Probably the Last Word in 1D SAS Data Formats

R.E. Ghosh, S.M. King & A.R. Rennie

ILL, 6, rue Jules Horowitz, BP 156, 38042 Grenoble Cedex 9, France

ISIS, STFC Rutherford Appleton Laboratory, Harwell Science & Innovation Campus, Didcot, OX11 0QX, UK

Dept of Physics, Box 530, SE-751 21 Uppsala, Sweden

ghosh@ill.fr ; s.m.king@rl.ac.uk ; adrian.rennie@fysik.uu.se

Technology can now federate the innate tendencies of each SAS facility to develop its own data and software standards. Writing data in simple XML[1] format allows standard tools to interconvert from one structure to another, with appropriate changes in tag names. Increasing use of XML files in many aspects of GUI control and system layout ensures there will be continued tool development in the future. For the SAS community the XML file offers a man-readable and machine-treatable data form. Further extension to include data measured with a range of sample conditions in a single file comes as an obvious extension to the structured layout.

The sasCIF format specification of Svergun & Malfois[2] established a basic dictionary of terms, and provides a formal basis for archiving data, though it lacks structure for easy data extraction or manipulation. The NeXus raw data project evolved at much the same time, having an implicit structure from its roots in the Hierarchical Data File of the NCSA. The nomenclatures from the two approaches were compared[3]. We adopt a similar style to NeXus, though simplify presentation and parsing by proposing the clear inclusion of the units in the name of the metadata component rather than as a separate attribute to be parsed. Providing a four column table of scattering vector, intensity, and the experimental deviations of each, allows many instrumental parameters to be (optionally) omitted.

Many occasional users of the SAS technique will appreciate the ability to drop XML data into **Excel2003** (or later) and find the data automatically entered into the spreadsheet with labelled columns. For many packages, Java, Python, etc, there are library routines to parse XML. For traditionalists using Fortran, a C++ wrapper has been developed[4] for use with Frank van den Berghen's simple XML parser[5]. These routines can be easily adapted to other languages. The use is illustrated in a simple file converter for existing standard ILL data, using a dictionary of multiple aliases. Extension to parse local sections of multi-measurement files is straightforward.

References

1. http://www.xml.org/xml/resources_focus_beginnerguide.html

2. http://www.embl-hamburg.de/ExternalInfo/Research/Sax/reprints/sascif_2000.pdf

3. ftp://ftp.ill.fr/pub/cs/sans/can3king.pdf

4. http://www.ill.fr/lss/canSAS/forxml.htm

5. http://www.applied-mathematics.net/tools/xmlParser.html