a listing of ORDINATE (dependent) VARIABLES. Multiple ORDINATE VARIABLES are allowed (real/imaginary data pairs, output of multiple detectors, etc.)

7.2.3.3 INDEPENDENT VARIABLES IDENTIFYING PAGES IN A STACK PLOT.

Independent variables which identify PAGES, called PAGE COORDINATES, are listed in columns to the right of those for DEPENDENT VARIABLES. Examples of PAGE COORDINATES are GCIR, GCMS fraction numbers, spatial co-ordinates of a sampling point for a spectrum, dates or times of measurements, etc.

7.2.3.4 PROCESSED DATA VARIABLES.

When a Data Table contains arrays of raw data, it may be desirable to include header information on VARIABLES which apply to processed data which is not included in Data Tables. For example, when the tabulated data consists of an interferogram or FID it may be desirable to include the spectral variables in the Attribute Table. ATTRIBUTES for such processed variables are appended in columns to the right of ATTRIBUTES for all measured variables. **FIRST and LAST spectral VARIABLES for an infrared interferogram can be used to indicate the beginning and end of valid data, as influenced by detector sensitivity, filters, etc. (what! to be deleted???)**.

e.g. ##VAR_NAME= TIME1, TIME2, FID/REAL, FID/IMAG

7.2.4 ##SYMBOL= (STRING).

This LDR provides a symbol for each variable, except as indicated in 7.2.3.1-2 below. These symbols are used in the VARIABLE LIST which specifies the organization of values in DATA TABLES.

e.g. ##SYMBOL= T1, T2, R, I

7.2.4.1 ABSCISSA, UNIFORMLY INCREMENTED.

The symbol ++ is appended directly after the ABSCISSA symbol (X++) in the ##SYMBOL= entry, (no we dont!) and also in the VARIABLE LIST for the Data Table to indicate that the abscissa is incremented by $(##LAST= - ##FIRST=)/(##VAR_DIM= - 1)$ between each dependent variable group. Thus, (X++(Y..Y)) indicates that the first value in a line is the abscissa corresponding to the first Y value, and that X is incremented by a constant amount to obtain the abscissa for each subsequent Y value in the line.

Certain conventions (X-sequence and Y-value checks, Section 5.7) hold for repeating abscissa/ordinate pair at the beginning of a line which follows a line ending with a DIF value.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

7.2.4.2 OMISSION of ABSCISSA from DATA TABLE.

Uniformly incremented ABSCISSA values can be omitted from a Data Table when adequate information to calculate each ABSCISSA is given in the form of FIRST, LAST, and number of points. If so, the ABSCISSA symbol and the two + signs are enclosed in curly brackets, i.e. (X++), in the ##SYMBOL= row in the ATTRIBUTE LIST, and also in the VARIABLE LIST for the Data Table. Certain conventions and retractions hold when ordinate data is presented in DIF form (See Section 'em). (*to be deleted???*).

7.2.5 ##VAR TYPE= (STRING).

This OPTIONAL LDR will indicate the variable type for each variable as "ABSCISSA", "ORDINATE", PAGE", "INDEPENDENT", "DEPENDENT", or "CALCULATED". VARITYPE is always required for data which requires multiple PAGES. It is optional if data requires only a single PAGE. PAGE is used to identify a VARIABLE which is used to identify a PAGE.

e.g. ##VAR_TYPE= INDEPENDENT, INDEPENDENT, DEPENDENT,
DEPENDENT

7.2.6 ##VAR FORM= (STRING).

This LDR gives the form of the data values for each variable as "AFFN", "ASDF", or "STRING". In addition to the generic "STRING", various predefined strings, i.e., DateString, TimeString, CasNameString, etc., are allowed. "TEXT" is not usually applicable.

e.g. ##VAR_FORM= AFFN, ASDF, ASDF, ASDF

7.2.7 ##VAR DIM= (AFFN).

This LDR will indicate the array DIMENSION for each variable for use in allocating memory. VAR_DIM is REQUIRED for multiple PAGES to indicate the dimensions of PAGE COORDINATES to inform users and software concerning the number and organization of PAGES. When the size of each PAGE may be different, i.e., for GCMS scans, the dimension is given via ##NPOINTS= in each PAGE HEADER. In that case, ##VAR_DIM gives the maximum array size for any PAGE in the current NTUPLES structure. ##VAR_DIM= entries are omitted for *calculated* variables because number of elements is not defined until the calculation is performed.

e.g. ##VAR_DIM= 64, 1024, 1024, 1024

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 23/40

7.2.8 ##UNITS= (STRING).

