NAPI Advanced Techniques

Mark Könnecke

Paul Scherrer Institute
Switzerland

May 11, 2010

Linking

```
#- Step 1: open the dataset to link
nf.openpath(“/entry/instrument/detector/data”)
#- Step 2: get the NXlink data
nxl = nf.getdataID()
#- Step 3: navigate to the target group
nf.openpath(“/entry/data”)
#- Step 4: actually link
nf.makelink(nxl)
```

Reading NeXus Files Stupidly

```
nxfile = nxs.open(’hrpt2008n152088.hdf’,’r’)
nxfile.openpath(’/entry1/data1/two_theta’)
x = nxfile.getdata()
nxfile.openpath(’/entry1/data1/counts’)
y = nxfile.getdata()
nxfile.openpath(’/entry1/title’)
txt = nxfile.getdata()
nxfile.close()

plot(x,y)
xlabel(’two theta’)
ylabel(’counts’)
title(txt)
show()
```

Changes

- The file to plot becomes a command line argument

Changes

- The file to plot becomes a command line argument
- The path strings to use as x,y and title are put into variables

Changes

- The file to plot becomes a command line argument
- The path strings to use as x,y and title are put into variables
- The axis descriptions are changed to use the data names

Changes

- The file to plot becomes a command line argument
- The path strings to use as x,y and title are put into variables
- The axis descriptions are changed to use the data names
- The path strings to use for x,y and title are externalized into a separate file

Sly NeXus Reading

```
nxfile = nxs.open(sys.argv[1],'r')
xpath,ypath,titlepath = loadPathNames()
nxfile.openpath(xpath)
x = nxfile.getdata()
nxfile.openpath(ypath)
y = nxfile.getdata()
nxfile.openpath(titlepath)
txt = nxfile.getdata()
nxfile.close()

plot(x,y)
xlabel(extractName(xpath))
ylabel(extractName(ypath))
title(txt)
show()
```

Writing NeXus File with NXdict

- Writing NeXus files can be tedious

Writing NeXus File with NXdict

- Writing NeXus files can be tedious
- Your computer is good at tedious things

Writing NeXus File with NXdict

- Writing NeXus files can be tedious
- Your computer is good at tedious things
- NXdict

Writing NeXus File with NXdict

- Writing NeXus files can be tedious
- Your computer is good at tedious things
- NXdict
    - Provides a description for a SDS in a NeXus hierarchy

Writing NeXus File with NXdict

- Writing NeXus files can be tedious
- Your computer is good at tedious things
- NXdict
    - Provides a description for a SDS in a NeXus hierarchy
    - Externalises that description

Writing NeXus File with NXdict

- Writing NeXus files can be tedious
- Your computer is good at tedious things
- NXdict
  - Provides a description for a SDS in a NeXus hierarchy
  - Externalises that description
  - An API for reading/writing data using the description

Definition Strings

/entry1,NXentry/SANS,NXinstrument/detector,NXdetector/SDS counts -type NX_UINT32 -LZW -rank 2 -dim {128,128} -attr {signal,1}

Default: single floating point number

NXdict File

```
##NXDICT-1.0
#----- NXentry
etitle = /entry1,NXentry/SDS title -type NX_CHAR -rank 1
-dim 132
etime = /entry1,NXentry/SDS start_time -type NX_CHAR   -ran
endtime = /entry1,NXentry/SDS end_time -type NX_CHAR   -ran
#------ NXinstrument
iname = /entry1,NXentry/SANS,NXinstrument/SDS  name   -type
```

NXdict Dictionary Maintenance

```
NXstatus NXDinitfromfile(char *filename, NXdict * pDict);
NXstatus NXDclose(NXdict handle, char *filename);

NXstatus NXDadd(NXdict handle, char *alias,
                char *DefString);
NXstatus NXDget(NXdict handle, char *alias,
                char *pBuffer, int iBufLen);
NXstatus NXDupdate(NXdict handle,
                   char *alias, char *pNewVal);
NXstatus NXDtextreplace(NXdict handle, char *pDefString,
                        char *pBuffer, int iBuflen);
```

NXdict I/O

```
NXstatus NXDputalias(NXhandle file, NXdict dict,
                     char *alias, void *pData);
NXstatus NXDgetalias(NXhandle file, NXdict dict, char *alias
                     void *pData);
NXstatus NXDdefget(NXdict handle, char *pKey,
                   char *pBuffer, int iBufLen);
NXstatus NXDaliaslink(NXhandle file, NXdict dict,
                      char *pAlias1, char *pAlias2);
NXstatus NXDopenalias(NXhandle file, NXdict dict,
                      char *alias);
```

NXdict Example

```
NXdict dict;
NXhandle hfil;
float val = 27.8;

NXopen(“hugo.hdf”,NXACC_CREATE5,&hfil)
NXDinitfromfile(“hugo.dic”,&dict)

NXDputalias(hfil,dict, “hugo”, &val);
NXDputalias(hfil,dict,”data”,NULL);
NXDaliaslink(hfil,dict,”hugo”, “data”);
NXclose(&hfil);
```

Writing NeXus with the HDF-5 API

- NeXus groups map to HDF groups, NeXus classes map to a HDF-5 group attribute with name NX_class
- NeXus datasets map to HDF-5 datasets
- NeXus attributes map to HDF-5 group or dataset attributes
- On linking a special attribute target must be created with the path of the linked data item as a value
- External linking is implemented with a special group attribute with the name NAPImount. The value is the URL into the external file