Towards more reproducible and FAIRer research data: documenting provenance during data acquisition using the Infofile format
—Supporting Information—

Bernd Paulus[1] and Till Biskup[1, 2, *]

[1] *Physikalische Chemie, Albert-Ludwigs-Universität Freiburg, Albertstr. 21, 79104 Freiburg, Germany*
[2] *Present address: Bundesinstitut für Risikobewertung, Max-Dohrn-Straße 8–10, 10589 Berlin, Germany*

I. SPECIFICATION OF THE INFOFILE FORMAT

The Infofile format was originally developed to support the different MATLAB® toolboxes written by the authors to process and analyse trepr [1], cwepr [2], and TA [3] data, and this explains some of the restrictions of the format. The current reference implementation for a parser for the Infofile format is part of the ASpecD framework [4, 5], support for special formats (and mappings) is contained in derived packages, namely the trepr [6] and cwepr [7, 8] Python packages.

A. Format

The basic file format has historically been ASCII (7-bit). The restriction to the 7-bit ASCII character table ensures compatibility beyond operating system limits. This is only relevant due to MATLAB® not being able to cope with UTF-8 until recently. For the current reference implementation of the Infofile parser as part of the ASpecD framework [4, 5], this restriction does not apply any more, and any Infofile can make use of the full UTF-8 character set.

B. File name and extension

The file extension is '.info'. The file name is identical to the basic name of the associated data files. Note: Having identical file basenames is the ideal case, but not always possible to implement in practice. Therefore, this is not a hard rule, but a recommendation, albeit a strong one.

C. Format identifier

The first line of the file is reserved for an identifier. This enables the unique recognition of the file format during parsing. The identifier is separated from the rest of the file by a blank line and should contain a version number.

* E-mail: research@till-biskup.de

D. Blocks

The info file is divided into blocks. Blocks are introduced by the block name in capital letters. Each block is separated from the previous content of the file by a blank line. Within the blocks are key-value pairs consisting of a field identifier followed by a corresponding value. All block names should be in English for internationalisation.

E. Field identifiers (keys)

Field identifiers (keys) may contain spaces, but no special characters and no colon. The only exception at the moment are round brackets. If further exceptions are added, this could also be implemented. The reason for the restriction is that the field identifiers are/were originally used as field names internally in MATLAB® structures (struct) and hence need to conform to the naming scheme of variables. Field identifiers must start with a letter (no number!). Each field identifier is terminated by a colon. The block name should not be repeated in the field identifier. Example: 'Preparation' instead of 'Sample preparation' in the block 'SAMPLE'. This ensures shorter field names and at the same time greater clarity. All identifiers (keys) within the file should be in English for internationalisation.

F. Values

Values are always placed after a field identifier (key). Within a block, the values should all be indented so that they are flush with each other, *i.e.* the longest field identifier defines the indentation of the values. Values may contain special characters and colons. Depending on the environment, however, the use of special characters is not advisable, as they often fall victim to the different character set encoding. Note, however, that nowadays most operating systems support and use UTF-8 encoding. Hence, this should not be a problem any more, if not too old software and hardware is used. Values may extend over several lines. In this case, each new line must begin with a 'whitespace character' (space, tabulator, ...).

G. Use of colons

Colons are only used to separate field identifiers and values. In all other places (after an additional identifier, after a block heading) colons are forbidden. The only exception (see above) are field values. Colons are used internally during parsing to separate the corresponding field identifiers and values from each other.

H. Optional blocks and fields

Blocks and fields can be optional as long as the values of certain (obligatory?) fields ('switches') can be used to infer the (non-)existence of these fields and/or blocks. If there are no values for a field, but still good reasons not to remove the field, 'N/A' is set as the value. The latter may be the case for fields where, depending on the experiment, a relevant value should definitely be entered and only under special circumstances there is no value available. In such cases, it is strongly recommended to keep the field in all files, as otherwise it will be quickly forgotten.

I. Comments

Comments are always introduced with the percent sign (%). If a percent sign is to be used as such, it is protected ('escaped') by the preceding backslash (\\), according to the standard UNIX behaviour [9, p. 42]. Comments are possible both within a line and on their own. Comment blocks are also possible directly before blocks. In this case, the blank line must be moved to above the comment block in order to still clearly identify the block.

