NeXus Technical Reference Manual

Edited by
Peter F Peterson

NeXus Technical Reference Manual

Edited by Peter F Peterson

Table of Contents

1. Overview **1**
2. Terminology **3**
The NeXus XML Meta-DTD Format 3
3. Methodology **5**
4. Base Classes **7**
5. Definitions **33**
Monochromatic Reflectometry 33
Monochromatic Triple Axis Spectrometer 40
Time of Flight Neutron Direct Geometry Spectrometer 43
Time of Flight Neutron Powder Diffractometer 45
TOFNPD:Time Focus 46
Time of Flight Neutron Reflectometer 48

Chapter 1. Overview

An overview of NeXus goes here.

Chapter 2. Terminology

The NeXus XML Meta-DTD Format

The contents of NeXus files are defined using XML. The hierarchical structure of NeXus files maps very conveniently into XML files with NeXus groups and items as the XML entities, and data attributes as XML attributes. NeXus utilities are being developed to help people determine whether their files are standard-conforming. However, formal XML format definitions (DTDs) are difficult for the non-expert to read, so we have produced a much simpler meta-DTD format, which produces well-formed (DTD-less) XML files that will be converted into DTD files. This page describes the rules for producing these files - some examples are available below and in the NeXus content section. The utility NXtoDTD can be used to generate the skeleton of such a file from an existing NeXus file; it outputs the XML tags without the data or any annotation. Meta-DTD Definition

1. Each meta-DTD file should begin with a standard XML document tag, i.e.

   ```
   <?xml version="1.0" ?>
   ```

2. This should be followed by a comment block giving the URL of the XML file, the name of the editor, the keyword $Id: terminology.docbook 500 2005-06-14 16:50:35Z pfp $, which will generate a revision number when the file is committed to the NeXus CVS server, and a brief description of the file, e.g.

   ```
   <!--
   URL:     http://www.neutron.anl.gov/nexus/xml/NXgroup.xml
   Editor:  Jean Dupont <JDupont@some.where>
   $Id: terminology.docbook 500 2005-06-14 16:50:35Z pfp $
   Definition of a fake but well-formed NeXus group.
   -->
   ```

3. Each NeXus group is an XML entity defined by its class, e.g. NXuser, NXdata, ...

4. The name of the group is given by the name attribute of the entity. N.B. XML attributes are the name="value" pairs located within the opening tag of the XML entity, e.g.

   ```
   <NXsample name="sample">
   ```

   .

5. All other data items are XML entities defined by their name, e.g.

   ```
   <temperature>
   ```

6. Data attributes are stored as XML attributes. The data type is defined as an XML attribute although it is not defined as an HDF attribute in the NeXus file itself, e.g.

   ```
   <temperature type="NX_FLOAT32" units="K">
   ```

7. If the value of an attribute is not defined by the DTD, a short description is enclosed within quotes and curly braces, e.g.

   ```
   <NXdetector name="{Name of detector bank}">
   ```

8. Similarly, the value of a data item which is not defined by the DTD should be placed within curly braces between the opening and closing tag, e.g.

   ```
   <temperature>{Temperature of sample}</temperature>
   ```

9. Following the opening tag of a group entity and before the closing tag of a data entity, there may be one of three symbols, which have the same meanings that they have in regular expressions.

   ```
   *   May occur 0 or more times
   +   May occur one or more times (i.e. at least once)
   ```

```
?   May occur 0 or one times (i.e. no more than once)
```

e.g.

```
<NXsample>?
   <temperature>{Temperature of sample}?</temperature>
</NXsample>
```

If no symbol is given, the item is mandatory.

10. If a data item is an array, add the array dimensions in square brackets to the type attribute. Use a colon if the dimension length is not defined by the DTD, e.g.

    ```
    <polar_angle type="NX_FLOAT32[:]">
    ```

    Replace the colon with i, j, ... if you wish to match the dimension length to other data items within the same group.

11. If no data type is specified, it is assumed to be a character string (NX_CHAR).

12. The "version" attribute of the "analysis" entity, defined in each NXentry group should be set to $Revision: 500 $ when the file is first written so that the CVS revision number is substituted when the XML file is committed to the CVS server, e.g.

    ```
    <analysis version="$Revision: 500 $">
    ```

Chapter 3. Methodology

To send out a call to the instrument editors to assist in writing instrument and other specialized NeXus file definitions by following these steps:

a. Draw a schematic diagram of the instrument (line art, not a CAD style drawing).

b. Write a brief document outlining the purpose of the instrument. In the summary, list existing packages that perform data reduction and/or analysis for this type of data. Also list the information that will (and will not) exist in the file including, but not limited to, characterization measurements used for subtracting a background or normalizing to an incident spectrum.

c. Ask the people that maintain the packages listed in step b to provide a list of essential variables that the data file should contain.

d. Send the document and the diagram to the NIAC who will nominate someone to construct a XML definition.

e. The NIAC will construct the XML definition.

f. Write a sample NeXus file conforming to the instrument definition (for example, using NXtranslate).

g. Test the file, repeating steps e and f as appropriate.

h. Request ratification once the testing phase is complete.

Chapter 4. Base Classes

This chapter will list all of the base classes and their contents.

Example 4-1. `NXaperture.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXaperture.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of a beamline aperture.-->
<NXaperture name="{Name of aperture}">
  <NXgeometry name="">
    {location and shape of aperture}?
  </NXgeometry>
  <material type="NX_CHAR">
    {Absorbing material of the aperture}?
  </material>
  <description type="NX_CHAR">
    {Description of aperture}?
  </description>
</NXaperture>
```

Example 4-2. `NXattenuator.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXaperture.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of a beamline attenuator.-->
<NXattenuator name="{Name of attenuator}">
  <distance units="m" type="NX_FLOAT">
    {Distance from sample}?
  </distance>
  <type type="NX_CHAR">
    {Type of attenuator, e.g. polythene}?
  </type>
  <thickness units="cm" type="NX_FLOAT">
    {Thickness of attenuator along beam direction}?
  </thickness>
  <scattering_cross_section units="barns" type="NX_FLOAT">
    {Scattering cross section (coherent+incoherent)}?
  </scattering_cross_section>
  <absorption_cross_section units="barns" type="NX_FLOAT">
    {Absorption cross section}?
  </absorption_cross_section>
  <attenuator_transmission type="NX_FLOAT">
    {The nominal amount of the beam that gets through (transmitted
    intensity)/(incident intensity)}?
  </attenuator_transmission>
</NXattenuator>
```

Example 4-3. `NXbeam_stop.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXbeam_stop.xml
Editor:  Mark Koennecke
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

A class for a beamstop. Beamstops and their positions are important for SANS
and SAXS experiments.
```

