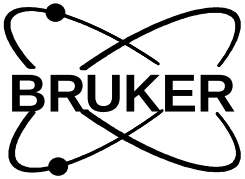
BES^{3}T™

THE BRUKER EPR STANDARD FOR SPECTRUM STORAGE AND TRANSFER

REFERENCE MANUAL



Version 2.0, Spring 2000

Dear Reader, Rheinstetten, Mai 2000

The following documentation defines the BRUKER EPR Standard for Spectrum Storage and Transfer. The BRUKER EPR Software Group has developed this standard to provide the EPR community with a generally accepted and widely available format for simple exchange of EPR spectra, experimental parameters, and related data.

In order to permit additions and corrections from your side, we have added an appropriate form at the end of this booklet. Any contributions are gratefully acknowledged.

Revision History:

Revision 1 : Minor changes of text for clarity.

March 1992: BES^{3}T^{TM} Version 1.1

- introduces the *influent quantity* concept after a suggestion from M. K. Bowman, Argonne National Laboratories,
- *explicitly* handles *complex* quantities instead of hiding them in result sets as result of a BRUKER EPR software group discussion,
- introduces specifications of the Manipulation History Layer for processed datasets.

March 1993: BES^{3}T^{TM} Version 1.1, Revision 1

- Further typing errors removed
- Specifications unchanged

March 1994: BES^{3}T^{TM} Version 1.2

- Extensions .PAR and .SPC changed to .DSC and .DTA, respectively. Consequently, corresponding files are called 'Description File' and 'Data file'.
- ETIM keyword added

July 1996: BES^{3}T^{TM} Version 1.3

- Added new SPL keywords: EXPT, OXS1
- Additional SPN keywords: NONE, B0RS, B0FL, B0TH,
- New argument keywords for EXPT: CW, PLS, SIM, CAL
- New argument keywords for OXS1: IADC, FADC, TADC

February 1998: BES3T^{TM} Version 1.4

- Specification of the parameter entry format changed.

May 2000: BES3T^{TM} Version 2.0

- Redesign and extension of all BES^{3}T^{TM} Layers.

Acknowledgements:

We wish to thank Robert H. D. Nutall from Eastman Kodak Co., Michael K. Bowman from Argonne Nat'l Labs, and Ted McKinney from Rockwell Int'l Science Center, for their ideas and contributions to Version 1.1.

The BRUKER EPR Software Group.

Changes And Additions To Version 1.3, 1.2 and Version 1.1, Revision 1

This section is intended for readers that are familiar with version 1.1, Rev. 1 of BES^{3}T™ and merely lists changes of and additions to the standard description introduced for BES^{3}T™, 1.1, Rev. 2. New sections, chapters, paragraphs are indicated with (N) below, while changes and additions to existing portions are indicated with (C).

- (C) IV.2.5 MHL Entries: Typo removed
- (C) IV.3.1.i Order of axis assignments is explicitly specified now.
- (C) IV.3.1.vi Intensity transformations also appliccable to non-integral data.
- (N) IV.3.2, Table 11, SPN Names: New ETIM keyword.
- (C) Appendix B.1: Conversion utility will be available together with BES^{3}T™-compatible software.
- (C) Appendix A.2.2: Erroneous '< >' around DESC version indication in example removed.
- (C) Appendix B.2: Removed

- (C) Throughout: 'Parameter File' replaced by 'Description File'
- (C) Throughout: 'Spectrum File' replaced by 'Data File'

The following section describes the changes and additions made to Version 1.2 of BES^{3}T™:

- (N) IV.3.2, Table 9, SPL Names: New EXPT and OXS1 keywords
- (N) IV.3.2, Table 11, SPN Names: New NONE, B0RS, B0TH, B0FL keywords
- (N) IV.3.2, Table 11, New argument keywords for OXS1: IADC, FADC, TADC
- (N) IV.3.2, Table 11, New argument keywords for EXPT: CW, PLS, SIM, CAL

The changes and additions made to Version 2.0 are:

- manifold

Notes:

I. Introduction

I.1 Objectives

The **B**RUKER **E**PR **S**tandard for **S**pectrum **S**torage and **T**ransfer, herein referred to as BES^{3}T™, shall serve the following purposes:

- *Permit worldwide exchange of EPR spectra in hardware-independent computer-readable format.*
- *Supplement the raw spectrum data with a rich context for interpretation and presentation.*
- *Provide standardized storage formats for all EPR-related data, from simple Intensity-vs.-Field spectra to multi-dimensional datasets.*
- *Preserve the flexibility of individual solutions by means of a layered structure and permit BES^{3}T™ use also for local spectrum storage.*

BES^{3}T™ does *not* make any assumptions about

- *The hardware used for spectrum exchange, e.g. floppy disks, CD-ROM, modems, etc.*
- *The source of the spectrum data, e.g CW-EPR, FT-EPR, simulations, etc.*

In the following, the format of BES^{3}T™ is published. Section II presents an overview of the standard and of the ideas behind it. Section III focuses on the concepts of the standard by dilating upon the structures of data and parameters that can be stored, and how they are arranged. Technical aspects like file structures, data formats, keyword tables and so on are precisely defined in section IV. Finally, the appendices present the interested reader with some hints about generating BES^{3}T™ datasets and a few examples, plus a discussion of the migration from the current BRUKER standard to the BES^{3}T™, and they treat a number of complicated or rare cases in detail.

I.2 Definitions

The definition of a spectrum storage standard is invariably a mixture of spectroscopy and computer science. Therefore, a few points are defined below, which otherwise might cause confusion or are used with a specific meaning throughout this document. The reader may well skip these lines and return later on if something seems to be unclear.

I.2.0 i) Notation

The following pages will show many examples of text that represents user input, or the contents of a file, and the like. In such cases, the following notation is used:

- Non-proportional writing **`like this`** indicates computer text.
- Text in angular brackets is to be replaced by its meaning, e.g **`<Version>`** could mean **`1.2`**.
- Text in square brackets is optional, like **`[parameter]`**.
- Items separated by **`|`** are exclusive, e.g. if **`[a|b|c]`** is shown, there is an optional **`a`** or **`b`** or **`c`**.
- The notation **`A=B`** means, that **`A`** can be replaced by **`B`**. For instance **`unit=Gauss|Tesla`** describes, that **`Gauss`** and **`Tesla`** are valid replacements for **`unit`**.
- If a definition uses [], | or = as text symbols, they are enclosed in quotes (’), to distinguish them from the above meta-symbols.

I.2.0 ii) Layer

In software terms, a layer is something like a screen around an object, providing the outside world with a uniform interface to the object. Often, there are many such layers between the user and a hardware-object. For example, a print program may have layers, the uppermost of which provides commands like 'Select Font', 'Print', and 'Form-Feed', while the lowermost works with raw ASCII and control characters.

I.2.0 iii) ASCII Format

A data item given in ASCII format can be interpreted as a sequence of 8-bit bytes, each of which is assigned a character given in the **A**merican **S**tandard **C**ode for

Information **I**nterchange character table. In other words, ASCII items are likely to be text items.

