THE CANSAS XML DATA FORMAT ('SASXML')

SMK, 08/04/08

Introduction

Although this document will specifically concern itself with treated/reduced SANS data from ISIS, the format that will be described is generic. Only the implementation of the format is actually specific to LOQ or SANS2D. Data in comparable implementations is now also available from the ILL (Grenoble) and HFIR (Oakridge) SANS instruments, I22 at Diamond, and the USAXS beam lines at the APS (Chicago), and other facilities will adopt the format in the near future, thus facilitating data interchange and empowering data treatment.

The new format, colloquially called SASXML, has been conceived by a consortium of interested small-angle beam line scientists and their software engineers from around the world and 'wraps' the data in XML, eXtended Markup Language[1]. XML looks similar to HTML and indeed can be read by web browsers. However XML offers different functionality to HTML. It has been adopted because it is fast becoming the *de facto* industry standard for data that is both human and machine-readable, and it is supported by the likes of Microsoft.

Programs like Microsoft Excel, and other third-party software, increasingly read XML data files. This provides a convenient route to data manipulation and display without the need for specialised programming as was often the case in the past.

This document uses Excel as an example.

Support Issues

You must have Microsoft Excel 2002, or a more recent release, loaded as only these have the necessary XML support.

Despite this, there is an Add-In of XML Tools for Excel that can be downloaded from the Microsoft Developers Network (MSDN). For convenience it is also available on the ISIS Large-Scale Structures website at:
http://www.isis.rl.ac.uk/largescale/loq/programs/OfficeExcel2003XMLToolsAddin.exe

Install it by double-clicking on the installer and following the instructions, then open up a blank worksheet, click on "Tools" → "Add-ins" → "Browse", navigate to the "XmlTools.xla" file, select it, click "Ok", and ensure that "xmltools" is checked in the Add-ins list.

Getting the Stylesheet

XML has the ability to tell a program how it should interpret the data in a file. It is not absolutely necessary to make use of this feature, called a stylesheet, and a good program will always have a default means of dealing with an XML file, but using a stylesheet is highly recommended.

You can download the SASXML-specific stylesheet at:
http://svn.smallangles.net/svn/canSAS/1dwg/trunk/

(NB: This URL also appears at the top of any SASXML file, on the line beginning "xsi:" – though it *intentionally* points to a different file from the one you need!).

Right-click on the file example.xsl and save it to the directory containing your SASXML files.