```
-->
<NXbeam_stop name="">
  <NXgeometry name="">
    {engineering shape, orientation and position of the beam stop.}?
  </NXgeometry>
  <description type="NX_CHAR">
    {description of beamstop: circular | rectangular}?
  </description>
  <size units="cm" type="NX_FLOAT">
    {size of beamstop}?
  </size>
  <x units="cm" type="NX_FLOAT">
    {x position of the beamstop in relation to the detector}?
  </x>
  <y units="cm" type="NX_FLOAT">
    {y position of the beamstop in relation to the detector}?
  </y>
  <distance_to_detector units="cm" type="NX_FLOAT">
    {distance of the beamstop to the detector}
  </distance_to_detector>
  <status type="NX_CHAR">
    {in|out}?
  </status>
</NXbeam_stop>
```

Example 4-4. `NXbeam.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXbeam.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of the state of the neutron or X-ray beam at any location. It will be
referenced by beamline component groups within the NXinstrument group or by the
NXsample group. Note that variables such as the incident energy could be scalar
values or arrays. This group is especially valuable in storing the results of
instrument simulations in which it is useful to specify the beam profile, time
distribution etc. at each beamline component. Otherwise, its most likely use is
in the NXsample group in which it defines the results of the neutron scattering
by the sample, e.g., energy transfer, polarizations.
-->
<NXbeam name="{Name of beam plane}">
  <distance units="m" type="NX_FLOAT">
    {Distance from sample}?
  </distance>
  <incident_energy units="meV" type="NX_FLOAT[:]">
    {Energy on entering beamline component}?
  </incident_energy>
  <final_energy units="meV" type="NX_FLOAT[:]">
    {Energy on leaving beamline component}?
  </final_energy>
  <energy_transfer units="meV" type="NX_FLOAT[:]">
    {Energy change caused by beamline component }?
  </energy_transfer>
  <incident_wavelength units="Angstroms" type="NX_FLOAT[:]">
    {Wavelength on entering beamline component}?
  </incident_wavelength>
  <incident_wavelength_spread units="Angstroms" type="NX_FLOAT[:]">
    {Wavelength spread FWHM on entering component}?
  </incident_wavelength_spread>
  <incident_beam_divergence units="degree" type="NX_FLOAT[2,:]">
    {Divergence of beam entering this component}?
  </incident_beam_divergence>
  <final_wavelength type="NX_FLOAT[:]">
    {Wavelength on leaving beamline component}?
```

```
  </final_wavelength>
  <incident_polarization type="NX_FLOAT[3,:]">
    {Polarization vector on entering beamline component}?
  </incident_polarization>
  <final_polarization type="NX_FLOAT[3,:]">
    {Polarization vector on leaving beamline component}?
  </final_polarization>
  <final_wavelength_spread units="Angstroms" type="NX_FLOAT[:]">
    {Wavelength spread FWHM of beam leaving this component}?
  </final_wavelength_spread>
  <final_beam_divergence units="degrees" type="NX_FLOAT[2,:]">
    {Divergence FWHM of beam leaving this component}?
  </final_beam_divergence>
  <flux units="s-1cm-2" type="NX_FLOAT[i]">
    {flux incident on beam plane area}?
  </flux>
  <NXdata name="{spectrum}">
    {Distribution of beam with respect to relevant variable e.g. wavelength.
    This is mainly useful for simulations which need to store plottable
    information at each beamline component. }?
  </NXdata>
</NXbeam>
```

Example 4-5. NXcharacterizations.xml

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXentry.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of the top-level NeXus group which contains all the data and
associated information that comprise a single measurement. It is mandatory that
there is at least one group of this type in the NeXus file.
-->
<NXcharacterization NXS:location="" NXS:source="{If missing the source file is
    the current file}?"
    name="empty_environment|empty_environment_background|empty_container
    |empty_container_background|isotropic_scatterer
    |isotropic_scatterer_background" NXS:mime_type="{If missing the
    source file is NAPI readable}?">
  <definition URL="?" version="?">?
  </definition>
</NXcharacterization>
```

Example 4-6. NXcollimator.xml

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXcollimator.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of a beamline collimator.-->
<NXcollimator name="{Name of collimator}">
  <NXgeometry name="">
    {position, shape and size}?
  </NXgeometry>
  <type type="NX_CHAR">
    "Soller"|"radial"|"oscillating"|"honeycomb"?
  </type>
  <soller_angle units="minutes" type="NX_FLOAT">
    {Angular divergence of Soller collimator}?
  </soller_angle>
  <divergence_x type="NX_FLOAT">
    {divergence of collimator in local x direction}?
```

```
  </divergence_x>
  <divergence_y type="NX_FLOAT">
    {divergence of collimator in local y direction}?
  </divergence_y>
  <frequency type="NX_FLOAT">
    {Frequency of oscillating collimator}?
  </frequency>
  <NXlog name="frequency">
    {Log of frequency}?
  </NXlog>
  <blade_thickness type="NX_FLOAT">
  </blade_thickness>
  <blade_spacing type="NX_FLOAT">
  </blade_spacing>
  <absorbing_material type="NX_CHAR">
  </absorbing_material>
  <transmitting_material type="NX_CHAR">
  </transmitting_material>
</NXcollimator>
```

Example 4-7. NXcrystal.xml

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXcrystal.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of a crystal monochromator or analyzer. Permits double bent
monochromator comprised of multiple segments with anisotropic  Gaussian mosaic.
If curvatures are set to zero or are absent, array  is considered to be flat.
Scattering vector is perpendicular to surface. Crystal is oriented parallel to
beam incident on crystal before rotation, and lies in vertical plane.
-->
<NXcrystal name="{Name of crystal}">
  <NXgeometry name="">
    {Position of crystal}?
  </NXgeometry>
  <type type="NX_CHAR">
    { "PG (Highly Oriented Pyrolytic Graphite)" | "Ge" | "Si" | "Cu" | "Fe3Si"
    | "CoFe" | "Cu2MnAl (Heusler)" | "Multilayer" }?
  </type>
  <cut_angle units="degrees" type="NXFLOAT">
    {Cut angle of reflecting Bragg plane and plane of crystal surface}?
  </cut_angle>
  <unit_cell type="NX_FLOAT[6])">
    {Unit cell parameters (lengths and angles)}?
  </unit_cell>
  <unit_cell_volume units="Angstroms3" type="NX_FLOAT" rank="1">
    {Volume of the unit cell}?
  </unit_cell_volume>
  <orientation_matrix type="NX_FLOAT[3,3]">
    {Orientation matrix of single crystal sample using Busing-Levy convention}?
  </orientation_matrix>
  <wavelength units="Angstroms" type="NX_FLOAT[i]">
    {Optimum diffracted wavelength}?
  </wavelength>
  <lattice_parameter units="Angstrom" type="NX_FLOAT">
    {Lattice parameter of the nominal reflection}?
  </lattice_parameter>
  <scattering_vector units="Angstrom^-1" type="NX_FLOAT">
    {Scattering vector, Q, of nominal reflection}?
  </scattering_vector>
  <reflection type="NX_INT[3]">
    {[hkl] values of nominal reflection}?
  </reflection>
```