I.2.0 iv) Swept Quantity

A quantity which is *varied* during an experiment in order to record information about another quantity. Multi-dimensional experiments employ multiple swept quantities, one for each axis of the final plot.

I.2.0 v) Acquired Quantity

A quantity which is *recorded* during an experiment in dependence of one or more swept quantities. There are experiments, in which more than one acquisition quantity is recorded, e.g. when using quadrature detection.

I.2.0 vi) Influent Quantity

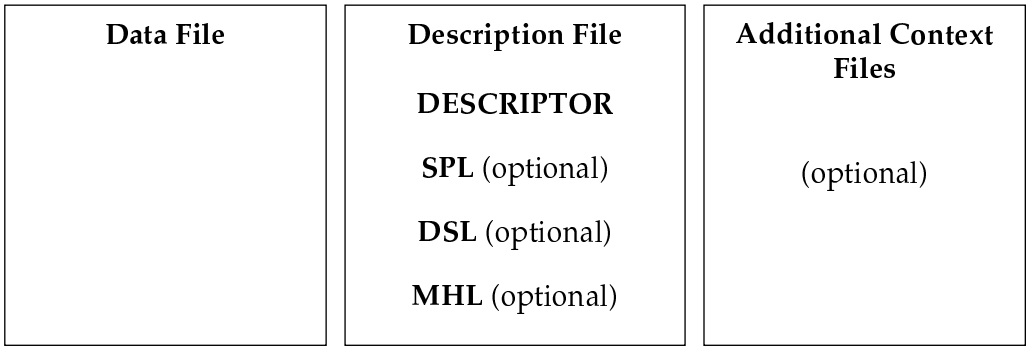
A quantity that may affect the outcome of an experiment, but cannot be freely set because it must either obey a certain rule (like the magnetic field value in a temperature-controlled ENDOR experiment with shifting EPR-lines) or is technically not accessible (like the atmospheric pressure in an open-air experiment). In order to interpret experimental results, recording influent quantities is at least desirable.

I.2.0 vii) Grid

A specific arrangement of cells along any swept axis, into which the corresponding values of an acquired quantity have to be filled to reconstruct an experiment.

II. General Structure of the BES^{3}T™

This chapter outlines the information that is stored in BES^{3}T™ datasets and how it is stored. A BES^{3}T™ dataset can be divided into three parts: A Data File, a Description File, and additional context files. The following figure shows this structure.



The *Data File* contains experimental data points, but no other information. This data may be stored in many different formats, most of which are *binary*, i.e. in the format a computer would store the data in its memory. Looking at such a Data File with a text editor will thus not tell you very much about its contents. In its simplest form, a Data File may consist of a sequence of 1D-spectrum points in a binary 32-bit integer format, which makes it just the same as a current BRUKER '.SPC' file. However, more sophisticated data structures may be stored in the file, which are recognized by looking at the Description File.

The *Description File*, in contrast to the Data File, uses *ASCII* characters to store its information. If you look at a Description File with a text editor, you will be able to read its contents, albeit in an abbreviated form. The Description File consists of two main parts.

The *Descriptor* part of the Description File (see III.2 on page 14) provides the information required to read in and present the data in the Data File. In other words, after reading the Descriptor and the Data File, a suitable program has everything it needs to produce a publication-ready plot of the data, including axes, labels, units, a title, and, of course, the data points. In addition, the descriptor indicates whether the Data File contains experimental or manipulated data.

If the Data File contains experimental data, the Description File can contain experiment parameters. Because not all spectrometers are the same, this information is split into a *Standard Parameter Layer* (SPL, see III.3.1 on page 16), applicable to all spectrometers, and a *Device Specific Layer* (DSL, see III.3.2 on page 17), which accounts for the particularities of specific equipment.

If the Data File contains manipulated data, the Description File can also contain a description of how the data was obtained from which source(s). This part is called the *Manipulation History Layer* (MHL, see III.4 on page 17).

Several *Additional Context Files* are defined in this version of the standard to represent nonlinear sweeps and to report values of influent quantities. Other definitions may be added in the future, if required.

III. BES^{3}T^{TM} Concepts

This section gives a definition of the standard, which is intelligible not only to the computer expert, but everybody the BES^{3}T^{TM} is dedicated to: The scientist involved with EPR spectroscopy, casually or constantly, expert or novice.

III.1 The BES^{3}T^{TM} Data Structures

An EPR Spectrum Storage Standard must provide ways to store many different kinds of data arrangements. Often, the spectrum to be stored will simply consist of a series of points which lie evenly spaced along a field axis; thus only the data points, the center field, and the field range need to be given in order to completely describe the spectrum. However, imagine a case where a nonlinear field sweep was performed for several temperatures, thus yielding a '2D-spectrum', intensity vs. field vs. temperature. It is obvious that such a dataset requires a more sophisticated storage format than the simple example does.

Therefore, the various data structures supported by the BES^{3}T^{TM}, and all items required to describe them shall be treated first. In this treatment, please observe the distinction between *swept quantities, acquired quantities*, and *influent quantities* (cf. I.2 on page 7) and keep in mind that a BES^{3}T^{TM} *Data File* always contains *acquired quantities*, while information about *swept quantities* and *influent quantities* is found in the *Descriptor* or in *Additional Context Files*. Together, all these quantities represent the course of an experiment and its results. The principles of these representations shall be stated now; if you are interested in the factual file storage format used, please refer to section IV on page 19.

III.1.1 Swept Quantities

There are three ways to represent the values that a swept quantity has assumed during an experiment:

III.1.1 i) Indexed data

This representation is used if a quantity was swept in *constant steps*. In this case, the swept quantity defines an *equidistant grid* into which the acquired data values can be filled. This grid is defined by *minimum* and *width* values and a *number of*

points; there is no need to store individual values of the swept quantity. Indexed data is by far the most common structure.

III.1.1 ii) Index gauged data

This representation is used if a quantity was swept in *variable* steps, i.e. in a non-linear manner. In such a case, *all* values the swept quantity assumed during the experiment are stored. The flexible grid thus defined is referred to as *index-gauge-array*.

For multidimensional experiments, the above formats assume that the swept quantities do define a *regular grid* to fill in the acquired values, e.g. that for *different* values of a quantity A the *same* sweep of quantity B is performed, and that quantity A is held *constant* during the sweep of B. If this is not the case, the different sweeps would have to be stored *individually*, which requires a *third* data format. The same data format could be used to store the result of data-reducing algorithms, like peak picking.

Consider the result of a peak-picking on a series of temperature-dependent spectra. Previously identical field sweeps at different temperatures have been reduced to a varying number of peaks for each spectrum, because lines may coalesce or separate. Obviously, each point (peak) is most easily defined by a coordinate triple (temperature, field-position, intensity). Therefore, for these cases it seems advisable to drop the concept of a *grid* defined by the swept quantities, and to replace it by the most general way of coordinate storage, namely an N-tuple.