This LDR gives the customary units for each numeric variable.

e.g. ##UNITS= SECONDS, SECONDS, ARBITRARY UNITS,
ARBITRARY UNITS

7.2.9 ##FIRST= (AFFN).

This LDR gives the first value of each numeric variable. It is REQUIRED to verify scaling when tabulated DATA is scaled via ##FACTOR=.

e.g. ##FIRST= .1144165E-03, .0000000E+00, ,

7.2.10 ##LAST= (AFFN).

This LDR gives the last value of each numeric variable. This LDR may be required for use as a data validity check in certain cases when dependent variables are compressed via DIF mode.

e.g. ##LAST= .7322654E-02, .4255673E-01, ,

7.2.11 ##MIN= (AFFN).

This OPTIONAL LDR gives the minimum value of each numeric variable for all PAGES.

e.g. ##MIN= .1144165E-03, .0000000E+00, -669731, -694236

7.2.12 ##MAX= (AFFN).

This OPTIONAL LDR gives the maximum value of each numeric variable for all PAGES. For a string variable ##MAX= gives the maximum length of the string.

e.g. ##MAX= .7322654E-02, .4255673E-01, 773095, 691948

7.2.13 ##FACTOR= (AFFN).

This OPTIONAL LDR gives the factor by which each numeric variable is to be scaled. It is required if any numeric variable has a scale factor different from 1.0, i.e., if ##FACTOR= is missing all scale factors are unity. ##FACTOR= is placed in the last row in order to emphasize that it is applied only to the values tabulated in Data Tables which follow. It is not applicable to string variables.

e.g. ##FACTOR= .1000000E+01, .4159993E-04, .2359371E+02, .2118705E+02

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

7.2.14 APPLICATION ATTRIBUTE LDRs (AFFN or STRING).

Additional application-specific LDRs are used where appropriate (Examples: ##.SCAN LAW= or ##.INLET= for GCMS). Attributes may be Scalar or Vector type, as appropriate. Vector attributes assume same ordering of variables as given by Var_Name, which they must follow. Scalar attributes may be placed either before or after ##NTUPLES=, depending on their intended scope.

7.2.15 ##END NTUPLES= (application name). (do we need the application name here?)

This optional LDR is used to terminate an NTUPLES Data Structure if necessary to terminate the scope of all variables defined within the NTUPLES structure.

7.3 NTUPLE DATA TABLES.

Any number of Data Tables can follow the Ntuples Global Attribute Table. For example, all of the mass spectra of a GCMS run can be tabulated in the order in which they were measured, or the spectra of many thickness of a given substance. This provides a very wide range of possibilities, not limited to spectra. Each data table contains a header which starts with ##PAGE=. The PAGE HEADER may contain LDRs which give values which apply only to the current PAGE. The combination of the PAGE HEADER and the Data Table is called a PAGE. All of the attributes in the NTUPLES ATTRIBUTE TABLE apply to the PAGE except where overridden by a corresponding LDR in the PAGE HEADER.

7.3.1 ##PAGE= (STRING).

This LDR indicates the start of a PAGE which contains tabular data. It may have no argument, or it may be omitted when the data consists of one PAGE. When the Data Table represents a property like a spectrum or a particular fraction, or at a particular time, or at a specific location in two or three dimensional space, the appropriate PAGE VARIABLE values will be given as arguments of the ##PAGE= LDR, as in the following examples:

```
##PAGE= N=1          $$ Spectrum of first fraction of GCIR run
##PAGE= T=10:21      $$ Spectrum of product stream at time: 10:21
##PAGE= X=5.2, Y=7.23  $$ Spectrum of known containing 5.2 % X and 7.23%
Y
```

The end of a PAGE is indicated by a new ##PAGE=, ##END TUPLES=, or ##END=, depending on the context.

7.3.2 PAGE VARIABLE LDRs (AFFN or STRING).

These LDRs can be any which have different values for each PAGE of a multi-PAGE dataset, for example:

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 25/40

##NPOINTS= \$\$ when each PAGE lists a different number of points
##CASNAME= identification of the component for a given peak table or any
appropriate LDR containing sample information, sample prep, computational procedure,
which differs from PAGE to PAGE.

7.3.3 ##DATA TABLE=.