```
  <segment_width units="m" type="NX_FLOAT">
    {Horizontal width of individual segment}?
  </segment_width>
  <segment_height units="m" type="NX_FLOAT">
    {Vertical height of individual segment}?
  </segment_height>
  <segment_thickness units="m" type="NX_FLOAT">
    {Thickness of individual segment}?
  </segment_thickness>
  <segment_gap units="m" type="NX_FLOAT">
    {Typical gap between adjacent segments}?
  </segment_gap>
  <segment_columns units="m" type="NXFLOAT">
    {number of segment columns in horizontal direction}?
  </segment_columns>
  <segment_rows units="m" type="NXFLOAT">
    {number of segment rows in vertical direction}?
  </segment_rows>
  <mosaic_horizontal units="arc minutes" type="NXFLOAT">
    {horizontal mosaic Full Width Half Maximum}?
  </mosaic_horizontal>
  <mosaic_vertical units="arc minutes" type="NXFLOAT">
    {vertical mosaic Full Width Half Maximum}?
  </mosaic_vertical>
  <curvature_horizontal units="degrees" type="NX_FLOAT">
    {Horizontal curvature of focusing crystal}?
  </curvature_horizontal>
  <curvature_vertical units="degrees" type="NX_FLOAT">
    {Vertical curvature of focusing crystal}?
  </curvature_vertical>
  <polar_angle units="degrees" type="NX_FLOAT[i]">
    {Polar (scattering) angle at which crystal assembly is positioned}?
  </polar_angle>
  <azimuthal_angle units="degrees" type="NX_FLOAT[i]">
    {Azimuthal angle at which crystal assembly is positioned}?
  </azimuthal_angle>
  <bragg_angle units="degrees" type="NX_FLOAT[i]">
    {Bragg angle of nominal reflection}?
  </bragg_angle>
  <temperature Units="Kelvin" type="NX_FLOAT">
    {average/nominal crystal temperature}
  </temperature>
  <temperature_log type="NXlog">
    {log file of crystal temperature}
  </temperature_log>
  <reflectivity type="NXdata">
    {crystal reflectivity versus wavelength }
  </reflectivity>
  <transmission type="NXdata">
    {crystal transmission versus wavelength }
  </transmission>
</NXcrystal>
```

Example 4-8. **NXdata.xml**

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXdata.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of plottable data and their dimension scales. It is mandatory that
there is at least one group of this type in each NXentry group.  Note that
"variable" and "data" can be defined with different names.  The "signal" and
"axes" attribute of the "data" item define which items are plottable data and
which are dimension scales.
```

```
-->
<NXdata name="{Name of data block}">
  <variable long_name="{Axis label}" distribution="0|1" first_good="{Index of
      first good value}" type="NX_FLOAT[:]|NX_INT[:]" last_good="{Index of last
      good value}">
    {Dimension scale defining an axis of the data}?
  </variable>
  <variable_errors type="NX_FLOAT[:]|NX_INT[:]">
    {Errors associated with axis "variable"}?
  </variable_errors>
  <data long_name="{Title of data}" signal="1" axes="[...]"
      type="NX_FLOAT[:...]|NX_INT[:...]">
    {Data values}?
  </data>
  <errors type="NX_FLOAT[:...]">
    {Standard deviations of data values - the data array is identified by the
    attribute signal="1". This array must have the same dimensions as the data}?
  </errors>
</NXdata>
```

Example 4-9. `NXdetector.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXdetector.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of a detector, detector bank, or multidetector.-->
<NXdetector name="{Name of detector bank}">
  <time_of_flight primary="1?" long_name="{Axis label}" link="{absolute path to
      location in NXdetector}" units="10^-6 second|10^-7 second"
      type="NX_FLOAT[j+1]" axis="1">
    {Total time of flight}
  </time_of_flight>
  <detector_number link="{absolute path to location in NXdetector}"
      long_name="{Axis label}" type="NX_INT[i]" primary="1?" axis="2">
    {Identifier for detector}?
  </detector_number>
  <data check_sum="{Integral of data as check of data       integrity}
      (NX_INT)?" signal="1"
      axes="[time_of_flight,detector_number,x_offset?,y_offset?]?"
      long_name="{Title of measurement}?" link="{absolute path to location in
      NXdetector}" units="number" type="NX_FLOAT[i,j,k?,l?]|NX_INT[i,j,k?,l?]">
    {Data values}?
  </data>
  <data_error units="number" link="{absolute path to location in NXdetector}"
      type="NX_FLOAT[i,j,k?,l?]|NX_INT[i,j,k?,l?]">
    {Data values}
  </data_error>
  <x_offset primary="1?" long_name="{Axis label}" link="{absolute path to
      location in NXdetector}" units="10^-3 meter|10^-2 meter"
      type="NX_FLOAT[k+1]" axis="3">
    {offset from the detector center in x-direction}?
  </x_offset>
  <y_offset primary="1?" long_name="{Axis label}" link="{absolute path to
      location in NXdetector}" units="10^-3 meter|10^-2 meter"
      type="NX_FLOAT[l+1]" axis="4">
    {offset from the detector center in the y-direction}?
  </y_offset>
  <distance axes="detector_number,x_offset?,y_offset?"
      type="NX_FLOAT[j,k?,l?]">
  </distance>
  <polar_angle axes="detector_number,x_offset?,y_offset?"
      type="NX_FLOAT[j,k?,l?]">
  </polar_angle>
```