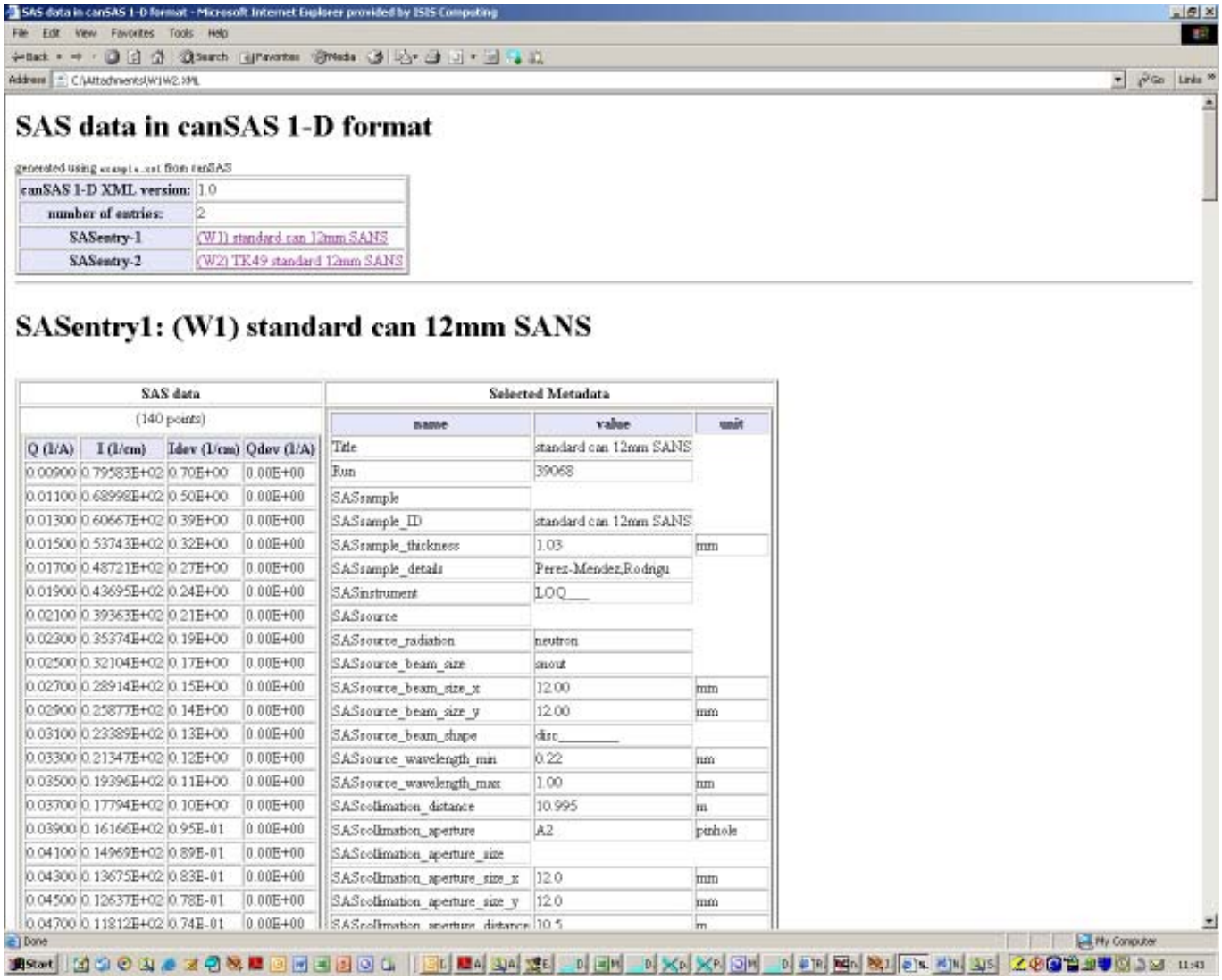
If you do not use, or have, the stylesheet, go to Appendix A of this document.

For More Information

For more detailed information on SASXML, please visit:
http://www.smallangles.net/wgwiki/index.php/cansas1d_documentation

Viewing the Data

If the SASXML files and stylesheet (example.xsl) are in the same directory, simply double-click on a data file. The default web browser will open, load the data file and apply the stylesheet:



SAS data in canSAS 1-D format - Microsoft Internet Explorer provided by ISIS Computing

File Edit View Favorites Tools Help

Address C:\Attachments\W1W2.XML

SAS data in canSAS 1-D format

generated using example.xsl from canSAS

canSAS 1-D XML version:	1.0
number of entries:	2
SASentry-1	(W1) standard can 12mm SANS
SASentry-2	(W2) TK49 standard 12mm SANS

SASentry1: (W1) standard can 12mm SANS

SAS data (140 points)

Q (1/A)	I (1/cm)	Idev (1/cm)	Qdev (1/A)
0.00900	0.79583E+02	0.70E+00	0.00E+00
0.01100	0.68998E+02	0.50E+00	0.00E+00
0.01300	0.60667E+02	0.39E+00	0.00E+00
0.01500	0.53743E+02	0.32E+00	0.00E+00
0.01700	0.48721E+02	0.27E+00	0.00E+00
0.01900	0.43695E+02	0.24E+00	0.00E+00
0.02100	0.39363E+02	0.21E+00	0.00E+00
0.02300	0.35374E+02	0.19E+00	0.00E+00
0.02500	0.32104E+02	0.17E+00	0.00E+00
0.02700	0.28914E+02	0.15E+00	0.00E+00
0.02900	0.25877E+02	0.14E+00	0.00E+00
0.03100	0.23389E+02	0.13E+00	0.00E+00
0.03300	0.21347E+02	0.12E+00	0.00E+00
0.03500	0.19396E+02	0.11E+00	0.00E+00
0.03700	0.17794E+02	0.10E+00	0.00E+00
0.03900	0.16166E+02	0.95E-01	0.00E+00
0.04100	0.14969E+02	0.89E-01	0.00E+00
0.04300	0.13675E+02	0.83E-01	0.00E+00
0.04500	0.12637E+02	0.78E-01	0.00E+00
0.04700	0.11812E+02	0.74E-01	0.00E+00