III.1.1 iii) N-tuple data

If a swept quantity is given as N-tuple data, its particular values associated with *each acquisition data point* are given in a so-called *tuple-array*, which has therefore the *same size* as the acquired data array. In the example given above, 240 peaks found at 13 temperatures would therefore be represented by a temperature array, a position array, and the Data File containing the acquired quantity, and *each* of them would have 240 elements. Therefore, N-tuple representation consumes a lot of space in exchange for its flexibility.

Obviously, for a *1D-dataset*, index gauged and N-tuple data contain the same information, since the construction of an irregular 1D-grid (index-gauge-array) which is filled with acquired values is equivalent to the formation of swept/acquired value pairs (tuple-array). For 1D-datasets, it is recommended to call the

representation 'index-gauged' if ignoring the gauge-array might be affordable (e.g. for slight x-irregularities), and 'N-tuple' otherwise.

III.1.2 Acquired Quantities

III.1.2 i) Real Or Complex Data

As was mentioned above, the actual acquisition data resides in the Data File. Acquisition data consists of individual data points, which for the purpose of data presentation or interpretation are arranged in the way defined by all swept quantities.

It is well known that some observables of an experiment are closely linked to another quantity by the *Kronig-Kramers relation*. Mathematically seen, such pairs can be treated like *complex quantities*. By *explicitly* handling complex quantities, BES^{3}T™ permits data acquisition and manipulation software to recognize the physical and mathematical relationship of their components.

III.1.2 ii) Result Sets

In the vast majority of cases, only one (real or complex) quantity is sampled and recorded during an experiment. However, if multiple sampling methods like simultaneous EPR and optical detection will become more common, or if datasets shall be used to store more general information than just spectra, it might be required to associate more than a single value with each data point. BES^{3}T™ supports such structures by permitting so-called *result-sets*.

A result-set is a structure that is imposed upon every single data item in a Data File, which permits it to hold the acquired data of *any given number of experiments*. However, since all this data resides in the same dataset, there is only one set of parameters, which implies that the experiments have been performed in *parallel* with *identical parameters*. The Descriptor of a BES^{3}T™ dataset exactly specifies the result-set structure.

III.1.3 Influent Quantities

In more complicated cases, swept and acquired quantities may not provide all the information required to interpret an experiment. Sometimes, there are quantities that must be adjusted during an experiment in order to keep it nicely running, or

one may suspect that the value of an independent quantity may influence the result of an acquisition. In both cases, two statements apply:

- The additional quantity of interest is not a swept quantity, because it cannot be freely set.
- The additional quantity does not necessarily vary between two single point acquisitions, it may be much more inert than the acquired quantity.

Together, these statements mean that an influent quantity must be stored in a data structure *comparable* to that used for the acquired quantity of a certain experiment, but the *dimensionality may be lower*. As an example, consider ENDOR measurements with a single crystal, which are carried out at various temperatures while the crystal is turned. Presumably, the magnetic field will have to be adjusted after a change of the sample temperature or the crystal orientation, to compensate for EPR-line shifts, but certainly not during the ENDOR sweep. Therefore, the acquired quantity will require a three-dimensional data structure I(temp., angle, freq.), while a two-dimensional structure B_0(temp., angle) will suffice to store the adjusted values of the magnetic field.

The values of influent quantities are stored in Additional Context Files, whose structure is defined by the Descriptor of the dataset, in a format very similar to that used for the Data File.

Since so many different data structures are allowed for BES^{3}T^{TM} data, a raw Data File requires considerable description in order to be correctly interpretable. Therefore, there is a *Descriptor* at the beginning of each Description File, whose entries will be described now.

III.2 Description File, Part 1: The Descriptor

The Descriptor part of a Description File consists of:

- A *Header*, which signals the start of the Descriptor and also includes the BES^{3}T^{TM} version number,
- A *Dataset type and format* section, which determines whether the associated spectrum data is
  - Experimental or manipulated,

  - Motorola or Intel byte format,
  - The dimensionality of the data and sweep representations as defined in III.1 on page 11,
  - The format of an optional *Result-Set structure*,
  - The dimensionality of optional influent quantities and their sweep dependencies,
- A *Data item format*, which allows data to be stored as integer numbers, real numbers, etc.,
- *Data ranges and resolutions* which define axis areas covered by the data as well as point spacings,
- *Transformation parameters*, which permit space-saving integer number storage for floating point intensity values, if their range is not too large,
- *Documentational Texts*, which permit the specification of a dataset title, axis labels, and units.

In addition, comments can be introduced wherever desired in order to make a Descriptor more readable.

The Descriptor entries as listed above together with the Spectrum Data File contain everything required to produce a publication-ready plot of a whole dataset or any part of it. *Therefore, these two elements constitute the minimum of a BES^{3}T™ dataset*.

III.3 Description File, Part 2a: Experimental Parameters

The interpretation of experimental datasets may be very difficult if one does not know how the data was acquired. Therefore, an experimental dataset should include the setting of all relevant spectrometer parameters. Since not all spectrometers are the same, and some units have parameters that others don't, it is advisable to distinguish two classes of parameters:

- Standard Parameters, which *must* be given in order to perform a specific experiment on *any* spectrometer, e.g. microwave power for an EPR experiment,
- Device Specific Parameters, which are still required to reproduce a specific experiment but concern a setting that is specific to a *particular*

instrument, like a cavity selector switch.

Both classes have members that are irrelevant to certain experiment types (like the microwave pulse width in a CW-experiment). Such members are simply omitted from the description.

It is important to notice that the values of influent quantities are not given in the experimental parameter section, because these quantities are either completely uncontrollable or have to obey a certain rule, i.e. the *rule*, if any, must be reapplied in order to reproduce the experiment, rather than to re-set the original *value*s of an influent quantity. If the rule is applied by the machinery, like in an FF-lock experiment, the setting of a switch will reflect its application, while the execution of manual adjustments can only be represented in a comment.

III.3.1 The Standard Parameter Layer

The *Standard Parameter Layer* contains all physically meaningful acquisition parameters in a device *independent* format. Therefore, the Standard Parameter Layer is compatible for *all* BES^{3}T™ datasets and makes sense on *all* spectrometers, i.e. it provides something like a 'common parameter language' to BES^{3}T™ datasets. This language can express information about

- The operator performing the experiment,
- Date and time of the acquisition,
- The name or formula and the state of aggregation of the sample,
- Sample concentration,
- The temperature and spatial orientation of the sample,
- All magnetic field parameters, fixed or swept, including field modulation,
- All microwave parameters,
- Detection parameters like time constant or absolute amplifier gain,
- RF-frequency parameters for ENDOR, including frequency modulation,
- Swept parameters other than magnetic field or RF-frequency, like temperature or goniometer angle, in a flexible scheme.

Especially the last point is quite important since it indicates that a BES^{3}T™ Description File can describe acquisitions along *arbitrary sweep axes*.

Standard Parameter Layer information is given by so-called keywords and associated values. In section IV on page 19, a precise definition of all currently defined keywords is found.