```
  <azimuthal_angle axes="detector_number,x_offset?,y_offset?"
      type="NX_FLOAT[j,k?,l?]">
  </azimuthal_angle>
  <description type="NX_CHAR">
    {name/manufacturer/model/etc. information}?
  </description>
  <NXgeometry name="">
    {Position and orientation of detector element}?
  </NXgeometry>
  <translation units="centimeter" type="NX_FLOAT[2]">
    {translation normal to direct beam}?
  </translation>
  <solid_angle units="steradians" type="NX_FLOAT[i]">
    {Solid angle subtended by the detector at the sample}?
  </solid_angle>
  <x_pixelsize units="mili*metre" type="NX_FLOAT[i?]">
    {Size of each detector pixel. If it is scalar all pixels are the same
    size}?
  </x_pixelsize>
  <y_pixelsize units="mili*metre" type="NX_FLOAT[i?]">
    {Size of each detector pixel. If it is scalar all pixels are the same
    size}?
  </y_pixelsize>
  <dead_time type="NX_FLOAT[i]">
    {Detector dead time}?
  </dead_time>
  <hold_off units="micro.second" type="NX_FLOAT[i]">
    {Delay in detector registering an event}?
  </hold_off>
  <gas_pressure units="bars" type="NX_FLOAT[i]">
    {Detector gas pressure}?
  </gas_pressure>
  <detection_gas_path units="cm" type="NX_FLOAT">
    {maximum drift space dimension}?
  </detection_gas_path>
  <crate type="NX_INT[i]" local_name="{Equivalent local term}">
    {Crate number of detector}?
  </crate>
  <slot type="NX_INT[i]" local_name="{Equivalent local term}">
    {Slot number of detector}?
  </slot>
  <input type="NX_INT[i]" local_name="{Equivalent local term}">
    {Input number of detector}?
  </input>
  <type type="NX_CHAR">
    "He3 gas cylinder"|He3 PSD"|"He3 planar multidetector"| "He3 curved
    multidetector"| "multi-tube He3 PSD"|"BF3 gas"|"scintillator"|"fission
    chamber"?
  </type>
  <NXdata name="efficiency">
    {Efficiency of detector with respect to e.g. wavelength}?
  </NXdata>
  <calibration_date type="ISO8601">
    {date of last calibration (geometry and/or efficiency) measurements}?
  </calibration_date>
  <calibration_method type="NXnote">
    {summary of conversion of array data to pixels (e.g. polynomial
    approximations) and location of details of the calibrations}?
  </calibration_method>
</NXdetector>
```

Example 4-10. NXdisk_chopper.xml

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXdisc_chopper.xml
Editor:  Mark Koennecke
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of NXdisc_chopper.-->
<NXdisk_chopper name="{chopper_name}">
  <type type="NX_CHAR">
    {Chopper type single|contra_rotating_pair|synchro_pair}?
  </type>
  <rotation_speed units="rpm|hertz" type="NX_FLOAT">
    {chopper rotation speed}?
  </rotation_speed>
  <slits type="NX_INT">
    {Number of slits}
  </slits>
  <slit_angle units="degree" type="NX_FLOAT">
    {angular opening}
  </slit_angle>
  <pair_separation units="cm" type="NX_FLOAT">
    {disc spacing in direction of beam}?
  </pair_separation>
  <radius units="cm" type="NX_FLOAT">
    {radius to centre of slit}
  </radius>
  <slit_height units="cm" type="NX_FLOAT">
    {total slit height}
  </slit_height>
  <phase units="degree" type="NX_FLOAT">
    {chopper phase angle}?
  </phase>
  <ratio type="NX_INT">
    {pulse reduction factor of this chopper in relation to other
    choppers/fastest pulse in the instrument}?
  </ratio>
  <distance units="cm" type="NX_FLOAT">
    {Effective distance to the origin}?
  </distance>
  <wavelength_range units="nm" type="NX_FLOAT[2]">
    {low and high values of wavelength range transmitted}?
  </wavelength_range>
  <NXgeometry name="">
    {geometry of the disk chopper}?
  </NXgeometry>
</NXdisk_chopper>
```

Example 4-11. NXentry.xml

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXentry.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of the top-level NeXus group which contains all the data and
associated information that comprise a single measurement. It is mandatory that
there is at least one group of this type in the NeXus file.
-->
<NXentry name="{Entry Name}">
  <title>
    {Extended title for entry}?
  </title>
  <definition URL="{URL of DTD file}" version="{DTD version number}">
    {Name of entry DTD}?
```

```
  </definition>
  <start_time type="ISO8601">
    {Starting time of measurement}?
  </start_time>
  <end_time type="ISO8601">
    {Ending time of measurement}?
  </end_time>
  <duration units="seconds" type="NX_INT">
    {Duration of measurement}?
  </duration>
  <experiment_identifier type="NX_CHAR[]">
    {}?
  </experiment_identifier>
  <run_number type="NX_INT">
    {Number of run or scan stored in this entry}?
  </run_number>
  <run_cycle type="NX_CHAR[]">
    {}?
  </run_cycle>
  <program_name version="{Program version number}">
    {Name of program used to generate this file}?
  </program_name>
  <command_line>
    {Name of command line used to generate this file}?
  </command_line>
  <notes>
    {Notes describing entry}?
  </notes>
  <thumbnail type="NXnote" mime_type="{image/*}">
    {An small image that is representative of the entry.} {An example of this
    is a 640x480 jpeg image automatically produced by a low resolution plot of
    the NXdata.}?
  </thumbnail>
  <NXcharacterization name="">*
  </NXcharacterization>
  <NXuser name="{user}">
  </NXuser>
  <NXsample name="{sample}">
  </NXsample>
  <NXinstrument name="{Name of instrument}">
  </NXinstrument>
  <NXmonitor name="{Name of monitor}">
  </NXmonitor>
  <NXdata name="{Name of data block}">
  </NXdata>
  <NXprocess name="{Name of the process}">
  </NXprocess>
</NXentry>
```

Example 4-12. `NXenvironment.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXenvironment.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This class describes an external condition applied to the sample-->
<NXenvironment name="{Name of sample environment}">
  <name type="NX_CHAR">
    {Apparatus identification code/model number; e.g. OC100 011 }?
  </name>
  <short_name type="NX_CHAR">
    {Alternative short name, perhaps for dashboard display like a present
    Seblock name}?
  </short_name>
```

```
  <type type="NX_CHAR">
    {Type of apparatus. This could be the SE codes in scheduling database; e.g.
    OC/100}?
  </type>
  <description type="NX_CHAR">
    {Description of the apparatus; e.g. 100mm bore orange cryostat with Roots
    pump }?
  </description>
  <program type="NX_CHAR">
    {Program controlling the apparatus; e.g. LabView VI name}?
  </program>
  <position type="NXgeometry">
    {The position and orientation of the apparatus}?
  </position>
  <NXnote name="{name of note}">
    {Additional information, LabView logs, digital photographs, etc}*
  </NXnote>
  <NXsensor name="{name of sensor}">
  </NXsensor>
</NXenvironment>
```