II. EXAMPLES

All examples shown as listings are available via GitHub as well [10], together with information on how to contribute to their further development. Due to the permissive license, everybody is welcome to use and further develop the Infofile format for own purposes. Development of the templates for specific methods will always be closely connected to the respective data model, *e.g.* in context of the trepr [6] and cwepr [7, 8] Python packages.

A. Time-resolved EPR spectroscopy

Listing 1. Example of an info file for time-resolved EPR spectroscopy. Usually, these setups are laboratory-build and consist of many different and exchangeable parts. Hence, proper documentation of each individual component is crucial.

```
trEPR Info file - v. 0.1.6 (2016-01-18)

GENERAL
Filename:                sample42
Date start:              2014-12-01
Date end:                2014-12-01
Time start:              10:00:00
Time end:                10:15:00
Operator:                John Doe
Label:                   #42 @ 120 K
Purpose:                 First overview

SAMPLE
Name:                    something
ID:                      42
Description:             frozen solution
Solvent:                 oDCB
Preparation:             1 mM, degassed
Tube:                    3.8 x 3 x 180 mm

EXPERIMENT
Runs:                    1
Shot Repetition Rate:    1 Hz

SPECTROMETER
Model:                   ESP380E
Software:                Transient, Vers. 0.6

MAGNETIC FIELD
Field probe type:        Hall
Field probe model:       xxx
Start:                   290 mT
Stop:                    390 mT
Step:                    0.4 mT
Sequence:                inward
Controller:              Bruker 032 T
Power supply:            Bruker 083 C

BACKGROUND
Field:                   300.0 mT
Occurrence:              10
Polarisation:            absorptive
Intensity:               8.8 mV

BRIDGE
Model:                   Bruker ER 046 MRT
Controller:              Bruker MBC
Attenuation:             20 dB
Power:                   2.01 mW
Detection:               mixer
Frequency counter:       HP 5352B
MW frequency:            9.6657 GHz

VIDEO AMPLIFIER
Bandwidth:               25 MHz
Amplification:           42 dB

RECORDER
Model:                   LeCroy 9354A
Averages:                1000
Time base:               2 ns
Bandwidth:               500 MHz
```

```
Pretrigger:              1.001 us
Coupling:                DC
Impedance:               50 Ohm
Sensitivity:             20 mV

TRANSIENT
Points:                  5000
Length:                  10 us
Trigger Position:        500

PROBEHEAD
Type:                    dielectric
Model:                   Bruker ER 4118X-MD5
Coupling:                critical

PUMP
Type:                    Laser
Model:                   GCR 190-10
Wavelength:              460 nm
Power:                   1 mJ
Repetition rate:         10 Hz
Tunable type:            OPO
Tunable model:           OPTA BBO VIS/IR
Tunable dye:             N/A
Tunable position:        11450
Filter:                  ND T=0.25

TEMPERATURE
Temperature:             120 K
Controller:              Oxford ITC 503
Cryostat:                Oxford ESR935
Cryogen:                 LN2

FIELD CALIBRATION
Filename:                N/A
Field probe type:        Gaussmeter
Field probe model:       xxx
Standard:                LiLiF
Signal field:            xx G
MW Frequency:            xx GHz

COMMENT
Deviation GM-HP of about 1.1 G.
After 480 transients, coupling and field
had been drifted. Readjustment at 3499.2 G.
```