This LDR contains the tabulated data for the PAGE. The label is followed by a VARIABLE LIST which defines the sequence of data items using symbols declared above in ##SYMBOLS=. The Variable List must be parsed to determine the sequence because the variables can be different from one PAGE to the next. If the VARIABLE LIST is missing, the order of the variables is assumed to be the same as for ##VAR_NAME=, and the variables in each group must be separated by commas, and groups must be separated by at least one space.

7.3.3.1 VARIABLE LIST.

The form of ##DATA= is represented symbolically as a VARIABLE LIST which can be understood from the following examples: (This section is not consistent with the other standards – I suggest that to avoid confusion it is essential to keep the same definitions as previously used and not to expand the range of possible variable lists! T.D.)

- (XY) one value each of X and Y per line of data, separator unspecified.
(X,Y) same, but separated by commas (new)
(XY .. XY) as many XY pairs as will fit on a line separator unspecified (new version?)
(XY..XY) *X and Y are separated by commas; pairs are separated by semicolons or
blanks. This LDR should not be used for peak tables (use ##PEAK TABLE=). (v.2.42)*
(X,Y .. X,Y) same, but X and Y separated by commas, XY groups separated by blanks
(X,Y ...) fill the line, X and Y separated by commas, XY groups separated by blanks,
continue on next line after X,
(X++(Y..Y)) indicates the order of values of the indicated variables in each LINE of the
current DATA TABLE. The initial X indicated that the first value in a line is the X value
corresponding to the first Y value. (Y..Y) indicates an ordered sequence of Y values to fill
the remainder of the line. The ++ indicates that X is incremented by $\Delta X = (\text{LASTX} - \text{FIRSTX}) / (\text{NPOINTS} - 1)$ between Y values (sign of ΔX determines direction. Additional
dependent variables can be included as follows: (X++(YZ..YZ)).
(X++)(Y,Z ...) The symbol (X++) indicates that X is incremented between each XY pair, but
X values are omitted from table.

7.3.3.2 PLOT DESCRIPTOR.

The VARIABLE LIST may optionally be followed by one of the following plot descriptors: PROFILE, XYDATA, PEAKS, or CONTOUR which will indicate the appropriate method for

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

plotting the data, as described below. No plot descriptor is required for Data which is to be plotted as a set of isolated points.

7.3.3.2.1 PROFILE.

PROFILE is used for fully digitized monotonic 2 dimensional curves in which adjacent points are not evenly spaced, required each abscissa value to be tabulated with its corresponding ordinate(s). for spectra, chromatograms, thermal methods, etc., which are displayed by connecting adjacent points. The data should be in sorted order based on the independent variable(s).

7.3.3.2.2 XYDATA.

XYDATA is used for fully digitized 2 dimensional curves in which adjacent points are evenly spaced. Normally, such data is (X++(Y..Y)).

7.3.3.2.3 PEAKS.

This descriptor indicates that data should be plotted as a series of peaks. The data should be in sorted order based on the abscissa. Normally, the number of peaks will differ from PAGE to PAGE in multi-PAGE tables. ~~The number of peaks in each page should be given in the page header using ##NPOINTS= to allow software which reads the table to allocate arrays of the appropriate size.~~ The number of peaks in each page should be given in the individual page headers using ##VAR_DIM= to allow software which reads the table to allocate arrays of the appropriate size. The total number of peaks for a PAGE in a multi-PAGE table should be given under ##VAR_DIM= in the ##NTUPLES= header.

7.3.3.2.4 CONTOUR.

The descriptor is used for re-entrant curves representing contours (hysteresis, electrochemical cycles, etc.). The order of the points indicates the order in which they should be connected in a plot. If the data represents disconnected curves i.e., multiple contours, a separate PAGE should be used for each curve.

7.4 EXAMPLES.

The following examples suggest ways that the NTUPLES form can be used. They are not OFFICIAL forms for the applications which are mentioned. These forms will normally be specified by appropriate user groups via Meta Files. Interested user groups or individuals are invited to suggest more appropriate examples.

7.4.1 NTUPLES EXAMPLE .. IR PEAK TABLE

Quotes which surround strings permit this table to be read into a spreadsheet like Lotus or Quattro. JCAMP-DX software which reads this table should ignore the quotes.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 27/40

```
##NTUPLES=          "IR PEAKS"  
##VARNAME=          "Frequency", "Intensity",      "FW-M"  
##SYMBOL=          "X",      "Y",      "W"  
##VARDIM=           100,      "X",      "X"  
##VARFORM=         "AFFN",    "AFFN",    "AFFN"  
##UNITS=            "1/CM",    "Absorbance",  "1/CM"  
##FIRST=            200  
##LAST=             4000  
##PAGE=  
##POINTS= 7  
##PEAKS= "(XYW)"  
300.15, 1.5, 2.0  
403. 2.5, 3.0  
500 1.1 10.0  
563.21 .15 3.2  
1020.22 1.32 20.0  
1592.11 2.34 15  
3021.1 1.34 11.2  
##END NTUPLES=
```

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 28/40

8. JCAMP-DX GLOBAL DICTIONARY

8.1 THE CONCEPT

The JCAMP-DX Global Dictionary consists of all GLOBAL LABELS and special strings, together with definitions. They have been collected from the various published JCAMP-DX protocols and are a definitive list of the information fields which may be used for generic information.

