cansas1d binding IgorPro

From canSAS

Contents

canSAS 1-D SAS XML format binding to IgorPro

An import tool (binding) for IgorPro has been created (cansasXML.ipf). You can check out the IgorPro working directory from the SVN server (see instructions below).

To use the canSASxml.ipf procedure, you must have the XMLutils XOP IGOR plugin installed. See the Usage Notes below.

Note that the code described here is *not a complete user interface*. (See further comments below.) It is expected that this code will be called by a graphical user interface routine and that routine will handle the work of copying the loaded SAS data in IgorPro from the root:Packages:CS_XMLreader data folder to the destination of choice (including any renaming of waves as desired).

file	cansasXML.ipf (http://svn.smallangles.net/trac/canSAS/browser/1dwg/trunk/IgorPro /cansasXML.ipf)
author	Pete R. Jemian <jemian@anl.gov>
date	2008-09-02
version	1.10 (**requires** latest XMLutils XOP -- see below)

purpose	implement an IgorPro file reader to read the canSAS 1-D reduced SAS data in XML files adheres to the cansas1d/1.0 standard: http://www.smallangles.net/wgwiki/index.php/cansas1d_documentation
URL	TRAC: http://svn.smallangles.net/trac/canSAS/browser/1dwg/trunk/IgorPro/cansasXML.ipf SVN: http://svn.smallangles.net/svn/canSAS/1dwg/trunk/IgorPro/cansasXML.ipf
requires	IgorPro: http://www.wavemetrics.com XMLutils - XOP: http://www.igorexchange.com/project/XMLutils (IGOR.5.04.x-1.x-dev, 2008-Aug-22)

Checkout of support code in Subversion

Subversion (http://subversion.tigris.org/) is a program for managing software versions. There are command line and GUI clients for a variety of operating systems. We won't recommend any here but will show the command lines necessary.

XMLutils XOP

The XMLutils XOP, written by Andrew Nelson (ANSTO), is hosted on the IgorExchange (http://www.igorexchange.com/).

One good location to place the checked out XMLutils directory is in the Wavemetrics directory, next to the Igor Pro Folder

```
svn://svn.igorexchange.com/packages/XMLutils/ XMLutils
```

In the future, to retrieve an updated version of this support, go into the XMLutils directory (created above) and type the command

```
svn update
```

This will check the repository and update files as needed. If the installer program was updated, you;ll need to run the new installer program. It is not necessary to uninstall first.

The installer executables contained in the download will do all the installation for you. They will place the XOP in the folder */User Procedures/motofit/XMLutils*, and create a shortcut/alias to the plugin in /Igor Extensions. Packages from other facilities should place the XOP there as well.

cansasXML.ipf

Check out the canSAS 1d SAS XML reader from the subversion repository:

```
svn checkout http://svn.smallangles.net/svn/canSAS/1dwg/trunk cansas-1dwg
```

This will download lots of extra files. The file of interest is in the IgorPro directory and is called cansasXML.ipf

In the future, to retrieve an updated version of this support, go into the cansas-1dwg directory (created above) and type the command

```
svn update
```

This will check the repository and update files as needed.

Installation

1. License and Install IgorPro (should have already been done by now)
2. Quit IgorPro if it is running
3. Download XMLutils XOP. Either checkout from subversion (see above) or, with a web browser, go to http://svn.igorexchange.com/viewvc/packages/XMLutils/trunk/
4. Install XMLutils XOP by double-clicking the installer for you operating system.
5. Download cansasXML.ipf. Either checkout from subversion (see above) or, with a web browser, copy cansasXML.ipf from on-line subversion repository: http://svn.smallangles.net/svn/canSAS/1dwg/trunk/IgorPro/cansasXML.ipf
6. Copy cansasXML.ipf file to ...\Wavemetrics\Igor Pro Folder\User Procedures (or file system equivalent)
7. Then, you should be able to restart IgorPro and progress from there

Usage Notes

To use the canSASxml.ipf procedure, you must have the XMLutils XOP IGOR plugin installed. This may be downloaded from the IgorExchange Project site. There are installer executables contained in the download that will do all the installation for you. Each installer will place the XOP in the folder ...\Wavemetrics\Igor Pro Folder\User Procedures\motofit\XMLutils, and create a shortcut/alias to the plugin in ...\Wavemetrics\Igor Pro Folder\Igor Extensions.