III.3.2 The Device Specific Layer

The *Device Specific Layer* (DSL) contains data for special, often optional devices. Therefore, the Device Specific Layer is guaranteed to be compatible for all BES^{3}T™ datasets produced by *identically configured* instruments. In addition, spectrometer software will often support both *downward* and *upward compatibility* of Device Specific Layers, i.e. they will make sense on *all* spectrometers of a *specific manufacturer*.

The dictionary of device specific parameters is not part of this standard, since each manufacturer is free to define these parameters on its own. BES^{3}T™ only defines the technical rules that must be obeyed by all manufacturers creating Device Specific Layer definitions; these are given in section IV on page 19. Device manufacturers are required to provide a dictionary of all the keywords their devices use. Separate documentation on Device Specific Layer keywords as defined for BRUKER will be available.

III.4 Description File, Part 2b: Manipulation History

The interpretation of manipulated datasets may be very difficult if one does not know how the data was obtained. Therefore, a manipulated dataset should include an indication of the original (unmanipulated) dataset and a description of all processing steps performed. Usually it is also helpful, if the dataset contains the experimental parameters of the original dataset.

Since not all processing softwares are the same, it is advisable to classify this layer as software dependent. Therefore, the Manipulation History Layer is guaranteed to be compatible for all BES^{3}T™ datasets produced by *identical processing packages*. In addition, it is hoped that software vendors producing new processing programs will adopt the 'MHL Language' used by an already established program, in the same way as 'HPGL' became a generally used language for graphics descriptions. In this way, Manipulation History Layers could make sense with *many different processing packages*.

The dictionary of Manipulation History languages is not part of this standard, since in principle each software manufacturer is free to define this language on its own. BES^{3}T™ only defines the technical rules that must be obeyed by all manufacturers creating Manipulation History definitions; these are given in section IV on page 19. Software manufacturers not adopting an existing language are required to provide a dictionary of their Manipulation History Language. Separate documentation for BRUKER processing packages will be available.

IV. BES^{3}T™ Data Format and Implementation

The following pages are intended as an extension to the BES^{3}T™ description given in section III on page 11. While that section has shone light on the BES^{3}T™ concepts, section IV defines all file names and structures, data formats, keywords, etc. The information presented herein is required for application programmers that want to read or produce BES^{3}T™ files.

As introduced before, the acronym SPL refers to the *Standard Parameter Layer*, DSL refers to the *Device Specific Layer*, and MHL abbreviates *Manipulation History Layer*.

IV.1 File Specifications

IV.1.1 Filenames

IV.1.1 i) The Data File

The Data Filename syntax is prescribed as

<Name>.DTA

independent of the operating system that is used to create the file. The characters allowed in **<Name>** must obey local operating system requirements.

IV.1.1 ii) The Description File

The Description Filename syntax is prescribed as

<Name>.DSC

independent of the operating system that is used to create the file. The characters allowed in **<Name>** must obey local operating system requirements.

IV.1.1 iii) Additional Context Files

The context filename syntax is prescribed as

<Name>.<Extension>

independent of the operating system that is used to create the file. The characters allowed in **<Name>** must obey local operating system requirements. **<Extension>** depends on the contents of the file. Table 1 displays the currently defined context filename extensions.

.<Extension>	*Contents*
.GFn	Index-gauge-array for axis n, n = 1,2,...
.TPn	Tuple-array for axis n, n = 1,2,...
.IQn	Influent quantity data, n = 1,2,...

Table 1: Filename extensions for Additional Context Files

Note: In previous BES^{3}T^{TM} versions these extensions were called **XGF**, **YGF**, **ZGF** (now **GF1**, **GF2**, **GF3**) and **XTP**, **YTP**, **ZTP** (now **TP1**, **TP2**, **TP3**).

IV.1.2 File Formats

IV.1.2 i) The Data File

The Data File exclusively contains acquisition data in numeric form. The number format (integer, floating point, etc.) is given by the descriptor entry *Data Item Format*, the arrangement of the numbers is defined by *Dataset Type and Format*. See IV.3.1 on page 27 for descriptor entries.

IV.1.2 ii) The Description File

The Description File exclusively contains descriptive and parameter data in ASCII form. See IV.2 on page 22 for individual entry format descriptions and IV.3 on page 27 for keyword tables.

The individual parts of the Description File are arranged as follows:

- *Descriptor*, mandatory (see III.2 on page 14)
- *Standard Parameter Layer*, optional (see III.3.1 on page 16)
- *Device Specific Layer*, optional (see III.3.2 on page 17)
- *Manipulation History Layer*, optional (see III.4 on page 17)

Although the syntax of the Description File would allow the detection of split and interspersed layers, the current version of BES^{3}T^{TM} does not permit such constructions.

IV.1.2 iii) Additional Context Files

The format of Additional Context Files depends on their extension. Table 2 displays the currently defined context filename formats.

.<Extension>	*Format*
.GFn	Binary numeric, Data Item Format **AXnFMT**, n = 1,2,...
.TPn	Binary numeric, Data Item Format **AXnFMT**, n = 1,2,...
.IQn	Binary numeric, Data Item Format **IQnFMT**, n = 1,2,...

Table 2: Additional Context Files data formats

For Data Item Formats, see IV.3.1 iv on page 30. For **.IQn** Files, the arrangement of the numbers is defined by Dataset type and format (mandatory) on page 27.

IV.2 Description File Entry Formats

IV.2.1 General

There are two formats for Description File entries, one of which is meant for actual data storage, while the other one may be used for file-documentation purposes.

IV.2.1 i) The Parameter entry format

<keyword> [content] [comment] CR[LF]

content = <value>[,<value>[,...]] [unit]

content = header [values[,values]]

header = {<n>;<dim$_1$>,..,<dim$_n$>;<default> [unit]}

values = ['['range$_1$,..,range$_n$']'] <value>[,<value>]

range$_i$ = <x$_j$>[..<x$_k$>] j,k∈[1..dim$_i$]

unit = '['[<prefix>]<physikal_unit>']'

i.e. a keyword followed by an optional **content** and an optional **comment**, terminated by a carriage return (**CR**) and an optional line feed (**LF**). The **content** field may be absent if the mere presence of the **keyword** defines a switch setting. The format of the **comment** field is defined in IV.2.1 ii on page 24.

The **content** can either be a comma-separated list of single values with an optional **unit**, or it can be represented as an n-dimensional matrix. The later is split into a **header** and a list of **values**. The **header** describes the number of dimensions, the number of values for each dimension, a default value, which is used for matrix elements which does not occure in the **values** list and an optional **unit**.

The **values** are given as a list of coordinate-value pairs like **[1,12,3] 17.36**, which enumerate the matrix elements, that differ from the default value, given in the **header**. Coordinates can be omitted, if the values belong to consecutive matrix elements. Matrix elements are enumerated, such that the leftmost index varies fastest. The coordinate of the first value in a **values** list is implicitly

[1,1,..,1]. If the coordinate contains ranges (e.g. **[1,2..5,1..8]**, the following value list contains all elements of this (e.g. 4x8) sub-matrix.