Selected Metadata

name	value	unit
Title	standard can 12mm SANS	
Run	39068	
SASsample		
SASsample_ID	standard can 12mm SANS	
SASsample_thickness	1.03	mm
SASsample_details	Perez-Mendez,Rodrigu	
SASinstrument	LOQ___	
SASsource		
SASsource_radiation	neutron	
SASsource_beam_size	snout	
SASsource_beam_size_x	12.00	mm
SASsource_beam_size_y	12.00	mm
SASsource_beam_shape	disc_____	
SASsource_wavelength_min	0.22	nm
SASsource_wavelength_max	1.00	nm
SAScollimation_distance	10.995	m
SAScollimation_aperture	A2	pinhole
SAScollimation_aperture_size		
SAScollimation_aperture_size_x	12.0	mm
SAScollimation_aperture_size_y	12.0	mm
SAScollimation_aperture_distance	10.5	m

Done

My Computer

Start

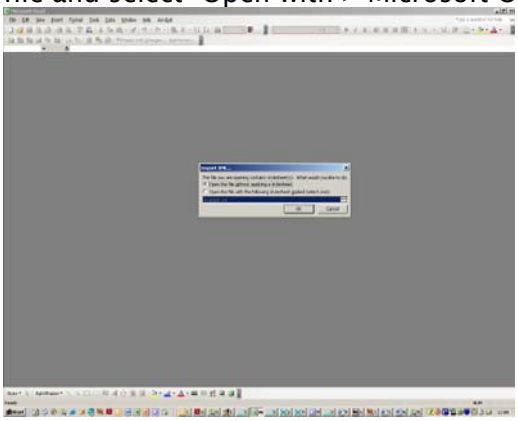
Notice that in this example, the data file actually contains *two* different data entries: SASentry-1 & SASentry-2, written from COLETTE workspaces W1 & W2 respectively.

The bookmark links in the 'contents' box at the top can be used to jump to the start of each entry.

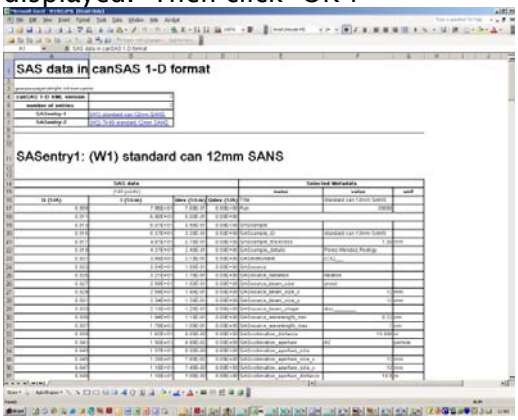
Each entry consists of two tables. The left-hand contains the data values, whilst the right-hand contains the metadata – in this case information about the sample and instrument setup.

Importing the Data

1. If the SASXML files and stylesheet (example.xsl) are in the same directory, right-click on a data file and select "Open with > Microsoft Office Excel". An 'Import XML…' dialog box will appear:

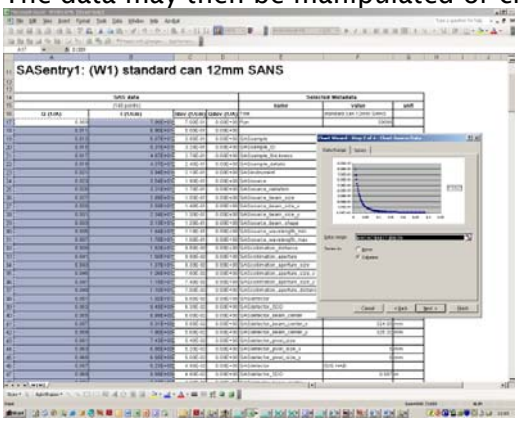


2. Select "Open the file with the following stylesheet applied" and ensure that 'example.xsl' is displayed. Then click "OK":



Notice the similarity between how the data is presented here in Excel and in Internet Explorer above (see 'Viewing the Data').