B. continuous-wave EPR spectroscopy

Listing 2. Example for an info file for continuous-wave EPR spectroscopy. While these measurements are usually performed on commercial spectrometers, still, most EPR spectrometers except of the benchtop models consist of many different and exchangeable parts. Hence, proper documentation of each individual component is crucial.

```
cwEPR Info file - v. 0.1.3 (2016-01-18)

GENERAL
Filename:                sample42
Date start:              2019-02-27
Date end:                2019-02-27
Time start:              07:22:00
Time end:                07:26:00
Operator:                John Doe
Label:                   sample42-m01
Purpose:                 orientation dependence

SAMPLE
Name:                    BTZ
ID:                      42
Description:             Lorem ipsum
Solvent:                 CHCl3
Preparation:             20 ul drop-cast
Tube:                    3.8 x 3.0 x 250 mm

EXPERIMENT
Type:                    field-sweep
Runs:                    1
Variable parameter:      field
Increment:               xx mT

SPECTROMETER
Model:                   Bruker Elexsys E580
Software:                Xepr 2.6b.146

MAGNETIC FIELD
Field probe type:        Hall
Field probe model:       xxx
Start:                   346.5 mT
Stop:                    356.5 mT
Step:                    10/1023 mT
Sequence:                up
Controller:              Bruker
Power supply:            Bruker

BRIDGE
Model:                   Bruker Super-X FT-EPR
Controller:              Bruker
Attenuation:             30 dB
Power:                   150 uW
Detection:               diode
Frequency counter:       Bruker
MW frequency:            9.83894 GHz
Q value:                 N/A

SIGNAL CHANNEL
Model:                   Bruker EPR SPU
Modulation amplifier:    Bruker
Accumulations:           4
Modulation frequency:    100 kHz
Modulation amplitude:    0.05 mT
Receiver gain:           60 dB
Conversion time:         60 ms
Time constant:           N/A
Phase:                   0 deg

PROBEHEAD
Type:                    HQ
Model:                   Bruker 4119HS-W1
Coupling:                critical

TEMPERATURE
Temperature:             295 K
Controller:              N/A
Cryostat:                N/A
Cryogen:                 N/A

COMMENT
5 min irradiation with UV light
```

C. Transient absorption spectroscopy

Listing 3. Example for an info file for optical transient-absorption spectroscopy. In this case, a commercial setup has been used. Still. it is assembled from different exchangeable parts that each need to be documented in sufficient detail.

```
TA Info file - v. 0.2d (2012-03-31)

GENERAL
Filename:                 sa21
Date:                     2017-01_23
Time start:               14:30:00
Time end:                 15:40:00
Operator:                 John Doe
Label:                    sa369, oDCB, 10us
Spectrometer:             LP920-K
Software:                 L900, Version 7.3.6
Runs:                     1
ShotRepetitionRate:       0.2 Hz
Purpose:                  First try

SAMPLE
Name:                     Something
Description:              Half the truth
Preparation:              0.1 mM in oDCB
Cuvette:

TRANSIENT
Points:                   2000
Trigger position:         200
Length:                   10 us

SPECTROGRAPH
Type:                     Czerny-Turner with
    ➥ Triple Grating Turret
Model:                    standard
Aperture front:
Aperture back:

DETECTION
Type:                     PMT
Model:                    standard
Power supply:             standard
Impedance:                50 Ohm
Time constant:

RECORDER
Model:                    Tektronix TDS 3012C
Averages:                 1
Sensitivity:              10 mVOhm
Bandwidth:                1.0
Time base:                5 ns
Coupling:

PUMP
Type:                     Laser
Model:                    Continuum Surelite 1
Wavelength:               454 nm
Power:                    6.5 mJ
Repetition rate:          10 Hz
Tunable type:             OPO
Tunable model:            Continuum OPO Plus

PROBE
Type:                     Lamp
Model:                    standard
Wavelength start:         300 nm
Wavelength stop:          800 nm
Wavelength step:          4 nm
Wavelength sequence:      up
Power:
Filter:
Background:               lamp

TEMPERATURE
Temperature:              RT
Controller:               Lauda alpha RA8
Cryostat:                 N/A
Cryogen:                  H2O

COMMENT
Unfortunately, no usable signal
```

III. ADDITIONAL DISCUSSION

A. Precedence of parameter values

One direction to further develop the Infofile format discussed in the outlook is to minimise the contents of the Infofile that need to be entered manually by the operator. This could be achieved by files containing only those parameters *not* collected automatically by the setup, and adding all the other parameters afterwards automatically by the software used to process and analyse the data. Beware, however, that the 'truth' is not necessarily always in the parameter values collected by the setup. From own experience, there is spectrometer control software that does not automatically save the recorded data after the measurement finished, but only after explicit user interaction. However, the parameter values saved to the file are those read from the hardware when actually asked to save the data. Any changes to these parameters from the values during data acquisition, be it drifts or accidential setting different values by the user, *e.g.*, as preparation for the next measurement, will be saved to the file in this case. The result: The parameter values stored in the data file are not the correct ones. On the other hand, as the Infofile contents are written entirely by hand, values need not necessarily be correct as well, be it due to copy and paste or typos. Eventually, only manual inspection will help here—or changing the spectrometer control software.