A **unit** is composed of an optional **prefix**, e.g. M for Mega or n for nano and a physical unit like T or Hz, enclosed in square brackets. A list of valid prefixes is given in Table 3.

Prefix	*Value*	*Meaning*
E	10^{18}	Exa
P	10^{15}	Peta
T	10^{12}	Tera
G	10^{9}	Giga
M	10^{6}	Mega
k	10^{3}	kilo
h	10^{2}	hecto
D	10^{1}	Deka
d	10^{-1}	deci
c	10^{-2}	centi
m	10^{-3}	milli
u	10^{-6}	micro
n	10^{-9}	nano
p	10^{-12}	pico
f	10^{-15}	femto
a	10^{-18}	atto

Table 3: Prefixes for physikal units

In addition to the above definitions **\CR[LF]**, i.e. a backslash at the end of a line, can be inserted at any position to make long parameter lines more readable.

For some keywords, the value field must be identical to a *member* of a predefined *set* given with the keyword definition.

If a keyword requires a *numeric* value, its notation must either be

- Integer, like **123**, or
- Floating point, like **123.45**, or
- Exponential, like **123.45e67**.

Usually, values may be given in any notation. In some explicitly stated cases, an integer value is required. A preceding minus sign (-) is allowed for all three types, a plus sign (**+**) is optional. It is recommended to use integer multiples of 3 when specifying an exponent ('engineer's notation').

Finally, some keywords require a *text* parameter, which must be enclosed in quotes ('), like **'this is a text'**.

White space (spaces, tabs) is only allowed between **keyword**, **content** and **comment**, before the **unit** field, between **header** and **values**, between the coordinate and the value of a **values** list item and within quotes, i.e. in text parameters.

Neither the mandatory nor the optional parameters of the Description File have default values, i.e. missing keywords leave the corresponding values undefined.

IV.2.1 ii) The comment format

```
comment CR[LF]

comment = * <comment_text>
```

An line that begins with ***** is skipped and may be used for documentation purposes. An example of a comment line is

```
* This is a comment!
```

As stated before, an in-line **comment** can be appended to parameter entries. The parameter **content** must be followed by at least one white space and the comment must also start with a *****, like in

```
DATE 13-03-91   * That is my birthday!
```

The use of comments is strongly recommended even for automatically generated files.

IV.2.2 Descriptor Entries

The Descriptor must begin with the header entry

```
#DESC <Version>
```

where **<Version>** gives the BES^{3}T™ version number (currently 2.0). All following lines are assumed to contain descriptor entries, until another line starting with a **#** is encountered.

Descriptor entries themselves use the parameter entry format or the comment format. See IV.3.1 on page 27 for Descriptor keywords and values. Within the descriptor, legal keywords may appear in arbitrary order.

IV.2.3 Standard Parameter Layer Entries

The SPL must begin with the header entry

```
#SPL <Version>
```

where **<Version>** gives the SPL specification version number (currently. 2.0). All following lines are assumed to contain SPL entries, until another line starting with a **#** is encountered.

SPL entries themselves use the parameter entry format or the comment format. See IV.3.2 on page 33 for SPL keywords and values. Within the SPL, legal keywords may appear in arbitrary order.

Note that if inconsistencies between SPL and DSL entries occur, e.g. due to improper Description File editing, the *SPL* entry takes precedence.

IV.2.4 Device Specific Layer Entries

The DSL must begin with the header entry

```
#DSL <Version> <Manufacturer>
```

where **<Version>** gives the DSL specification version number (currently 2.0). All following lines are assumed to contain DSL entries, until another line starting with a **#** is encountered.

Parameter entries, following the DSL header, are interpreted as global parameters. They describe the spectrometer setup and the acquisition method.

The parameters of a particular device are collected in a device block. Such a block starts with the entry

```
.DVC <Device ID>,<Version>
```

where **<Device ID>** is the name of the device and **<Version>** gives the version number (e.g. 1.0). All following lines are assumed to contain DSL parameter entries, until another line starting with a **.** or a **#** is encountered. Of course, multiple device blocks are allowed.

Device block entries themselves use the parameter entry format or the comment format. The manufacturer of a device is required to publish a glossary of the DSL entries available (see *BRUKER DSL dictionary* for DSL keywords and values of BRUKER devices). Within a device block, legal keywords may appear in arbitrary order.

IV.2.5 Manipulation History Layer Entries

The MHL must begin with the header entry

```
#MHL <Version> <Software Manufacturer>
```

where **<Version>** gives the MHL version number (currently 2.0). All following lines are assumed to contain MHL entries, until another line starting with a **#** is encountered. Manipulation History entries must appear in chronological order.

The parameters of a particular manipulation are collected in processing blocks. Such a block starts with the entry

```
.PRC <Processing ID>,<Version>
```

where **<Processing ID>** identifies the processing and **<Version>** gives the version number (e.g. 1.0). All following lines are assumed to contain MHL parameter entries, until another line starting with a **.** or a **#** is encountered.

Processing block entries use the parameter entry format or the comment format. The manufacturer of a processing package is required to publish a glossary of the MHL entries available.

IV.3 Keywords, Values, And Meanings

IV.3.1 Descriptor Keywords

Note that several descriptor entries are mandatory.

IV.3.1 i) Dataset type and format (mandatory)

These entries define the dimensionality of the experimental and influent data and the representation of swept, acquired, and influent quantities in a dataset. Several keywords are defined in Table 4, which must be present in a Description File. The number of axes and the number of influent quantities are implicitly defined by the **AXnTYP** and **IQnTYP** keywords given.

The sweep representation types of **AXnTYP** have been described in III.1 on page 11. The types for all swept quantity axes can be combined in many ways. Valid representations are defined by the following rules:

- An n-dimensional dataset uses all axes from axis-1 to axis-n, i.e. gaps in the list of used axes (e.g. x- and z-axis only) are not allowed.
- For multi-dimensional datasets, the first index (x-index) of the spectrum data varies fastest, the last index slowest. This directly implies that the first axis corresponds to the physical quantity of the experiment that is swept fastest.
- For each **IDX** axis, Data Range and Resolution parameters define the point grid (see Data ranges and resolutions (mandatory) on page 31).
- For each **IGD** axis, there is an index-gauge file as explained in IV.1.1 iii on page 19 and IV.1.2 iii on page 21, which defines the point grid.
- For each **NTUP** axis, there is a tuple-file containing the coordinates to be associated with the acquired data in the Data File.
- The implicit spectrum drawing sequence is equal to the sequence of data points in the file. Of course, other drawing sequences may be provided by software.

Keyword	*Value*	*Meaning*
DSRC		Source of data
	EXP	Experimental data
	MAN	Manipulated Data
BSEQ		Byte format of data
	BIG	Big endian ('Motorola' format)
	LIT	Little endian ('Intel' format)
IKKF		Complex-data flag
	REAL	Real data, no imaginary part
	CPLX	Real and imaginary part interspersed
AXnTYP		Data type for axis n, n=1,2,...
	IDX	Indexed
	IGD	Index-gauged
	NTUP	N-tuple
IQnAXS	<dep>	Axes which define value space for influent quantity n, n = 1,2,...