Example 4-13. `NXevent_data.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXevent_data.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of NXevent_data-->
<NXevent_data name="">
  <time_of_flight units="10^n second" type="NX_INT[i]">
    {A list of time of flight for each event as it comes in. This list is for
    all pulses with information to attach to a particular pulse located in
    events_per_pulse.}?
  </time_of_flight>
  <pixel_number type="NX_INT[i]">
    {There will be extra information in the NXdetector to convert pixel_number
    to detector_number. This list is for all pulses with information to attach
    to a particular pulse located in events_per_pulse.}?
  </pixel_number>
  <pulse_time units="10^n second" type="NX_INT[j]" offset="{ISO8601}">
    {The time that each pulse started with respect to the offset}?
  </pulse_time>
  <events_per_pulse type="NX_INT[j]">
    {This connects the index "i" to the index "j". The jth element is the
    number of events in "i" that occured during the jth pulse.}?
  </events_per_pulse>
  <pulse_height units="" type="FLOAT[i,k?]">
    {If voltages from the ends of the detector are read out this is where they
    go. This list is for all events with information to attach to a particular
    pulse height. The information to attach to a particular pulse is located in
    events_per_pulse.}?
  </pulse_height>
</NXevent_data>
```

Example 4-14. `NXfermi_chopper.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXfermi_chopper.xml
Editor:  Ron Ghosh
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This is the description for a  Fermi chopper, possibly with curved slits.-->
<NXfermi_chopper name="chopper_name">
```

```
  <type type="NX_CHAR">
    {fchopper type}?
  </type>
  <rotation_speed units="rpm" type="NX_FLOAT">
    {chopper rotation speed}?
  </rotation_speed>
  <radius units="cm" type="NX_FLOAT">
    {radius of chopper}?
  </radius>
  <slit units="cm" type="NX_FLOAT">
    {width of an individual slit}?
  </slit>
  <r_slit units="cm" type="NX_FLOAT">
    {radius of curvature of slits}?
  </r_slit>
  <num type="NX_INT">
    {number of slits}?
  </num>
  <height units="cm" type="NX_FLOAT">
    {input beam height}?
  </height>
  <width units="cm" type="NX_FLOAT">
    {input beam width}?
  </width>
  <wavelength type="NX_FLOAT">
    {Wavelength transmitted by chopper}?
  </wavelength>
  <NXgeometry name="">
    {geometry of the fermi chopper}?
  </NXgeometry>
  <absorbing_material type="NX_CHAR">
    {absorbing material}?
  </absorbing_material>
  <transmitting_material type="NX_CHAR">
    {transmitting material}?
  </transmitting_material>
</NXfermi_chopper>
```

Example 4-15. `NXfilter.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXfilter.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template for specifying the state of band pass filters.-->
<NXfilter name="filter_name">
  <NXgeometry name="">
    {Geometry of the filter}?
  </NXgeometry>
  <description type="NX_CHAR">
    {"Beryllium" | "Pyrolytic Graphite" | "Graphite" | "Sapphire" | "Silicon" |
    "Supermirror"}?
  </description>
  <status type="NX_CHAR">
    {in | out}?
  </status>
  <transmission type="NXdata">
    {Wavelength transmission profile of filter}?
  </transmission>
  <temperature Units="Kelvin" type="NX_FLOAT">
    {average/nominal filter temperature}
  </temperature>
  <temperature_log type="NXlog">
    {Linked temperature_log for the filter}?
```

```
  </temperature_log>
  <sensor_type type="NXsensor">
    {Sensor(s)used to monitor the filter temperature}?
  </sensor_type>
  <unit_cell type="NX_FLOAT[n_comp,6])">
    {Unit cell parameters for single crystal filter(lengths and angles)}?
  </unit_cell>
  <unit_cell_volume units="Angstroms3" type="NX_FLOAT[n_comp]" rank="1">
    {Unit cell}?
  </unit_cell_volume>
  <orientation_matrix type="NX_FLOAT[n_comp,3,3]">
    {Orientation matrix of single crystal filter}?
  </orientation_matrix>
  <m_value type="NX_FLOAT">
    {m value of supermirror filter}
  </m_value>
  <substrate_material type="NX_FLOAT">
    {substrate material of supermirror filter}
  </substrate_material>
  <substrate_thickness type="NX_FLOAT">
    {substrate thickness of supermirror filter}
  </substrate_thickness>
  <coating_material type="NX_FLOAT">
    {coating material of supermirror filter}
  </coating_material>
  <substrate_roughness type="NX_FLOAT">
    {substrate rougness of supermirror filter}
  </substrate_roughness>
  <coating_roughness type="NX_FLOAT[nsurf]">
    {coating roughness of supermirror filter}
  </coating_roughness>
</NXfilter>
```

Example 4-16. `NXflipper.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXflipper.xml
Editor:  Nick Maliszewskyj <nickm@nist.gov>
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of a beamline spin flipper.-->
<NXflipper name="{Name of flipper}">
  <type type="NX_CHAR">
    {coil|current-sheet}?
  </type>
  <flip_turns type="NX_FLOAT">
    {Number of turns/cm in flipping field coils}?
  </flip_turns>
  <comp_turns type="NX_FLOAT">
    {Number of turns/cm in compensating field coils}?
  </comp_turns>
  <guide_turns type="NX_FLOAT">
    {Number of turns/cm in guide field coils}?
  </guide_turns>
  <flip_current units="amperes" type="NX_FLOAT">
    {Flipping field coil current in "on" state"}?
  </flip_current>
  <comp_current units="amperes" type="NX_FLOAT">
    {Compensating field coil current in "on" state"}?
  </comp_current>
  <guide_current units="amperes" type="NX_FLOAT">
    {Guide field coil current in "on" state"}?
  </guide_current>
  <thickness units="cm" type="NX_FLOAT">
    {thickness along path of neutron travel}?
```

```
  </thickness>
</NXflipper>
```

