

Reading NeXus Files

A Recommendation

Mark Könnecke
Laboratory for Neutron Scattering
Paul Scherrer Institute
5232-Villigen-PSI
Switzerland
December 7, 2004

1 Introduction

The intended audience of this document are programmers who wish to read NeXus files into their programs for further processing. Recent additions to the NeXus-API allow code for reading NeXus files to be written in a more clever way. This document describes how these new NeXus-API features are put to best use.

2 The Old Way to Read a NeXus File

The code sample below is the code necessary to open a NeXus file with the old NeXus-API functions and to read a single data item from it.

```
#include <napi.h>
int main(int argc, char *argv[])
{
    int datatype, rank, iDim[2], i, j;
    float floatData[5][5];
    NXhandle handle;
    NXstatus test;

    test = NXopen("workshop.hdf",NXACC_READ,&handle);
    test = NXopengroup(handle,"entry_1","NXentry");
    test = NXopendata(handle,"testdata");
    test = NXgetinfo(handle,&rank, iDim,&datatype);
    test = NXgetdata(handle, floatData);
    test = NXclosedata(handle);
    test = NXclosegroup(handle);
    test = NXclose(&handle);
}
```

This is a lot of code. In production code there would also be tests at each step in order to catch errors. This example will continue to work but we can do better.

3 The Magic of NXopenpath

There is a new NeXus-API function, NXopenpath, which opens groups and datasets from a path string, not unlike a unix path string. Both absolute and relative paths are supported. With NXopenpath the example above reduces to:

```
#include <napi.h>
```

```

int main(int argc, char *argv[])
{
    int datatype, rank, iDim[2], i, j;
    float floatData[5][5];
    NXhandle handle;
    NXstatus test;

    test = NXopen("workshop.hdf",NXACC_READ,&handle);
    test = NXopenpath(handle,"/entry_1/testdata");
    test = NXgetinfo(handle,&rank, iDim,&datatype);
    test = NXgetdata(handle, floatData);
    test = NXclose(&handle);
}

```

Rather than using sequences of `NXopengroup`, `NXopengroup`, `NXopendata` `NXopenpath` allows to address items in the file directly through its path. Please note, that the path is built from the names of the groups and datasets, the NeXus classes are ignored.

4 NXopenpath and Dictionaries

In the example above, the path string was hardcoded into the program. However, much can be gained when the path string to use for a desired data item is read from some form of a dictionary, for instance from a properties file or an initialization file. The advantage of this approach comes to bear if a not completely standard NeXus file comes along. Assuming that all necessary information is contained in the file, a NeXus reading program using the dictionary approach can read such a file without changing a single line of code after completing these two steps:

- Locate the necessary data items in the NeXus file using a NeXus browser and take a note of their paths within the file.
- Edit the configuration or properties file to reflect the new paths.

Usually a given program, for example a data analysis program, only needs a very limited subset of the information in the NeXus file. So the process described above should not take too much time.

`NXopenpath` used with dictionaries can overcome one of the weaknesses of the NeXus proposal: Currently there are very different NeXus files around because instrument definitions and name definitions were slow to emerge. Moreover this approach also offers a quick solution when people misinterpret the NeXus standard and write non conforming files. They usually do so in thousands before the error is spotted or even resolved.