8.2 GLOBAL LABELS defined in V4.24

##TITLE=(TEXT). \$\$ JCAMP-DX v.4.24
(6.1.1)

Required as the first LDR of all JCAMP-DX FILES and BLOCKS. Software which decodes JCAMP-DX files should check for the initial ##TITLE= to prevent accidental processing of non-JCAMP-DX files. For a BLOCK containing a spectrum, ##TITLE= should contain a concise description of the spectrum, suitable as a title for a plotted spectrum.

Software which generates a simple JCAMP-DX file should copy the internal field which most closely meets this requirement.

##JCAMP-DX=(STRING). \$\$ JCAMP-DX v.4.24
(6.1.2)

Required immediately after ##TITLE= of each JCAMP-DX BLOCK to show version of JCAMP-DX; for example: ##JCAMP-DX= 4.24.

##DATA TYPE=(STRING). \$\$ JCAMP-DX v.4.24 (6.1.3)

Required to define type of data in a BLOCK (Table IIa, line 3). ##DATA TYPE= for a data BLOCK containing an infrared spectrum is INFRARED SPECTRUM. ##DATA TYPE= for all link BLOCKS is LINK. Other ##DATA TYPES= are: RAMAN SPECTRUM, INFRARED PEAK TABLE, INFRARED INTERFEROGRAM, and INFRARED TRANSFORMED SPECTRUM.

##BLOCKS=(AFFN). \$\$ JCAMP-DX v.4.24
(6.1.4)

This LDR gives the number of link and data BLOCKS in the first level of the current BLOCK of a compound JCAMP-DX file (Section 11). This LDR is required in link BLOCKS; it

is not allowed/in data BLOCKS.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

##END=, \$\$ JCAMP-DX v.4.24
(6.1.5)

End of a JCAMP-DX BLOCK. There must be one ##END= for each ##TITLE= in a JCAMP-DX file.

##XUNITS=(STRING), \$\$ JCAMP-DX v.4.24 (6.2.1)

Specifies abscissa units for ##XYDATA=

##YUNITS=(STRING), \$\$ JCAMP-DX v.4.24 (6.2.2)

Specifies ordinate units of ##XYDATA=, such as in the infrared case where I = intensity, and R = reflectance:

TRANSMITTANCE (I_T/I_0)

REFLECTANCE (I_R/I_0)

ABSORBANCE ($\log_{10} I_0/I$)

KUBELKA-MUNK (O-RW2R))

ARBITRARY UNITS

##FIRSTX=, **##LASTX=** (AFFN), \$\$ JCAMP-DX v.4.24 (6.2.3)

First and last actual abscissa values of ##XYDATA=. First tabulated abscissa times ##XFACTOR= should equal ##FIRSTX=.

Spectral data are tabulated in order of either increasing or decreasing abscissa values.

##FIRSTX= specifies abscissa of the first datum of ##XYDATA=, not necessarily the smallest abscissa.

##MAXX=,**##MINX=**,**##MAXY=**,**##MINY=** (AFFN), \$\$ JCAMP-DX v.4.24 (6.2.4)

These optional LDRs give the largest and smallest actual X- and Y-values in a spectrum.

When used, they precede ##XYDATA=. They are used for scaling when spectra are plotted directly from JCAMP-DX files and as follows:

- (1) ##MAXX= and ##MINX= are required to give the full X-range if different regions of a spectrum are contained in separate BLOCKS of a compound file, and
- (2) ##MAXY= and ##MINY= are required if Yrange exceeds recommended ordinate scaling.

##DELTAX= (AFFN), \$\$ JCAMP-DX v.4.24 (6.3.2)

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

This LDR gives the nominal spacing between points for inspection by the user. Actual X-values are calculated from the spacing given by (LASTX-FIRSTX)/(NPOINTS-1) (see Section 5.1.1).