Example 4-17. NXgeometry.xml

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXgeometry.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This is the description for a general position of a component. Note that every
instance of an NXgeometry should be  named "geometry" so that it can be linked
to (linked items  must share the same name in HDF)
-->
<NXgeometry name="">
  <NXshape name="{shape}">
    {shape/size information of component}?
  </NXshape>
  <NXtranslation name="{translation}">
    {translation of component}?
  </NXtranslation>
  <NXorientation name="{orientation}">
    {orientation of component}?
  </NXorientation>
  <description type="NX_CHAR">
    {Optional description/label}{Probably only present if we are an additional
    reference point for components rather than the location of a real
    component}?
  </description>
  <component_index type="NX_INT">
    {Position of the component along the beam path.}{The sample is at 0,
    components upstream have negative component_index, components downstream
    have positive component_index.}?
  </component_index>
</NXgeometry>
```

Example 4-18. NXguide.xml

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXguide.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of NXguide-->
<NXguide name="">
  <NXgeometry name="">
  </NXgeometry>
  <description type="NX_CHAR">
  </description>
  <incident_angle type="NX_FLOAT">
  </incident_angle>
  <reflectivity type="NXdata">
    {Reflectivity as function of wavelength [nsurf,i]}
  </reflectivity>
  <bend_angle_x type="NX_FLOAT">
  </bend_angle_x>
  <bend_angle_y type="NX_FLOAT">
  </bend_angle_y>
  <interior_atmosphere type="NX_CHAR">
    "vacuum"|"helium"|"argon"
  </interior_atmosphere>
  <external_material type="NX_CHAR">
    {external material outside substrate}
  </external_material>
```

```
  <m_value type="NX_FLOAT[nsurf]">
  </m_value>
  <substrate_material type="NX_FLOAT[nsurf]">
  </substrate_material>
  <substrate_thickness type="NX_FLOAT[nsurf]">
  </substrate_thickness>
  <coating_material type="NX_FLOAT[nsurf]">
  </coating_material>
  <substrate_roughness type="NX_FLOAT[nsurf]">
  </substrate_roughness>
  <coating_roughness type="NX_FLOAT[nsurf]">
  </coating_roughness>
  <number_sections type="NX_INT">
    {number of substrate sections}
  </number_sections>
</NXguide>
```

Example 4-19. `NXinstrument.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXinstrument.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of instrument descriptions comprising various beamline  components.
Each component will also be a NeXus group defined by its  distance from the
sample. Negative distances represent beamline  components that are before the
sample while positive distances  represent components that are after the
sample. This device allows  the unique identification of beamline components in
a way that is  valid for both reactor and pulsed instrumentation.
-->
<NXinstrument name="{Name of instrument}">
  <name short_name="{abbreviated name of instrument}">
    {Name of instrument}?
  </name>
  <NXsource name="{Name of facility}">
  </NXsource>
  <NXdisk_chopper name="{Name of chopper}">
  </NXdisk_chopper>
  <NXfermi_chopper name="">
  </NXfermi_chopper>
  <NXvelocity_selector name="">
  </NXvelocity_selector>
  <NXguide name="">
  </NXguide>
  <NXcrystal name="{Name of crystal monochromator or analyzer}">
  </NXcrystal>
  <NXaperture name="{Name of beamline aperture}">
  </NXaperture>
  <NXfilter name="">
  </NXfilter>
  <NXcollimator name="{Name of collimator}">
  </NXcollimator>
  <NXattenuator name="{Name of beam attenuator}">
  </NXattenuator>
  <NXpolarizer name="{Name of beam polarizer}">
  </NXpolarizer>
  <NXflipper name="{Name of beam polarization flipper}">
  </NXflipper>
  <NXmirror name="{Name of beam guide mirror}">
  </NXmirror>
  <NXdetector name="{Name of detector, bank of detectors, or multidetector}">
  </NXdetector>
  <NXbeam_stop name="">
  </NXbeam_stop>
```

```
</NXinstrument>
```

Example 4-20. `NXlog.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXlog.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Definition of logged information, i.e. information monitored during the run.
They contain the logged values and the times at which they were measured as
elapsed time since a starting time recorded in ISO8601 format. This method of
storing logged data helps to distinguish instances in which a variable is a
dimension scale of the data, in which case it is stored in an NXdata group, and
instances in which it is logged during the run, when it should be stored in an
NXlog group.
-->
<NXlog name="{Name of logged measurements}">
  <time units="" start="{ISO8601}" type="NX_FLOAT">
    {Time of logged entry}{The times are relative to the "start" attribute and
    in the units specified in the "units" attribute.}
  </time>
  <value units="{units of logged value}" type="NX_FLOAT|NX_INT">
    {Array of logged value, such as temperature}
  </value>
  <raw_value units="{units of raw values}" type="NX_FLOAT|NX_INT">
    {Array of raw information, such as voltage on a thermocouple}?
  </raw_value>
  <description type="NX_CHAR">
    {Description of logged value}?
  </description>
  <average_value units="" type="NX_FLOAT">
  </average_value>
  <average_value_error units="" type="NX_FLOAT">
    {standard deviation of average_value}?
  </average_value_error>
  <minimum_value units="" type="NX_FLOAT">
  </minimum_value>
  <maximum_value units="" type="NX_FLOAT">
  </maximum_value>
  <duration units="" type="NX_FLOAT">
    {Total time log was taken}?
  </duration>
</NXlog>
```

Example 4-21. `NXmirror.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXmirror.xml
Editor:
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of a beamline supermirror.-->
<NXmirror name="{Name of supermirror}">
  <NXgeometry name="">
  </NXgeometry>
  <description type="NX_CHAR">
  </description>
  <incident_angle type="NX_FLOAT">
  </incident_angle>
  <reflectivity type="NXdata">
    {Reflectivity as function of wavelength}
  </reflectivity>
  <bend_angle_x type="NX_FLOAT">
```

```
  </bend_angle_x>
  <bend_angle_y type="NX_FLOAT">
  </bend_angle_y>
  <interior_atmosphere type="NX_CHAR">
    "vacuum"|"helium"|"argon"
  </interior_atmosphere>
  <external_material type="NX_CHAR">
    {external material outside substrate}
  </external_material>
  <m_value type="NX_FLOAT">
  </m_value>
  <substrate_material type="NX_CHAR">
  </substrate_material>
  <substrate_thickness type="NX_FLOAT">
  </substrate_thickness>
  <coating_material type="NX_CHAR">
  </coating_material>
  <substrate_roughness type="NX_FLOAT">
  </substrate_roughness>
  <coating_roughness type="NX_FLOAT">
  </coating_roughness>
</NXmirror>
```

Example 4-22. `NXmoderator.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXmoderator.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This is the description for a general moderator-->
<NXmoderator name="{Name of moderator}">
  <NXgeometry name="">
    {"Engineering" position of moderator}?
  </NXgeometry>
  <distance type="NX_FLOAT">
    {Effective distance as seen by measuring radiation}?
  </distance>
  <type type="NX_CHAR">
    { "H20" | "D20" | "Liquid H2" | "Liquid CH4" | "Liquid D2" | "Solid D2" |
    "C" |"Solid CH4" | "Solid H2"}?
  </type>
  <poison_depth units="cm" type="NX_FLOAT">
    {Poison depth}?
  </poison_depth>
  <coupled type="NX_BOOLEAN">
    {whether the moderator is coupled}?
  </coupled>
  <poison_material type="NX_CHAR">
    { Gd | Cd |...}
  </poison_material>
  <temperature Units="Kelvin" type="NX_FLOAT">
    {average/nominal moderator temperature}
  </temperature>
  <temperature_log type="NXlog">
    {log file of moderator temperature}
  </temperature_log>
  <pulse_shape type="NXdata">
    {moderator pulse shape}
  </pulse_shape>
</NXmoderator>
```