What it does

Given an XML file, **CS_XmlReader(fileName)** attempts to open the file and read its contents as if it conformed to the canSAS XML standard for reduced 1-D SAS data (cansas1d/1.0, also known as SASXML). If the file is found to be non-conforming, then **CS_XmlReader(fileName)** returns with an error code (show below), otherwise it returns **0** that indicates *no error*. All data read by this code is left in the IgorPro data folder ***root:Packages:CS_XMLreader*** for pickup by the calling routine. (Two examples are provided to show how a routine might retrieve the data.)

After opening the XML file (with a file identifier *fileID*), control is passed to **CS_1i_parseXml(fileID)** which then walks through the XML elements. For each **SASentry** in the file, a new data folder is created with the name derived from the Title element (or best effort determination). Efforts are taken to avoid duplication of data folder names (using standard IgorPro routines). For **SASentry** elements that contain more than one SASdata element, a **SASdata** folder is created for each and the corresponding *I(Q)* is placed in that subfolder. When only one **SASdata** is found, the *I(Q)* data is placed in the main *Title* folder.

data columns

Each column of data in the **SASdata/Idata/*** table is placed into a single IgorPro wave. At present, the code does not check for non-standard data columns. (The capability is built into the code but is deactivated at present).

metadata

Additional metadata is collected into a single text wave (*metadata*) where the first column is an identifier (or *key*) and the second identifier is the *value*. Only those keys with non-empty values are retained in the metadata table. **CAUTION:** The *values* are not checked for characters that may cause trouble when placed in a wave note. This will be the responsibility of the calling routine to *clean these up* if the need arises.

The code checks for most metadata elements and will check for repeated elements where the standard permits.

Here is an example of the metadata for the *cs_collagen_full.xml* data:

i **(*row*)**	**metadata[*i*][0] (*key*)**	**metadata[*i*][1] (*value*)**
0	xmlFile	cs_collagen_full.xml
1	namespace	cansas1d/1.0
2	Title	dry chick collagen, d = 673 A, 6531 eV, X6B
3	Run	Sep 19 1994 01:41:02 am
4	SASsample/ID	dry chick collagen, d = 673 A, 6531 eV, X6B
5	SASinstrument/name	X6B, NSLS, BNL
6	SASinstrument/SASsource /radiation	X-ray synchrotron
7	SASinstrument/SASsource /wavelength	1.898
8	SASinstrument/SASsource /wavelength/@unit	A
9	SASinstrument/SASdetector /@name	X6B PSD

10	SASnote	`Sep 19 1994     01:41:02 am     Elt: 00090 Seconds`
		`ID: No spectrum identifier defined`
		`Memory Size: 8192 Chls  Conversion Gain: 1024  Adc Offset: 0000 Chls`
		
		`dry chick collagen, d = 673 A`
		`6531 eV, X6B`

XML foreign namespace elements

These are ignored at this time.

XML namespace and header

The routine does a *best-efforts* check to ensure that the given XML file conforms to the required XML file header. If you take a minimalist view (*a.k.a.* a shortcut), it is likely that your file may be refused by this and other readers. Pay particular attention to UPPER/lower case in the text **cansas1d/1.0** as this is a **key component** used to index through the XML file.

XML stylesheet processing-instruction is not generated

The **XMLutils** (http://www.igorexchange.com/project/XMLutils) package does not provide a method to insert the prescribed XML stylesheet processing-instruction into the XML data file.

```
<?xml-stylesheet type="text/xsl" href="example.xsl" ?>
```

If this processing-instruction is desired, it must be added to each XML data file by other methods such as use of a text editor or application of an XSLT transformation.

List of Functions

These are (most of) the FUNCTIONS in the cansasXML.ipf code. The only functions of interest are **CS_XmlReader(fileName)** which reads the named XML file and and loads SAS data and the two demonstration functions **prj_grabMyXmlData()** and **prjTest_cansas1d()** that together show a usage example.