Table 4: Mandatory dataset type and format keywords

The following examples demonstrate applications of the above rules:

a) **AX1TYP IDX**

means the Data File consists of a 1D-array of intensity points, and the abscissa (x-) grid is defined by data range and resolution parameters described below.

b) **AX1TYP NTUP**
AX2TYP NTUP

means the Data File consists of an array of intensities i_n, which have to be associated with the contents of two tuple-files (**.TP1** for x and **.TP2** for y) to yield triplets (i_n,x_n,y_n) representing three-dimensional coordinates.

c) **AX1TYP NTUP**
AX2TYP IDX
AX3TYP IGD

means the Data File consists of an array of intensities i_n, which have to be associated with the contents of a tuple-file (**.TP1** for x) to yield pairs (i_n,x_n). These have to be filled into a two-dimensional grid defined by data range and resolution parameters for y and an index-gauge file **.GF3** for z. When filling the grid, proceed along the y-axis first, then repeat for the next z-value, etc.

d) no **AXnTYP** defined

means that no axis has been swept at all, i.e. the 'Data File' is empty. The corresponding parameter set may still describe a setup of static experiment parameters.

IV.3.1 ii) Result Set Specification

In addition to defining the 'complexity' of acquired values, the **IKKF** keyword is used to define the structure of *result-sets* as described in III.1.2 ii on page 13 by specifying the number of elements. For this purpose, the keyword must be followed by a list of values separated by commas, like in

```
IKKF CPLX,REAL,CPLX      * Three acquired quantities
```

This shows that the common single-value Data File structure is just the simplest case of a result-set. For *all* entries describing acquired quantities (i.e. starting with **I...**), there must be one value field per result set member.

In the Data File, a result set is stored in an *interspersed* arrangement of acquired quantities. If the quantities were named Q1, Q2, and Q3, then their arrangement would be $Q1_1$, $Q2_1$, $Q3_1$, $Q1_2$, $Q2_2$, $Q3_2$, $Q1_3$, etc. Like this, the generation of large result set files is possible also for small computers which cannot hold the whole file in memory.

IV.3.1 iii) Influent Quantity Specification Format

The <dep> value for the **IQnAXS** keyword represents a sequence of axes separated by commas, which define the value space for the corresponding influent quantity. The sequence consists of axis-numbers

```
1,3,...
```

The associated influent quantity file (see IV.1.1 iii on page 19 and IV.1.2 iii on page 21) has exactly the same structure as a Data File pertaining to the axes given in <dep> would have.

IV.3.1 iv) Data item format (mandatory)

For each data item (intensities, x-, y-, z-values, and influent quantities, if given), such an entry specifies the data storage format used as shown in Table 5. Hence, it is well possible to e.g. store a spectrum as an array of double precision floating point values. Of course, for x-, y-, and z-data the entry only makes sense if there is coordinate data for these axes, i.e. if they use index-gauged or n-tuple representation.

Keyword	*Value*	*Meaning*
IRFMT		Format of real intensity values
	C	Character (8 bit)
	S	Short (16 bit)
	I	Integer (32 bit)
	F	Float (32 bit)
	D	Double (64 bit IEEE)
	A	ASCII (CR-separated *text* format)
	0	No data
	N	No data (instead of 0)
IIFMT		Format of imaginary intensity values, settings as above
AXnFMT		Data format (**IGD** or **NTUP** only), settings as above
IQnFMT		Influent quantity n format (if **IQnTYP** given) **n** = 1..9, settings as above

Table 5: Data item format keywords

IV.3.1 v) Data ranges and resolutions (mandatory)

The keywords displayed in Table 6 are used to describe the range covered by a swept quantity, and the number of data points acquired during a sweep. In addition, they define the number of points present in (optional) influent quantity files.

Keyword	*Value*	*Format*	*Meaning*
AXnMIN	xmin	any	Minimum axis-value
AXnWID	xwid	any	Width of axis-range swept
AXnPTS	xpts	integer	Number of points acquired

Table 6: Data range and resolution keywords

As described in III.1.1 i on page 11 and IV.3.1 i on page 27, indexed representation of a sweep only requires range and resolution information, axis-coordinates themselves are not stored. In this case, the x-value of the N-th spectrum point is calculated by

$$x = ax1min + N * ax1wid/(ax1pts-1) \quad , \quad 0 <= N <= ax1pts-1$$

For axes given in **IGD** (index gauged) format, the **AXnPTS** entries specify the number of points the corresponding **.GFn** file contains. The keywords **AXnMIN** and **AXnWID** can be used to determine the range covered by a specific coordinate for scaling purposes. For axes given in **NTUP** (n-tuple) format, **AXnPTS** determines the **.TPn** file size and **AXnMIN** and **AXnWID** are again used specify the range covered by a coordinate.

IV.3.1 vi) Intensity transformation (optional)

Intensity values in the Data File can be linearly transformed for interpretation using the entries shown in Table 7.

Keyword	*Value*	*Format*	*Meaning*
IRTOF	irtof	any	Offset of transformation
IITOF	iitof	any	
IRTSP	irtsp	any	Slope of transformation
IITSP	iitsp	any	

Table 7: Intensity transformation keywords

This is especially useful whith integral values IR_i which are then transformed to floating point quantities IR_f by

IR_f = irtof + IR_i * irtsp (analogous for imaginary part)

If no transformation is required, **IRTOF**/**IITOF** and **IRTSP**/**IITSP** are omitted.

For *result-sets* as defined in III.1.2 ii on page 13, these keywords must be followed by a *list of values* separated by commas, like in

```
IRTOF 13.6,12.8   * Offset pairs for double-detection
```

Offset and slope values must either be given for each member of a result-set, or the keywords must be omitted.

IV.3.1 vii) Documentational text (optional)

For display and presentation purposes, some text fields may be given in the Descriptor:

Keyword	*Value*	*Meaning*
TITL	<text>	Title of dataset (up to 64 characters)
IRNAM	<text>	Axis labels (up to 16 characters)
IINAM	<text>	
AXnNAM	<text>	
IQnNAM	<text>	
IRUNI	<text>	Axis units (up to 8 characters)
IIUNI	<text>	
AXnUNI	<text>	
IQnUNI	<text>	

Table 8: Documentational text entries

Note that when calculating functions of two spectra in which the same physical quantity is expressed in different units, improper results will be obtained, since the processing software is not required to interpret the documentational text. However, for presentation purposes, arbitrary units may be chosen.

For *result-sets* as defined in III.1.2 ii on page 13, the keywords must be followed by a *list of values* separated by commas, like in

```
IRNAM 'EPR Int.','Opt. Abs.'   * Labels for double
                               * detection
```

Axis labels and units must either be given for each member of a result-set, or the keywords must be omitted.

IV.3.2 SPL Keywords

The set of SPL keywords defines some kind of a common parameter language, which is used to express all physically meaningful acquisition parameters in a device independent format.