Example 4-23. `NXmonitor.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXmonitor.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of monitor data. It is similar to the NXdata groups containing monitor
data and its associated dimension scale, e.g. time_of_flight or wavelength in
pulsed neutron instruments. However, it may also include integrals, or scalar
monitor counts, which are often used in both in both pulsed and steady-state
instrumentation.
-->
<NXmonitor name="{Name of monitor}">
  <mode type="NX_CHAR">
    "monitor"|"timer"?
  </mode>
  <preset type="NX_FLOAT">
    {preset value for time or monitor}?
  </preset>
  <distance units="m" type="NX_FLOAT">
    {Distance of monitor from sample}?
  </distance>
  <range type="NX_FLOAT[2]">
    {Time-of-flight range over which the integral was calculated}?
  </range>
  <integral units="" type="NX_FLOAT">
    {Total integral monitor counts}?
  </integral>
  <integral_log units="" type="NXlog">
    {Time variation of monitor counts}?
  </integral_log>
  <type type="NX_CHAR">
    "Fission Chamber"|"Scintillator"?
  </type>
  <time_of_flight units="microseconds" type="NX_FLOAT[i]">
    {Time-of-flight}?
  </time_of_flight>
  <efficiency type="NX_FLOAT[i]">
    {Monitor efficiency}?
  </efficiency>
  <data units="" signal="1" axes="" type="NX_INT[i]">
    {Monitor data}?
  </data>
  <sampled_fraction units="dimensionless" type="NX_FLOAT">
    {Proportion of incident beam sampled by the monitor (0<x<1)}
  </sampled_fraction>
  <NXgeometry name="">
    {Geometry of the monitor}?
  </NXgeometry>
</NXmonitor>
```

Example 4-24. `NXnote.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXnote.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This class can be used to store additional information in a NeXus file e.g.
pictures, movies, audio, additonal text logs
-->
<NXnote name="{name of note}">
  <author type="NX_CHAR">
    {Author or creator of note}?
```

```
  </author>
  <date type="ISO8601">
    {Date note created/added}?
  </date>
  <type type="NX_CHAR">
    {Mime content type of note data field e.g. image/jpeg, text/plain,
    text/html}?
  </type>
  <file_name type="NX_CHAR">
    {Name of original file name if note was read from an external source}?
  </file_name>
  <description type="NX_CHAR">
    {Title of an image or other details of the note}?
  </description>
  <data type="NX_BINARY">
    {Binary note data - if text, line terminator is \r\n.}?
  </data>
</NXnote>
```

Example 4-25. `NXorientation.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXorientation.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This is the description for a general orientation of a component - it is used
by the NXgeometry class
-->
<NXorientation name="{name of orientation}">
  <NXgeometry name="">
    {Link to another object if we are using relative positioning, else absent}?
  </NXgeometry>
  <value type="NX_FLOAT[numobj,6]">
    {The orientation information is stored as direction cosines.}{The direction
    cosines will be between the local coordinate directions and the reference
    directions (to origin or relative NXgeometry). Calling the local unit
    vectors (x',y',z') and the reference unit vectors (x,y,z) the six numbers
    will be [x' dot x, x' dot y, x' dot z, y' dot x, y' dot y, y' dot z] where
    "dot" is the scalar dot product (cosine of the angle between the unit
    vectors). The unit vectors in both the local and reference coordinates are
    right-handed and orthonormal.}?
  </value>
</NXorientation>
```

Example 4-26. `NXprocess.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXlog.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template for a process.-->
<NXprocess name="">
  <NXnote name="{numbered name to allow for ordering steps}">
    {}{The note will contain information about how the data was processed. The
    contents of the note can be anything that the processing code can
    understand, or simple text.}+
  </NXnote>
</NXprocess>
```