##XFACTOR=, ##YFACTOR= (AFFN). \$\$ JCAMP-DX v.4.24 (6.2.5)

Factors by which components of ##XYDATA= are multiplied to obtain actual values. ##XFACTOR= may be used to convert X-values in ##XYDATA= to integers. Rationally spaced X-values (intervals of 0.5, 1.0, etc.) are usually left unscaled. Laser spaced X-values require several decimal places. Such X-values can be converted to integers to reduce file size by a suitably chosen ##XFACTOR= (which must be stated with sufficient precision for accurate reconstruction of X-values).

##YFACTOR= is normally chosen to convert Y-values to integers to save space required by decimal points, i.e., if ##YFACTOR= is 0.0001, transmittance of 0.9999 is tabulated as 9999.

##NPOINTS= (AFFN). \$\$ JCAMP-DX v.4.24 (6.2.6)

Number of components of ##XYDATA=, ##XYPOINTS=, ##PEAKTABLE=, or ##RADATA=.

##FIRSTY= (AFFN). \$\$ JCAMP-DX v.4.24 (6.2.6)

Actual Y-value corresponding to ###FIRSTX=. ##FIRSTY= should equal ##YFACTOR= times the first Y-value value in ##XY-DATA=. and Y represent abscissa and ordinate of spectral data.

##RESOLUTION= (STRING). \$\$ JCAMP-DX v.4.24 (6.3.1)

Nominal resolution in units specified by ##XUNITS=, as a single number for spectra at constant resolution throughout, or as pairs of the form: R,,X1; . . . ;Ri,Xi, where Ri stands for resolution at abscissa XT. ##RESOLUTION= is strongly recommended.

reflectance:

##DELTA= (AFFN). \$\$ JCAMP-DX v.4.24 (6.3.2)

This LDR gives the nominal spacing between points for inspection by the user. Actual X-values are calculated from the spacing given by

(LASTX-FIRSTX)/(NPOINTS-1)

TABULAR SPECTRAL DATA.

##XYDATA= (AFFN or ASDF). \$\$ JCAMP-DX v.4.24 (6.4.1)

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

Contains a table of ordinates at equal X-intervals specified by parameters defined in Section 6.2. This LDR will extend over many lines. ##XYDATA= is followed on the same line by a variable list, i.e., (X++(Y..Y)), representing the form of the table as described in Section 5.1. (Table IIa, line 33.)

Under no circumstances should ##XYDATA= start with a region of undefined data. Software should adjust ##FIRSTX= so that the first point is valid data.

##CLASS= (STRING). \$\$ JCAMP-DX v.4.24 (7.1.1)

Specifies the Coblenz Class of the spectrum (1,2,3, or 4) and the IUPAC Class of digital representation (A, B, C).³ This LDR is required for spectra of pure compounds.

##ORIGIN= (TEXT). \$\$ JCAMP-DX v.4.24 (7.1.2)

Name of organization, address, telephone number, name of individual contributor, etc., as appropriate. This information is not optional; the originator of a JCAMP-DX file should always see that it is included.

##OWNER= (TEXT). \$\$ JCAMP-DX v.4.24 (7.1.3)

Name of owner of a proprietary spectrum. The organization or person named under ##ORIGIN= is responsible for accuracy of this field. If data are copyrighted, this line should read "COPYRIGHT (C) (year) by (name)." This information is not optional; the originator of a JCAMP-DX file should always see that it is included. If ##OWNER= contains PUBLIC DOMAIN, the implication is that the data may be copied without permission on the authority of whoever is named under ##ORIGIN=. Absence of ##OWNER= does not imply permission to copy.

##DATE= (STRING). \$\$ JCAMP-DX v.4.24 (7.1.4)

Date when spectrum was measured, in the form: YY/MM/DD (note order).

And

##TIME= (STRING). \$\$ JCAMP-DX v.4.24 (7.1.5)

Time when spectrum was measured, in the form: HH:MM:SS.

Both ##DATE= and ##TIME= Are now superceded by:

##LONGDATE=(STRING) \$\$ JCAMP-DX v.5.01(3.3)

The long date format is ##LONGDATE= YYYY/MM/DD [HH:MM:SS[.SSSS] [±UUUU]]
where

- YYYY is the year

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 32/40

- MM is the month
- DD is the day
- HH is the hour (optional)
- MM are the minutes (optional)
- SS.SSSS are the seconds and fractional seconds (optional)
- UUUU - is time zone difference to Greenwich Mean Time (GMT) (optional)

For example: ##LONGDATE=1998/08/12 23:18:02.0000 -0500
 ##LONGDATE=1988/08/12

##SOURCE REFERENCE= (TEXT). \$\$ JCAMP-DX v.4.24 (7.1.6)

Adequate identification to locate the original spectrum, i.e., name of file containing the spectrum, or library name and serial number of the spectrum.