Some acquisition parameters can, at least in principle, be swept, while others can't. Since the BES^{3}T™ is designed to permit the description of both constant parameters and arbitrary sweep axes, it distinguishes between these two types. Table 9 on page 34 shows all currently defined SPL keywords which correspond to *non-sweepable* parameters. Note that continuous variability is not required for sweepable parameter, generally speaking, even an on/off-switch is 'sweepable'.

The description of arbitrary sweep axes is rendered possible by a, though limited, structure of the SPL language. To describe parameter sweeps, it uses the *name* of the swept parameter in conjunction with a generalized sweep given by 2 parameters out of *center value, sweep width, lower limit and upper limit* and the *resolution* parameter. Table 10 on page 35 lists SPL entries which are used to express such general sweep descriptions. Sweepable parameter names <SPN> are given in Table 11 on page 36. Note that all *sweep description values* must be given in the unit pertaining to the swept parameter.

A time-per-point value is only given for the 1st parameter because all other times per point are given by the time required to perform a complete sweep of the previous parameter.

Table 11 on page 36 lists all currently defined sweepable parameter names (SPNs) together with their SI units implied by the standard (given in [] brackets). This SI unit is supposed, if the unit field of the parameter entry format is omitted. A sweepable parameter may as well remain *fixed* during an experiment, in which case its SPN is used as *keyword* together with the *constant* parameter value.

Keyword	*Value*	*Meaning*
OPER	<text>	Operator name (up to 32 characters)
DATE	dd-mm-yy	Acquisition date
TIME	hh:mm.ss	Acquisition time
CMNT	<text>	Comment (up to 64 characters)
AVGS	<integer>	# of complete runs averaged (0 = single sweep acquisition)
RESO	<text>	Resonator name (up to 16 characters)
SAMP	<text>	Sample name (up to 64 characters)

Table 9: SPL entries for non-sweepable parameters

Keyword	*Value*	*Meaning*
SFOR	<text>	Sample formula (up to 32 characters)
STAG		State of aggregation
	C	Crystal
	P	Powder
	A	Amorphous
	L	Liquid
	G	Gaseous
DTRS	<bits>	Detector ADC resolution
SPTP	<time>	Single point sampling time [s]
EXPT	<EXPT>	Experiment type (see Table 11)
OXSP	<OXS>	Ordinate parameter. Argument values can either be a measurable SPN keyword (see Table 11) or one of the keywords listed in Table 11.

Table 9: SPL entries for non-sweepable parameters

Keyword	*Value*	*Meaning*
AXSn	<SPN>	Swept parameter n (see Table 11)
AnCT	anct	Sweep center value of parameter n
AnSW	answ	Sweep width of parameter n
AnLL	anll	Lower limit value of parameter n
AnUL	anul	Upper limit value of parameter n
AnRS	anrs	Resolution (number of points) of sweep n
AnTP	anpt	Time per point for parameter n
AnDT	andt	Dead time before sweep n

Table 10: SPL entries for general sweep descriptions

Note: If more than two keywords of **AnCT**, **AnSW**, **AnLL** and **AnUL** are specified, the first two entries accordings to the order in Table 10 are taken.

<SPN>-values required above are taken from the following table:

SPN	*Meaning*
NONE	Quantity not used
SANX	Sample angle around x [rad]
SANY	Sample angle around y [rad]
SANZ	Sample angle around z [rad]
STMP	Sample temperature [K]
SCON	Sample concentration [M]
ETIM	Time [s]
B0VL	Magnetic field [T]
B0EI	Magnetic field (EIE experiment) [T]
B0MA	Magnetic field modulation amplitude [T]
B0MF	Magnetic field modulation frequency [Hz]
MWFQ	Microwave frequency [Hz]
MWPW	Microwave power [W]
R1VL	Multiple Resonance frequency 1 [Hz]
R1FM	Multiple Res. frequency 1 FM depth [Hz]
R1AM	Multiple Res. frequency 1 AM depth [1]
R1MF	Multiple Res. frequency 1 FM/AM frequency [Hz]
R1AT	Multiple Res. frequency 1 attenuation [dB]
R2VL	Multiple Resonance frequency 2 [Hz]
R2FM	Multiple Res. frequency 2 FM depth [Hz]
R2AM	Multiple Res. frequency 2 AM depth [1]
R2MF	Multiple Res. frequency 2 modulation frequency [Hz]

Table 11: SPNs / keywords for sweepable parameters

SPN	*Meaning*
R2AT	Multiple Res. frequency 2 attenuation [dB]
RCAG	Receiver amplifier gain [1]
RCTC	Receiver time constant [s]
RCPH	Receiver phase [rad]
RCHM	Receiver harmonic [1]
RCOF	Receiver offset [Full Scale]
IMGA	Gradient direction α [rad]
IMGB	Gradient direction β [rad]
IMGG	Gradient strength in (α,β)-direction [T/m]
GRDX	Gradient strength in x-direction [T/m]
GRDY	Gradient strength in y-direction [T/m]
GRDZ	Gradient strength in z-direction [T/m]

Table 11: SPNs / keywords for sweepable parameters

<OXS>-values required above are taken from the following table:

OXS	*Meaning*
IADC	High resolution integrating ADC [1]
FADC	Flash digitizer [1]
TADC	Transient recorder [1]
MWDI	Diode current [A]
MWLO	Lock offset [Full Scale]

Table 12: OXSs / keywords for ordinate parameter

<EXPT>-values required above are taken from the following table:

TYPE	*Meaning*
CW	C.W. experiment
PLS	Pulse experiment
IMG	Imaging experiment
SIM	Simulation experiment
CAL	System calibration experiment

Table 13: EXPTs / keywords for experiment type

V. Appendix: Generating BES3T™ Datasets

V.1 Minimum Requirements

As mentioned in earlier, a BES3T™ dataset must consist of at least two files:

- The Data File, containing the data, and
- The Description File, containing at least the Descriptor with all entries declared mandatory in IV.3.1 on page 27.

The corresponding formats have been described in section IV on page 19. At this point, it shall just be emphasized that *only the Descriptor is required* in a Description File, everything else may be omitted, if mere compliance with the standard is required. Such a dataset still completely defines a publication-ready plot of the data, it only lacks the information required to reproduce the acquisition.

V.2 A.2 Example Datasets

This chapter shows a few examples of BES3T™ datasets by describing the contents of a Data File and listing the corresponding Description File as it would appear in a text editor.

V.2.1 A Simple EPR 1D-Spectrum

In this example, the Data File contains 1024 32-bit integer values in binary form. These values have been acquired by sweeping the magnetic field B_0 of an EPR spectrometer from 3350 G to 3360 G and digitizing the EPR absorption signal with a 9 bit resolution as detected by the standard 100kHz field modulation setup. The 4096 bytes constituting the spectrum information are arranged in big endian integer format.

The reader will notice that, although not required, keywords appear in the order they are discussed in the corresponding chapters. Such an arrangement is recommended in order to make Description Files more readable.