3. The data may then be manipulated or charted as normal:



Example SAS XML data files are available at:

http://www.isis.rl.ac.uk/largescale/loq/xml/isis_sasxml_example.xml

http://www.isis.rl.ac.uk/largescale/loq/xml/ill_sasxml_example.xml

This document is available at:

http://www.isis.rl.ac.uk/largescale/loq/xml/cansas_xml_format.pdf

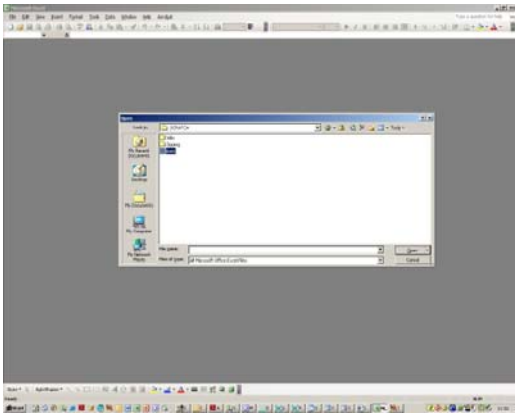
APPENDIX A

Importing XML Data without a Stylesheet

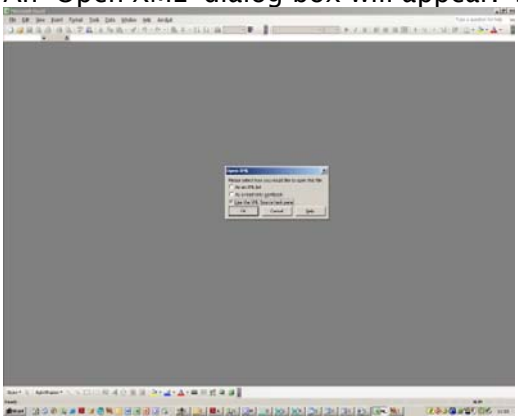
There are two routes of importing XML data into Excel without a stylesheet. The first utilises XML in the manner for which it was intended – recognising structured metadata carried alongside the data. The second route merely ignores the structure.

1st Method

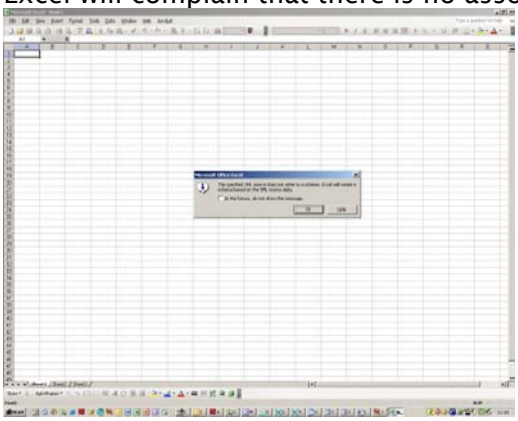
1. Launch Excel and select a data file:



2. An 'Open XML' dialog box will appear. Select the XML Source Task pane and click 'OK':

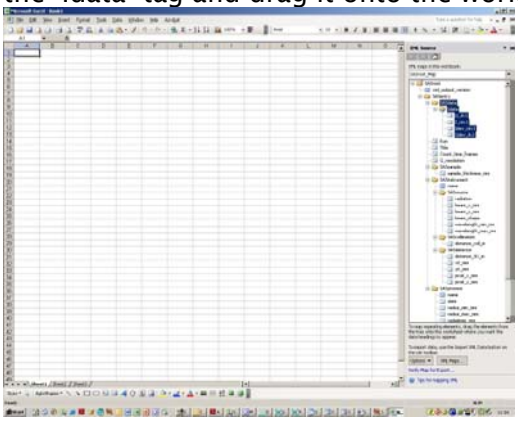


3. Excel will complain that there is no associated Schema. Just click 'OK':

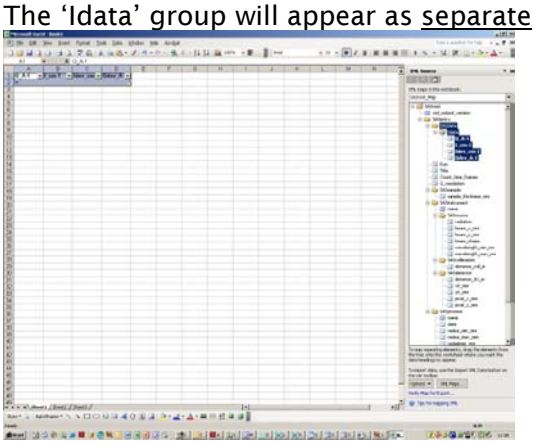


(Excel proceeds to generate it's own 'best guess' schema on-the-fly).

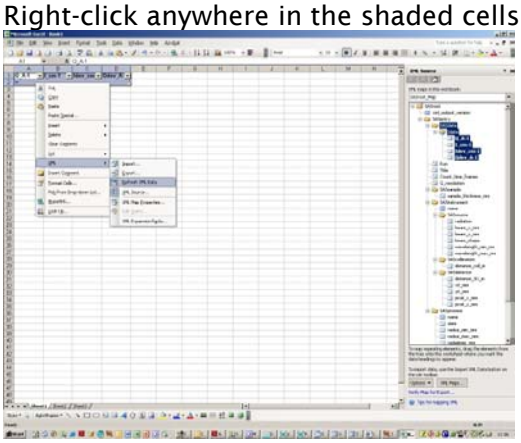
4. A ‘directory’ of the file contents will appear on the right-hand side of the screen. Any of the items may be highlighted and dragged into a cell on the worksheet. To import the data, select the ‘Idata’ tag and drag it onto the worksheet:



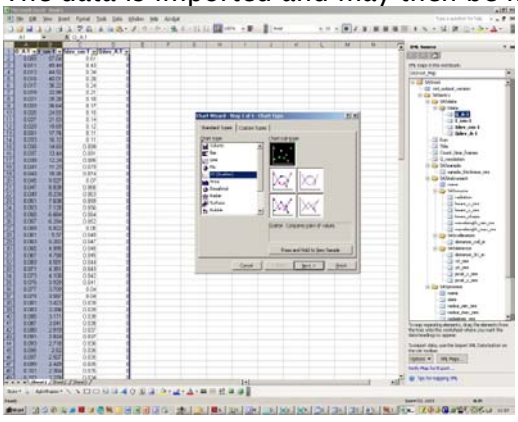
5. The ‘Idata’ group will appear as separate columns of data in the worksheet:



6. Right-click anywhere in the shaded cells and select ‘Refresh XML data’:

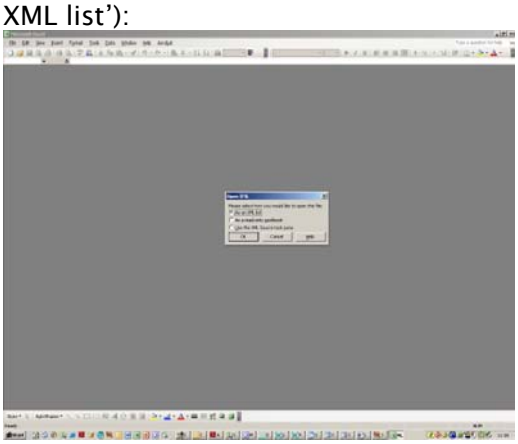


7. The data is imported and may then be manipulated or charted as normal:

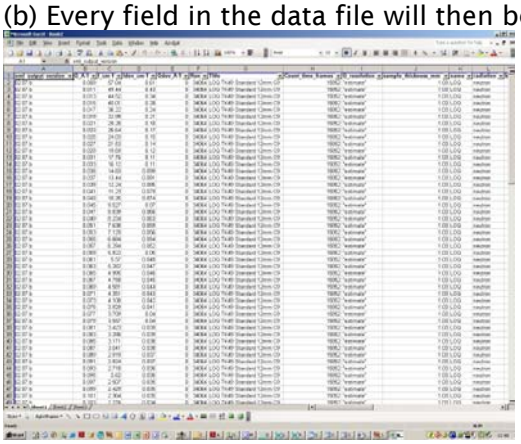


2nd Method

2. (b) When the 'Open XML' dialog appears, just click 'OK' to allow the default selection ('As an XML list'):



4. (b) Every field in the data file will then be imported into the worksheet:



3rd Method

A 'method of last resort' is to rename the XML files to .TXT files and to use the Excel Text Import Wizard (from following "File" → "Open").

Select: 'Delimited' & 'Start import at Row 11' → "Next" → 'Delimiters: Space' → "Next" → "Finish".

This actually illustrates one of the benefits of XML; namely that because it is ACSII text, accessing the data is not wholly dependent on a specialist or XML-compliant program.

APPENDIX B

Data Structure

The format requires the datafile to adhere to a particular structure. The order of this structure is not prescriptive, but those developing the format decided to place the data before the metadata. However, provided the structure is sound, the datafile can contain as much, or as little, information as required. Files do not need to be a fixed length.

The file structure is:

<?xml version="1.0"?> <?xml-stylesheet type="text/xsl" href="example.xsl" ?> <SASroot version="1.0" xmlns="cansas1d/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="cansas1d/1.0 http://svn.smallangles.net/svn/canSAS/1dwg/trunk/cansas1d.xsd"> <SASentry name="W1">	**Essential XML Content!**
<Title> standard can 12mm SANS </Title> <Run> 39068 </Run>	**Mandatory!**
<SASdata> <Idata><Q unit="1/A"> 0.00900 </Q><I unit="1/cm"> 0.79583E+02 </I><Idev unit="1/cm"> 0.70E+00 </Idev><Qdev unit="1/A"> 0.00E+00 </Qdev></Idata> . . et seq . <Idata><Q unit="1/A"> 0.28700 </Q><I unit="1/cm"> 0.36962E+00 </I><Idev unit="1/cm"> 0.18E+01 </Idev><Qdev unit="1/A"> 0.00E+00 </Qdev></Idata> </SASdata>	**The Data:** - is always present - note the 4 columns and their order! - lines must be bounded by <Idata> & </Idata> - column and unit names are prescribed by canSAS!
<SASsample> <ID> standard can 12mm SANS </ID> <thickness unit="mm"> 1.03 </thickness> <details> Perez-Mendez,Rodrigu </details> </SASsample> <SASinstrument> <name> LOQ __ </name> <SASsource> <radiation> neutron </radiation> <beam_size name="snout"> <x unit="mm"> 12.00 </x> <y unit="mm"> 12.00 </y> </beam size> <beam shape> disc </beam shape> <wavelength min unit="nm"> 0.22 </wavelength min> <wavelength_max unit="nm"> 1.00 </wavelength_max> </SASsource> <SAScollimation name="fixed"> <distance unit="m"> 10.995 </distance> <aperture name="A2" type="pinhole"> <size> <x unit="mm"> 12.0 </x> <y unit="mm"> 12.0 </y> </size>	**The Metadata:** - may be wholly or partly absent - may contain more information than that shown left (this allows for different facilities & instruments to tailor the content of the file to their own needs) - only tag names prefixed by 'SAS' should be considered prescribed names (they map to fields in the NeXus file format) - the meaning of tag names should be 'obvious' to the reader

```
    <distance unit="m"> 10.5 </distance>
   </aperture>
  </SAScollimation>
  <SASdetector>
   <name> ORDELA 2661N </name>
   <SDD unit="m"> 4.155 </SDD>
   <beam_center>
    <x unit="mm"> 324.03 </x>
    <y unit="mm"> 325.32 </y>
   </beam center>
   <pixel size>
    <x unit="mm"> 5.00 </x>
    <y unit="mm"> 5.00 </y>
   </pixel_size>
  </SASdetector>
  <SASdetector>
   <name> ISIS HAB </name>
   <SDD unit="m"> 0.587 </SDD>
   <beam_center>
    <x unit="mm"> 324.03 </x>
    <y unit="mm"> 325.32 </y>
   </beam center>
   <pixel size>
    <x unit="mm"> 12.00 </x>
    <y unit="mm"> 12.00 </y>
   </pixel_size>
  </SASdetector>
 </SASinstrument>
 <SASprocess>
  <name> COLETTE </name>
  <date> 09-MAR-2008 10:22:32 </date>
  <term name="mask_file"> USER:MASK.073K                                    </term>
  <term name="scale factor" unit="a.u."> 1.5260 </term>
  <term name="radius min" unit="mm">  38.0 </term>
  <term name="radius max" unit="mm"> 419.0 </term>
  <term name="radialstep" unit="mm">   3.0 </term>
  <term name="sector_width" unit="degree"> 180.0 </term>
  <term name="count time" unit="frame">      21610 </term>
  <SASprocessnote name="q resolution"> estimate </SASprocessnote>
  <SASprocessnote name="data mode"> pointdata </SASprocessnote>
  <SASprocessnote name="file_written">  8-APR-2008 09:18:19.93 </SASprocessnote>
 </SASprocess>
```

