NeXus Hands on Workshop

For online API documentation visit http://download.nexusformat.org/

Example 1

Sample solutions: example1.cpp, example1.python, example1.java

Generate a simple nexus program to:

- Open a new HDF5 file called **test1.nxs**
- Create a group called **entry 1** of class NXentry
- Inside **entry1** create a data element called **counts** – this is a one dimensional integer array
- Also Inside **entry1** create a data element called **time** – this is a one dimensional float array of the same length as **counts**
- Add a attribute called **attr1** to **time** of value 1.0
- Close the file

Compile the program (or the sample solution) with:

```
g++ -o example1 example1.cxx `nexus-config --cflags --libs`      # C++
javac –cp /path/to/jnexus.jar example1.java                      # JAVA
```

And then run

```
./example1                                                       # C++
java –cp .:/path/to/jnexus.jar   example1                        # JAVA
python example1.py                                               # PYTHON
```

To create **test1.nxs**

Using NXbrowse

To view the file with the **nxbrowse** utility type:

```
nxbrowse test1.nxs
```

Type **help** to see a list of commands and then enter

```
NX> dir
NX> cd entry1
NX> dir
NX> read counts
NX> dump counts test1.txt
NX> exit
```

You should now have a “test1.txt” file with counts data

Using NXconvert

Now use the nxconvert utility to turn the NeXus HDF5 representation into its XML equivalent

```
nxconvert -x test1.nxs test1.xml
```

and examine **test1.xml** with an editor

Using NXdir

The contents can also be output using **nxdir**

```
nxdir   -o  -p /entry1/counts   test1.nxs      #  you can also use  test1.xml  instead of test1.xml
```

Using NXsummary

First generate a basic configuration file using

```
nxsummary --writeconfig nxsummary.conf
```

Edit **nxsummary.conf** and change the path /entry/monitor/data for label "TOTAL MONITOR" to **/entry1/time** and then run

```
nxsummary --config  nxsummary.conf  test1.nxs
```

nxsummary --help lists other options you can try

Using NXdiff

Rename test1.nxs to test1_old.nxs, change the code to alter the data in either the counts or time array, create a new test1.nxs and try

```
nxdiff test1_old.nxs test1.nxs
```

Using NXtranslate

Run

```
nxtranslate  --hdf5  example1.nxt
```

This will extract portions of test1.nxs into a new file **example1.nxs** based on rules in example1.nxt

Using NXextract

Run

```
nxextract –t  example1.nxe  test1.nxs
```

This will create **example1.txt** based on rules in **example1.nxe**