##CROSS REFERENCE= (TEXT). \$\$ JCAMP-DX v.4.24 (7.1.7)

Cross references refer to additional spectra of the same sample, i.e., different thickness, mulling agent, polarization, temperature, time, etc., or serve to link a peak table or interferogram with a spectrum.

##SAMPLE DESCRIPTION= (TEXT). \$\$ JCAMP-DX v.4.24 (7.2.1)

If the sample is not a pure compound, this field should contain its description, i.e., composition, origin, appearance, results of interpretation, etc. If the sample is a known compound, the following LDRs specify structure and properties, as appropriate.

##CAS NAME= (STRING). \$\$ JCAMP-DX v.4.24 (7.2.2)

Name according to Chemical Abstracts naming conventions as described in the 1985 CAS Index Guide. Examples can be found in Chemical Abstracts indices or the Merck Index. Greek letters are spelled out, and standard ASCII capitals are used for small capitals. Sub-/superscripts are indicated by prefixes / and ^, respectively.

e.g. alpha-D-glucofuranose, 1-(dihydrogen phosphate).

##NAMES= (STRING). \$\$ JCAMP-DX v.4.24 (7.2.3)

Common, trade, or other names. Multiple names are placed on separate lines.

##MOLFORM= (STRING). \$\$ JCAMP-DX v.4.24 (7.2.4)

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 33/40

Molecular formula. Elemental symbols are arranged with carbon first, followed by hydrogen, and then remaining element symbols in alphabetic order. The first letter of each elemental symbol is capitalized. The second letter, if required, is lower case. One-letter symbols must be separated from the next symbol by a blank or digit. Sub-/superscripts are indicated by the prefixes / and ^, respectively. Sub- /superscripts are terminated by the next nondigit. Slash

may be omitted for subscripts. For readability, each atomic symbol may be separated from its predecessor by a space. For substances which are represented by dot-disconnected formulas (hydrates, etc.), each fragment is represented in the above order, and the dot is represented by *. Isotopic mass is specified by a leading superscript. D and T may be used for deuterium and tritium, respectively.

e.g. C2H4O2 or C2 H4 O2 (acetic acid)
C6 H9 Cr O6 * H2 O (chromic acetate monohydrate)
H2 ^17O (water, mass 17 oxygen)

##CAS REGISTRY NO= (STRING). *\$\$ JCAMP-DX v.4.24 (7.2.5)*

CAS Registry Numbers for many compounds can be found in Chemical Abstracts indices, Merck Index, or CAS ONLINE.

##WISWESSER= (STRING). *\$\$ JCAMP-DX v.4.24 (7.2.6)*

The Wiswesser line notation is a precise and concise means of expressing structural formulas as character strings. The basic idea is to use letter symbols to denote functional groups and numbers to express the lengths of chains and the sizes of rings. Wiswesser notation is described in detail by Smith.??? Wiswesser notation for many compounds is tabulated by Smith ??? and by, Grasselli and Ritchey.??? Wiswesser notation should be enclosed in angle brackets if it extends for more than one line, and care should be taken to terminate lines between expressions separated by blank.

##BEILSTEIN LAWSON NO= (STRING). *\$\$ JCAMP-DX v.4.24 (7.2.7)*

The Beilstein Lawson Number is a structural formula clustering code based on the Beilstein System. It is used for locating information in Beilstein databases, including the STN online database. The code contains only structural information and can be generated locally on mi

##BP= (AFFN). Boiling point in degrees C. *\$\$ JCAMP-DX v.4.24 (7.2.9)*

##REFRACTIVE INDEX= (AFFN). *\$\$ JCAMP-DX v.4.24 (7.2.10)*

In the form: ND = 1542A20 (index at 20°C for NaD line).

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 34/40

##DENSITY= (AFFN). \$\$ JCAMP-DX v.4.24 (7.2.11)

Density in g/cm³.

##MW= (AFFN). \$\$ JCAMP-DX v.4.24 (7.2.12)

Molecular weight.

##CONCENTRATIONS= (STRING). \$\$ JCAMP-DX v.4.24 (7.2.13)

List of known components and impurities and their concentrations in the following form, where N stands for name (enclosed in angle brackets if more than one word); C, for concentration; and U, for units of concentration:

##CONCENTRATIONS = (NCU)

(N₁, C₁, U₁)

.....