- <SASprocess> can be used to store comments about the data or the data treatment process

```
 </SASentry>
 </SASroot>
```

Mandatory!

The first line declares the file to be an XML file. The second line tells an XML-compliant program the name and location of the usage-specific stylesheet, whilst the next four lines tell an XML-compliant program the name and location of the 'XML Schema' that initialises the 'directory tree'-like structure of the file.

The 'SASentry' declaration creates a dataset (data+metadata, if metadata is present) within the file. There is no limit to the number of datasets in a file provided each is bounded by <SASentry> and </SASentry> tags.

The </SASroot> tag signals the end of the structure (ie, the end of the file).

APPENDIX C

Programs other than Excel

1. **Igor Pro**
   A set of Igor Pro routines have been developed to read SASXML data files.

   More information can be found at:
   http://www.smallangles.net/wgwiki/index.php/IGOR_Pro_Developers_Working_Group

   Alternatively contact Andrew Nelson (ANSTO): andyfaff@gmail.com

2. **Origin**

   Support
   You must have Origin or OriginPro 7.5 or a more recent release.

   There is no "native" XML support in Origin (as far as can be seen), however support can be added by way of Origin C scripts. An example of what one of these may look like can be found in the \OriginLab\OriginPro75\Samples\COM Client\WKSToXML.c sample file included in the program installation.

   Making a Script Available to a Workspace
   To make an Origin C script available to Origin workspaces it must be compiled and linked into the main program:

   Start Origin. Then select "View" → "Code Builder" (Alt+4).

   In the workspace 'tree' in the left-hand pane, right-click on "System", select "Add Files…" and navigate to the location of the Origin C script. Select the required file and click on "Open". The file will be added to the 'tree' under "System".

   Right-click on the newly-added script in the 'tree' and highlight "Compile" (Ctrl-F8).

   Now relink the whole program with "Tools" → "Rebuild All" (Alt-F8). Then choose "File" → "Exit" to return to Origin.

   Executing a Script in Origin
   Select "Window" → "Script Window" (Alt-F3).

   Enter a command line.

   For example, to use the sample file above to read in data from an XML file (assuming it conformed to the expected data structure) the syntax would be:

   XMLToWks drive:\directory\file.xml;

   Or, to write the contents of an Origin workspace as an XML file in the format described by the sample file the syntax would be:

   WksToXML Data1 drive:\directory\file.xml;

   Where Data1 is the name of the workspace in Origin.

[1] http://www.xml.org/