B. Comparing the Infofile format to JSON and YAML

It has been stated that the Infofile format is both, easier to write by humans and more robust. The latter is mostly due to the relative simplicity of the Infofile format and the rather forgiving handling of whitespace character. While JSON relies on brackets, YAML does so on whitespace. For programmers or technically skilled people, using YAML or even JSON is not much of a hurdle. However, the intended users of the Infofile format are those people in the lab that are less technically skilled. Therefore, when designing the Infofile format, human writability has been valued higher than simplicity of parsing.

For a direct comparison of the three file formats, the same content of the most generic version of an Infofile given in the main text is provided in the different formats and shown in the Listings 4, 5, and 6. This does not imply that either JSON or YAML are not useful file formats. Both are incredibly useful and are supported by many different programming languages out of the box or by using robust and well-proven libraries, and for both, not only parsers but validators are available, making checking a given file for syntactical correctness pretty easy.

Listing 4. Most generic version of an Infofile without any specific details for a method, presented in JSON format. Note the obligatory use of brackets for the different 'objects'. JSON does not rely on whitespace and can be minimised, removing any whitespace characters outside actual keys or values, for saving bandwidth during transport.

```
{
   "format": {
      "type": "common metadata",
      "version": "0.1.0"
   },
   "general": {
      "start": {
         "date": "2020-04-04",
         "time": "11:05:00"
      },
      "end": {
         "date": "2020-04-04",
         "time": "15:50:00"
      },
      "operator": "John Doe",
      "purpose": "Kill time"
   },
   "sample": {
      "name": "Random sample 1",
      "description": "Nicked from bench
         ➥ neighbour"
   },
   "comment": "To be or not to be\n"
}
```

Listing 5. Most generic version of an Infofile without any specific details for a method, presented in YAML format. Note that YAML relies on the consistent indentation, but therefore does not require any brackets. This makes this format much easier to write for humans than JSON (cf. Listing 4).

```
---
format:
  type: common metadata
  version: 0.1.0

general:
  start:
    date: 2020-04-04
    time: 11:05:00
  end:
    date: 2020-04-04
    time: 15:50:00
  operator: John Doe
  purpose: Kill time

sample:
  name: Random sample 1
  description: Nicked from bench neighbour

comment: >
  To be or not to be...
```

Listing 6. Most generic version of an Infofile without any specific details for a method, presented in the actual Infofile format. As mentioned, the Infofile format is the least verbose of the three, and in contrast to YAML, it is quite forgiving regarding the handling of whitespace characters.

```
common Info file - v. 0.1.0

GENERAL
Date start:  2020-04-04
Time start:  11:05:00
Date end:    2020-04-04
Time end:    15:50:00
Operator:    John Doe
Purpose:     Kill time

SAMPLE
Name:        Random sample 1
Description: Nicked from bench neighbour

COMMENT
To be or not to be...
```

[1] Biskup, T.; Paulus, B.; Meyer, D. trEPR toolbox. 2022; doi:10.5281/zenodo.7395548.

[2] Biskup, T.; Meyer, D. cwEPR toolbox. 2022; doi:10.5281/zenodo.7396037.

[3] Biskup, T. TA toolbox. 2022; doi:10.5281/zenodo.7395925.

[4] Biskup, T. ASpecD framework. 2022; `https://docs.aspecd.de/`, doi:10.5281/zenodo.4717937.

[5] Popp, J.; Biskup, T. ASpecD: A modular framework for the analysis of spectroscopic data focussing on reproducibility and good scientific practice. *Chem. Methods* **2022**, *2*, e202100097.

[6] Popp, J.; Schröder, M.; Biskup, T. trEPR Python package. 2021; `https://docs.trepr.de/`, doi:10.5281/zenodo.4897112.

[7] Schröder, M.; Biskup, T. cwepr Python package. 2021; `https://docs.cwepr.de/`, doi:10.5281/zenodo.4896687.

[8] Schröder, M.; Biskup, T. cwepr - A Python package for analysing cw-EPR data focussing on reproducibility and simple usage. *J. Magn. Reson.* **2022**, *335*, 107140.

[9] Raymond, E. S. *The Art of UNIX Programming*; Addison Wesley: Boston, 2004.

[10] Paulus, B.; Biskup, T. Infofile. 2022; `https://github.com/tillbiskup/infofile`, doi:10.5281/zenodo.7452780.