(N_i, C_i, U_i)

The group for each component is enclosed in parentheses. Each group starts a new line and may continue on following lines.

##SPECTROMETER/DATA SYSTEM= (STRING). \$\$ JCAMP-DX v.4.24 (7.3.1)

This field contains manufacturer's name, model of the spectrometer, software system, and release number, as appropriate, in the form used by the manufacturer.

##INSTRUMENTAL PARAMETERS= (STRING). \$\$ JCAMP-DX v.4.24 (7.3.2)

This optional field is a list of pertinent instrumental settings. Only settings which are essential for applications should be included.

##SAMPLING PROCEDURE= (TEXT). \$\$ JCAMP-DX v.4.24 (7.4.1)

First entry in this field should be MODE of observation (transmission, specular reflection, PAS, matrix isolation, photothermal beam deflection, etc.), followed by appropriate additional information, i.e., name and model of accessories, cell thickness, and window material for fixed liquid cells, ATR plate material, angle and cone of incidence, and effective number of reflections for ATR measurements, polarization, and special modulation techniques, as discussed by Grasselli et al. The LDRs defined below are used in place of, or to supplement,

##SAMPLING PROCEDURE=.

##STATE= (STRING). \$\$ JCAMP-DX v.4.24 (7.4.2)

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 35/40

Solid, liquid, gas, solution, KBr pellet, powder, nujol mull, etc. in infrared spectroscopy.

##PATH LENGTH= (STRING). \$\$ JCAMP-DX v.4.24 (7.4.3)

Pathlength in cm.

##PRESSURE= (STRING). \$\$ JCAMP-DX v.4.24 (7.4.4)

Sample pressure in appropriate units if significantly different from room temperature.

##TEMPERATURE= (STRING). \$\$ JCAMP-DX v.4.24 (7.4.5)

Sample temperature in degrees C if significantly different from room temperature.

##DATA PROCESSING= (TEXT). \$\$ JCAMP-DX v.4.24 (7.4.6)

Description of background correction, smoothing, subtraction, deconvolution procedures, apodization function, zero-fill, or other data processing, together with reference to original spectra used for subtractions.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 36/40

##=.

\$\$ JCAMP-DX v.4.24 (7.5.1)

Comments may be specified by a data-label-flag plus a data-label-terminator, with a null data-

8.3 New GLOBAL LABELS

##APPLICATION= (STRING). See Section 9.3.2.1.2.

##DATACLASS= (STRING). See Section 4.3.2.1.1.

##DICTIONARY= (STRING). See Section 4.3.2.1.3.

##BLOCKID= (STRING).

A name which identifies a BLOCK, more or less like a FILE name. Can serve as a name for the file which is obtained by extracting a BLOCK from a Complex JCAMP-DX file.

##INDEX= (STRING).

Index to contents of a BLOCK. Could be an ordered list of BLOCK-ID's in a LINK BLOCK to index inner BLOCKS more or less like a directory.

##SMILES= (STRING).

SMILES string. encoded according to rules in "SMILES. A Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. Weininger. D.. J. Chem. Inf. Comput. Sci., 1988 . 28 , 31-36 .

##NTUPLES= (STRING). See Section 9.2.2.

##VAR_NAME= (STRING). See Section 9.2.3.

##SYMBOL= (STRING). See Section 9.2.4.

##VAR_TYPE= (STRING). See Section 9.2.5

##VAR_FORM=- (STRING). See Section 9.2.6

##VAR_DIM= (AFFN). See Section 9.2.7.

##UNITS= (STRING). See Section 9.2.8.

##FIRST= (AFFN). See Section 9.2.9.

##LAST= (AFFN). See Section 9.2.10.

###MIN= (AFFN). See Section 9.2.11.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 37/40

##MAX= (AFFN). See Section 9.2.12.

##FACTOR= (AFFN). See Section 9.2.13.

##END NTUPLES= (STRING). See Section 9.2.14.

##PAGE= (STRING). See Section 9.3.1.

##DATA TABLE= (AFFN or ASDF). See Section 9.3.3.

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 38/40

ACKNOWLEDGEMENTS

We wish also to acknowledge technical contributions and other forms of support too numerous to list in detail from the following persons: Charles Anderson, Scott Campbell, Clara Craver, Ray Christopher, Paul DaSilva, Jenny Grasselli, Gregory Hamill, James Hegedus, Mark Holomany, Joe Laughlin, Ronald N. Lee, John Lien, Rich Lysakowski, Don Kuehl, Alex Macur, Howard Mark, Patricia Roush, Abraham Savitzky, David Stranz, Charles Thibault, Brian Toby, James Watt, James Wynne, Tony Williams, and many others.