CS_XmlReader(fileName)
: open a canSAS 1-D reduced SAS XML data file

- input: *fileName* (string) name of canSAS XML file (can include file system path name to file)
- returns:
  - 0 successful
  - -1: XML file not found
  - -2: root element is not <SASroot> with valid canSAS namespace
  - -3: <SASroot> version is not 1.0
  - -4: no <SASentry> elements (NOT USED NOW)
  - -5: XOPutils needs upgrade

CS_1i_parseXml(fileID)
: **This is what guides the work**, given a file ID returned from **XMLOpenFile()**, parses that file for SAS data and metadata
: (1i in the function name signifies this is a function that supports INPUT from version 1.0 XML files)

CS_1i_getOneSASdata(fileID, Title, SASdataPath)
: harvest the data and metadata in the specific SASdata element

CS_1i_getOneVector(file,prefix,XML_name,Igor_name)
: harvest just one column (vector) of data

CS_1i_GetReducedSASdata(fileID, SASdataPath)
: grab the data and put it in the working data folder

CS_1i_locateTitle(fileID, SASentryPath)
: determine the title for this experiment

CS_appendMetaData(fileID, key, xpath, value)
: queries XML file for **xpath**. If **value** is not empty, appends it to **metadata** where *last* is the new last row: metadata[last][0]=key; metadata[last][1]=value

CS_buildXpathStr(prefix, value)
: this function can be used only with very simple XPath constructions

CS_cleanFolderName(proposal)
: given a proposal string, returns a candidate folder name for immediate use

CS_findElementIndex(matchStr)
: looks for element index in structure *W_ElementList* returned from call to **XmlElemList(fileID)**

CS_getDefaultNamespace(fileID)

returns the string containing the default namespace for the XML file

CS_registerNameSpaces()
: Builds a table of all namespaces used in the XML file and appends **W_ElementList** with full namespace-xpath string for each element.

CS_simpleXmlListXpath(fileID, prefix, value)
: Calls **XMLlistXpath()** with proper namespace prefix attached.

CS_simpleXmlWaveFmXpath(fileID, prefix, value)
: Calls **XMLwaveFmXpath()** with proper namespace prefix attached.

CS_updateWaveNote(wavName, key, value)
: adds (or replaces) definition of *key*=*value* in the wave note of *wavName*

CS_XmlStrFmXpath(fileID, prefix, value)
: Calls **XmlStrFmXpath()** with proper namespace prefix attached. Trims the result string.

CS_XPath_NS(simpleStr)
: this function adds namespace info as necessary to simpleStr (an XPath)

TrimWS(str)
: Calls **TrimWSL(TrimWSR(str))**

TrimWSL(str)
: Trims white space from left (leading) end of **str**

TrimWSR(str)
: Trims white space from right (trailing) end of **str**

prjTest_cansas1d()
: Demonstration function that calls **CS_XmlReader(fileName)** for many of the test data sets.

prj_grabMyXmlData()
: Demonstration function that moves loaded data from root:Packages:CS_XMLreader to a user's data folder. (In this *example*, that folder is root:PRJ_canSAS.)

testCollette()
: Demonstration function that reads an ISIS/LOQ file and copies the data to the root folder a la COLLETE