V.2.1 i) The Description File EXAMPLE1.DSC

```
#DESC 1.1 * DESCRIPTOR INFORMATION
************************
*
*  Dataset Type and Format:
*
DSRC EXP
BSEQ BIG        * Motorola
IKKF REAL
AX1TYP IDX
*
*  Item Formats :
*
IRFMT I
*
*  Data Ranges and Resolutions:
*
AX1MIN 335.0
AX1WID 1.0
AX1PTS 1024
*
*  Documentational Text:
*
TITL   'EPR-spectrum for BESSST demonstration'
IRNAM  'Abs'
AX1NAM 'Field'
IRUNI  'Arb.Un.'
AX1UNI 'mT'   * You need not use SI units here ...
*
#SPL 1.0 * STANDARD PARAMETER LAYER
************************
*
*  Operator, Date, and Time of Acquisition:
*
OPER 'fje'
DATE 17-06-91
```

```
TIME 13:44.45
CMNT 'the first run...'
*
*  # of sweeps averaged:
*
AVGS 1
*
*  Resonator and Sample description
*
RESO 'ER4102ST/TE102'
SAMP 'Irradiated Alanine (2 Grays, 1cm)'
SFOR 'Dunno'
STAG P
*
*  Analog-to-digital conversion:
*
DTRS 9
SPTP .02
*
*  Sweep axis description:
*
AXS1 B0VL      * Magnetic Field Sweep
A1CT 335.5e-3
A1SW 1.000e-3
A1RS 1024
*
* Fixed parameters:
*
STMP 297
B0MA .1e-3
B0MF 100e3
MWFQ 9.785e9
MWPW 10e-3
RCAG 500
RCTC 20e-3
RCPH 1.496
RCHM 1
```

```
*
#DSL 1.0 * DEVICE SPECIFIC LAYER
*************************
*
.DVC ESP_300,1.0   * Setup Parameters:
*
DT  BRU_SCH        * Acquisition with Signal Channel
B0  BRU_HALL       * Field controlled by Hall
B0M BRU_SCH        * Modulation by Signal Channel
MW  BRU_MBC        * Bridge controlled by MBC
*
.DVC BRU_SCH,1.0   * Signal Channel Parameters:
*
ROF 0
RRE 1              * Modulation to Cavity 1
*
.DVC BRU_HALL,1.0  * Hall Parameters:
*
HFO 0.0
HSD UP
*
.DVC BRU_MBC,1.0   * MBC Parameters:
*
MCO ON
*
************************************************
```

V.2.2 A 2D-Spectrum Recorded By Quadrature Detection

In this example, the Data File contains 2048 32-bit floating point pairs in binary form, representing the absorption and dispersion components of the EPR-signal. These values have been acquired while sweeping the magnetic field B_0 of an EPR spectrometer from 3300 G to 3400 G, for microwave powers of 1 to 10mW in 9 steps. Because the attenuator of a microwave bridge is calibrated in dB, the experimentator has decided to measure the actual power in mW and generate an y-index-gauge file containing the power values in double-precision format.

The example also shows the minimal requirement of a BES^{3}T^{TM} dataset in that the Description File only contains a Descriptor.

V.2.2 i) The Description File EXAMPLE2.DSC

```
#DESC 1.1 * DESCRIPTOR INFORMATION
**********************
*
*  Dataset Type and Format:
*
DSRC EXP
BSEQ BIG      * Motorola
IKKF CPLX
AX1TYP IDX
AX2TYP IGD
*
*  Item Formats :
*
IRFMT  F
IIFMT  F
AX1FMT D
*
*  Data Ranges and Resolutions:
*
AX1MIN .33
AX1WID .01
AX1PTS 2048
AX2MIN .001
AX2WID .009
AX3PTS 10
*
*  Documentational Text:
*
TITL  'Complex 2D-Dataset'
IRNAM 'Disp'
IINAM 'Abs'
AX1NAM 'Field'
```

```
AX2NAM 'MW-Power'
IRUNI  'Arb.Un.'
IIUNI  'Arb.Un.'
AX1UNI 'T'
AX2UNI 'W'
*
*  Minimum Dataset does not require more...
*
**************************************************
```

VI. Appendix B: Migration To The BES^{3}T™

VI.1 Conversion of Former BRUKER EPR Spectrum Format

Since the BES^{3}T™ is more or less different to former EPR spectrum storage formats, it is recommended to convert previously recorded spectra to the current format. Of course, the conversion is only required if one wants to use software packages that rely upon BES^{3}T™ data input. Together with BES^{3}T™-compatible software, BRUKER provides a conversion program which automatically produces BES^{3}T™ datasets out of the former BRUKER EPR spectrum format.

VI.2 Centralized Format Control For The DSL

The BES^{3}T™ concept depends on unambiguous Description File interpretation. Therefore, all DSL formats must be available upon request. In addition, unique Device IDs and keywords must ensure correct assignment. To support all this, BRUKER will offer the EPR community the *BES^{3}T™ format control service*. It will be the duty of this service to collect DSL dictionaries from all origins, provide copies upon request, and issue unique Device IDs for all manufacturers that adopt the BES^{3}T™ format.

VII. Appendix: Pathological Cases

This appendix is for the interested reader and covers more complicated or unexpected cases of data storage than the preceding sections. They are discussed by explaining the application of given BES^{3}T™ principles, where a case is completely specified. However, because the specifications of BES^{3}T™ had to maintain a certain brevity, there may also be ambiguous cases, whose interpretation would be proposed in the following.

The latter purpose of this Appendix particularly invites users comments, which would be printed here if their topic does not require an extension of the standard specifications, but may help others that might have similar interpretation problems.

VII.1 Mixture Of Complex Storage And Result Sets

If complex representations and result sets are mixed, then for Descriptor keywords beginning with **`II...`** a value entry must be given for *every* member of the result set, even if that member is **`REAL`** and would thus not require an **`II...`** entry. For example, if a result set is specified as

```
IKKF CPLX,REAL,CPLX   * Three acquired quantities
```

the **`IRFMT`** and **`IIFMT`** entries must look like

```
IRFMT I,D,I           * One value for each member...
IIFMT I,0,I          * ... and here, too!
```

with the **`0`** being mandatory. The **`IRNAM`** and **`IINAM`** entries must look like

```
IRNAM 'Real1','Real2','Real3' * One per member...
IIFMT 'Imag1','','Imag3'      * ... and here, too!
```

with the empty string **`''`** being mandatory.

BES^{3}T^{TM} SUGGESTION FORM

If you would like to make any suggestions or additions to the **BES^{3}T^{TM}**, please fill in this form and mail it to

BRUKER ANALYTIK GMBH
EPR-Software Department
Silberstreifen
76287 Rheinstetten
Germany

Author's Name .....

Company or OrganizationDept .....

Title Phone .....

Address .....

City Zip .....

State .....

My Suggestion .....

.....

.....

.....

.....

(add additional sheets as required)

The BRUKER EPR Software Group **BES3T_2.0**