REFERENCES

1. JCAMP-DX: A Standard Form for Exchange of Infrared Spectra in Computer Readable Form, McDonald R. S., and Wilks P. A. Jr., *Appl. Spectrosc.* 1988, 42(1), p151-162.
2. JCAMP-CS: A Standard Exchange Format for Chemical Structure Information in Computer-Readable Form, Gasteiger, J. et al, *Appl. Spectrosc.* 1991, 45(1) pp4-11. (A partial merging of SMD and JCAMP-DX)
3. Computers in Spectroscopy Part 3, Coates, John P., *Spectroscopy* 1987, 2(4), pp 14-18, 120-22.
4. Use of the JCAMP-DX Format: A Practical Demonstration of Data Exchange McIntyre, P., and Reade, S., in *Computer Methods in UV, Visible, and IR Spectroscopy*, W. O. George and H Willis, CRC Press, Boca Raton FL, USA, 1990, ISBN 0-8517-6323-X.
5. Exchange of Infrared Data Using the JCAMP-DX Format, H. Somberg, *ibid*, pp 149-165.
6. JCAMP-DX: A Standard ?, Davies, A. N., Hill, H. and Linscheid, M., to appear in: *Proceedings of the 4th CIC-Workshop "Software Development in Chemistry"*, Springer Verlag, 1990.
7. Instrument Data Formats: The Current State, Kuehl, D., *Computers in Spectroscopy*, 1991, 6(2), 24, 26-27.
8. NET CDF: An Interface for Scientific Data Access, Rew, R, and Davis, G., *IEEE Computer Graphics and Applications*, July 1990, pp 76-82.
9. The STAR File: A New Format for Electronic Data Transfer and Archiving, Hall, Sydney R., *J. Chem. Information and Computer Sci.*, in press
10. Draft Specification for Revised Version of the Standard Molecular Data (SMD) Format, Barnard, J. M., *J. Chem. Inf., Animation and Comput. Sci* 1990,30, 81-96
11. Specification of Abstract Syntax Notation One (ANSI), ISO 8824:1987(E) with DAD1. Specification of Basic Encoding Rules for Abstract Syntax Notation One (ANSI), ISO 8825:1987(E) with DAD1.
12. Antony N. Davies, Peter Lampen, JCAMP-DX for NMR, *Appl. Spectrosc.*, **47**, 1093-1099 (1993).
13. Peter Lampen, Heinrich Hillig, Antony N. Davies and Michael Linscheid, JCAMP-DX for Mass Spectrometry, *Appl. Spectrosc.* **48**, 1545-1552 (1994).
14. Peter Lampen, Jörg Lambert, Robert J. Lancashire, Robert S. McDonald, Peter S. McIntyre, Douglas N. Rutledge, Thorsten Fröhlich and Antony N. Davies, An Extension to the JCAMP-DX Standard File Format, JCAMP-DX V.5.01 (IUPAC Recommendations 1999), *Pure Appl. Chem.*, **71**, 1549-1556 (1999)

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 39/40

15. A.N. Davies, A New NMR Data Standard for the Exchange and Archiving for Multi-Dimensional Data Sets, *Spectrosc. Eur.*, **11(1)**, 14-20, (1999). (see also http://www.spectroscopyeurope.com/td_col.html) and *Chem. Int.* **21(2)**, 37-39 (1999).
16. A.N. Davies, J. Lambert, R.J. Lancashire, Peter Lampen with W. Conover, M. Frey, M. Grzonka, E. Williams, and D. Meinhart Guidelines for the Representation of Pulse Sequences for Solution-State Nuclear Magnetic Resonance Spectrometry (IUPAC Recommendations 2001) *Pure Appl. Chem.*, **73**, 1749–1764, (2001)
17. J.I. Baumbach, A.N. Davies, Peter Lampen, Hartwig Schmidt, JCAMP-DX – A Standard Format for the Exchange of Ion Mobility Spectrometry Data (IUPAC Recommendations 2001) *Pure Appl. Chem.*, **73**, 1765-1782 (2001)

PROVISIONAL RECOMMENDATIONS

Do Not Cite – Do Not Implement - For Comment Only

08/02/2002 17:30

JCAMP6_2b Draft.doc

Page 40/40