Example test case

Here is an example running the test routine **prjTest_cansas1d()**.

```
•prjTest_cansas1d()
  XMLopenfile: File(path) to open doesn't exist, or file can't be opened
  elmo.xml either not found or cannot be opened for reading
         Completed in   0.00669666   seconds
  XMLopenfile: XML file was not parseable
  cansasXML.ipf: failed to parse XML
         Completed in   0.0133704   seconds
  root element is not <SASroot> with valid canSAS namespace
         Completed in   0.0134224   seconds
  bimodal-test1.xml              identified as: cansas1d/1.0 XML file
         Title:  SAS bimodal test1
         Completed in   0.068654   seconds
  root element is not <SASroot> with valid canSAS namespace
         Completed in   0.0172572   seconds
  root element is not <SASroot> with valid canSAS namespace
         Completed in   0.0123102   seconds
  root element is not <SASroot> with valid canSAS namespace
         Completed in   0.00930118   seconds
  ISIS_SANS_Example.xml                  identified as: cansas1d/1.0 XML file
         Title:   standard can 12mm SANS
         Completed in   0.0410387   seconds
  W1W2.xml               identified as: cansas1d/1.0 XML file
         Title:   standard can 12mm SANS
         Title:   TK49 standard 12mm SANS
         Completed in   0.0669074   seconds
  ill_sasxml_example.xml                 identified as: cansas1d/1.0 XML file
         Title:  ILL-D22 example: 7D1 2mm
         Completed in   0.0332752   seconds
  isis_sasxml_example.xml                identified as: cansas1d/1.0 XML file
         Title:   LOQ TK49 Standard 12mm C9
         Completed in   0.0388868   seconds
  r586.xml               identified as: cansas1d/1.0 XML file
         Title:  ILL-D11 example1: 2A 5mM 0%D2O
         Completed in   0.0213737   seconds
  r597.xml               identified as: cansas1d/1.0 XML file
         Title:  ILL-D11 example2: 2A 5mM 0%D2O
         Completed in   0.0221894   seconds
  xg009036_001.xml               identified as: cansas1d/1.0 XML file
         Title:  det corrn 5m
         Completed in   0.0286721   seconds
  cs_collagen.xml                identified as: cansas1d/1.0 XML file
         Title:  dry chick collagen, d = 673 A, 6531 eV, X6B
         Completed in   0.0296247   seconds
  cs_collagen_full.xml                   identified as: cansas1d/1.0 XML file
         Title:  dry chick collagen, d = 673 A, 6531 eV, X6B
         Completed in   0.0751836   seconds
  cs_af1410.xml                  identified as: cansas1d/1.0 XML file
         Title:  AF1410-10 (AF1410 steel aged 10 h)
         Title:  AF1410-8h (AF1410 steel aged 8 h)
         Title:  AF1410-qu (AF1410 steel aged 0.25 h)
         Title:  AF1410-cc (AF1410 steel aged 100 h)
         Title:  AF1410-2h (AF1410 steel aged 2 h)
         Title:  AF1410-50 (AF1410 steel aged 50 h)
         Title:  AF1410-20 (AF1410 steel aged 20 h)
         Title:  AF1410-5h (AF1410 steel aged 5 h)
         Title:  AF1410-1h (AF1410 steel aged 1 h)
         Title:  AF1410-hf (AF1410 steel aged 0.5 h)
         Completed in   0.338425   seconds
  XMLopenfile: File(path) to open doesn't exist, or file can't be opened
  cansas1d-template.xml either not found or cannot be opened for reading
         Completed in   0.00892823   seconds
  1998spheres.xml                identified as: cansas1d/1.0 XML file
         Title:  255 nm PS spheres
         Title:  460 nm PS spheres
         Completed in   2.87649   seconds
  XMLopenfile: File(path) to open doesn't exist, or file can't be opened
  does-not-exist-file.xml either not found or cannot be opened for reading
         Completed in   0.00404549   seconds
  cs_rr_polymers.xml             identified as: cansas1d/1.0 XML file
         Title:  Round Robin Polymer A
         Title:  Round Robin Polymer B
         Title:  Round Robin Polymer C
         Title:  Round Robin Polymer D
         Completed in   0.0943477   seconds
  s81-polyurea.xml               identified as: cansas1d/1.0 XML file
         Title:  S7 Neat Polyurea
         Completed in   0.0361616   seconds
```

Graphical User Interface

At least two groups are working on graphical user interfaces that use the canSAS 1-D SAS XML format binding to IgorPro. The GUIs are intended to be used with their suites of SAS analysis tools (and hide the details of using this support code from the user).

NOTE: There is no support yet for writing the data back into the canSAS format. Several details need to be described, and these are being collected on the discussion page for the XML format

Irena tool suite

Jan Ilavsky's **Irena** tool suite (http://usaxs.xor.aps.anl.gov/staff/ilavsky/irena.html) for IgorPro has a GUI to load the data found in the XML file. Refer to the WWW site for more details.