Example 4-27. `NXroot.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXroot.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Definition of the root NeXus group.-->
<NXroot NeXus_version="{Version of NeXus API used in writing the file}"
    HDF5_version="?" creator="{facility or program where file originated}?"
    file_name="{File name of original NeXus file}" HDF_version="?"
    file_time="{Date and time of file creation}" file_update_time="{Date and
    time of last file change at close}">
  <NXentry name="{entry name}">
  </NXentry>
</NXroot>
```

Example 4-28. `NXsample.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXsample.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of the state of the sample. This could include scanned variables that
are associated with one of the data dimensions, e.g. the magnetic field, or
logged data, e.g. monitored temperature vs elapsed time.
-->
<NXsample name="{name of the sample}">
  <name type="NX_CHAR">
    {Descriptive name of sample}?
  </name>
  <chemical_formula type="NX_CHAR">
    {The chemical formula specified using CIF conventions.}{Abbreviated version
    of CIF standard: 1. Only recognized element symbols may be used. 2. Each
    element symbol is followed by a 'count' number. A count of '1' may be
    omitted. 3. A space or parenthesis must separate each cluster of (element
    symbol + count). 4. Where a group of elements is enclosed in parentheses,
    the multiplier for the group must follow the closing parentheses. That is,
    all element and group multipliers are assumed to be printed as subscripted
    numbers. 5. Unless the elements are ordered in a manner that corresponds to
    their chemical structure, the order of the elements within any group or
    moiety depends on whether or not carbon is present. If carbon is present,
    the order should be: C, then H, then the other elements in alphabetical
    order of their symbol. If carbon is not present, the elements are listed
    purely in alphabetic order of their symbol. This is the 'Hill' system used
    by Chemical Abstracts.}?
  </chemical_formula>
  <temperature type="NX_FLOAT[:]">
    {Sample temperature. This could be a scanned variable}?
  </temperature>
  <electric_field direction="x|y|z" type="NX_FLOAT[:]">
    {Applied electric field}*
  </electric_field>
  <magnetic_field direction="x|y|z" type="NX_FLOAT[:]">
    {Applied magnetic field}*
  </magnetic_field>
  <stress_field direction="x|y|z" type="NX_FLOAT[:]">
    {External stress}*
  </stress_field>
  <pressure type="NX_FLOAT[:]">
    {Applied pressure}?
  </pressure>
  <changer_position type="NX_INT">
    {Sample changer position}?
```

```
</changer_position>
<unit_cell type="NX_FLOAT[n_comp,6])">
  {Unit cell parameters (lengths and angles)}?
</unit_cell>
<unit_cell_volume units="Angstroms3" type="NX_FLOAT[n_comp]" rank="1">
  {Volume of the unit cell}?
</unit_cell_volume>
<sample_orientation units="degree" type="NX_FLOAT[3]">
  {This will follow the Busing and Levy convention from Acta.Crysta v22, p457
  (1967)}?
</sample_orientation>
<orientation_matrix type="NX_FLOAT[n_comp,3,3]">
  {Orientation matrix of single crystal sample}{The is the orientation matrix
  using Busing-Levy convention}?
</orientation_matrix>
<mass units="g" type="NX_FLOAT[n_comp]">
  {Mass of sample}?
</mass>
<density units="g cm-3" type="NX_FLOAT[n_comp]">
  {Density of sample}?
</density>
<relative_molecular_mass type="NX_FLOAT[n_comp]">
  {Relative Molecular Mass of sample}?
</relative_molecular_mass>
<type type="NX_CHAR">
  { sample | sample+can | can | calibration sample | normalisation sample |
  simulated data | none | sample environment }?
</type>
<situation type="NX_CHAR">
  { air | vacuum | inert atmosphere | oxidising atmosphere | reducing
  atmosphere | sealed can | other }{The atmosphere will be one of the
  components, which is where its details will be stored; the relevant
  components will be indicated by the entry in the sample_component member.}?
</situation>
<description type="NX_CHAR">
  {Description of the sample}?
</description>
<preparation_date type="ISO8601">
  {Date of preparation of the sample}?
</preparation_date>
<geometry type="NXgeometry">
  {The position and orientation of the center of mass of the sample}?
</geometry>
<beam type="NXbeam">
  {Details of beam incident on sample - used to calculate sample/beam
  interaction point}?
</beam>
<component type="NX_CHAR[n_comp]">
  {Details of the component of the sample and/or can}?
</component>
<sample_component type="NX_CHAR[n_comp]">
  { What type of component we are "sample | can | atmosphere | kit" }?
</sample_component>
<concentration units="g.cm-3" type="NX_FLOAT[n_comp]">
  {Concentration of each component}?
</concentration>
<volume_fraction type="NX_FLOAT[n_comp]">
  {Volume fraction of each component}?
</volume_fraction>
<scattering_length_density type="NX_FLOAT[n_comp]">
  {Scattering length density of each component (cm-2)}?
</scattering_length_density>
<unit_cell_class type="NX_CHAR[n_comp]">
  { In case it is all we know and we want to record it "cubic | tetragonal |
  orthorhombic | monoclinic | triclinic" }?
</unit_cell_class>
```

```
  <unit_cell_group type="NX_CHAR[n_comp]">
    {Crystallographic point or space group}?
  </unit_cell_group>
  <path_length type="NX_FLOAT">
    {Path length through sample/can for simple case when it does not vary with
    scattering direction}?
  </path_length>
  <path_length_window type="NX_FLOAT">
    {Thickness of a beam entry/exit window on the can (mm) - assumed same for
    entry and exit}?
  </path_length_window>
  <transmission type="NXdata">
    {As a function of Wavelength}?
  </transmission>
  <temperature_log type="NXlog">
    {temperature_log.value is a link to e.g.
    temperature_env.sensor1.value_log.value}?
  </temperature_log>
  <temperature_env type="NXenvironment">
    {Additional sample environment information}?
  </temperature_env>
  <magnetic_field_log type="NXlog">
    {magnetic_field_log.value is a link to e.g.
    magnetic_field_env.sensor1.value_log.value}?
  </magnetic_field_log>
  <magnetic_field_env type="NXenvironment">
    {Additional sample environment information}?
  </magnetic_field_env>
  <external_DAC type="NX_FLOAT">
    {value sent to user's sample setup}?
  </external_DAC>
  <external_ADC type="NXlog">
    {logged value (or logic state) read from user's setup}?
  </external_ADC>
  <short_title type="NX_CHAR">
    {20 character fixed length sample description for legends}?
  </short_title>
</NXsample>
```

Example 4-29. `NXsensor.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXsensor.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This class describes a sensor used to monitor an external condition - the
condition itself in described in NXenvironment
-->
<NXsensor name="{Name of sensor}">
  <model type="NX_CHAR">
    {Sensor identification code/model number}?
  </model>
  <name type="NX_CHAR">
    {Name for the sensor}?
  </name>
  <short_name type="NX_CHAR">
    {Short name of sensor used e.g. on monitor display program}?
  </short_name>
  <attached_to type="NX_CHAR">
    { where sensor is attached to ("sample" | "can") }?
  </attached_to>
  <NXgeometry name="">
    {Defines the axes for logged vector quantities if they are not the global
    instrument axes}?
```

```
  </NXgeometry>
  <measurement type="NX_CHAR">
    { what we measure "temperature | pH | magnetic_field | electric field |
    conductivity | resistance | voltage | pressure | flow | stress | strain |
    shear | surface_pressure" }?
  </measurement>
  <type type="NX_CHAR">
    { The type of hardware we use for the measurement e.g. Temperature: "J | K
    | T | E | R | S | Pt100 | Rh/Fe" pH: "Hg/Hg2Cl2 | Ag/AgCl | ISFET" Ion
    selective electrode: "specify species; e.g. Ca2+" Magnetic field: "Hall"
    Surface pressure: "wilhelmy plate" }?
  </type>
  <run_control type="NX_BOOLEAN">
    { Is data collection controlled/synchronised to this quantity: 1=no, 0=to
    "value", 1=to "value_deriv1" etc.}?
  </run_control>
  <high_trip_value units="{}" type="NX_FLOAT">
    {Upper control bound of sensor reading if using run_control}?
  </high_trip_value>
  <low_trip_value units="{}" type="NX_FLOAT">
    {Lower control bound of sensor reading if using run_control}?
  </low_trip_value>
  <value units="{}" type="NX_FLOAT[n]">
    {nominal setpoint or average value - need [n] as may be a vector}?
  </value>
  <value_deriv1 units="{}" type="NX_FLOAT[n]">
    {Nominal/average first derivative of value e.g. strain rate - need [n] as
    may be a vector}?
  </value_deriv1>
  <value_deriv2 units="{}" type="NX_FLOAT[n]">
    {Nominal/average second derivative of value - need [n] as may be a vector}?
  </value_deriv2>
  <value_log type="NXlog">
    {Time history of sensor readings}?
  </value_log>
  <value_deriv1_log type="NXlog">
    {Time history of sensor readings}?
  </value_deriv1_log>
  <value_deriv2_log type="NXlog">
    {Time history of sensor readings}?
  </value_deriv2_log>
  <external_field_brief type="NX_CHAR">
    { along beam | across beam | transverse | solenoidal | flow shear gradient
    | flow vorticity }?
  </external_field_brief>
  <external_field_full type="NXorientation">
    {For complex external fields not satisfied by External_field_brief}?
  </external_field_full>
</NXsensor>
```

Example 4-30. `NXshape.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXshape.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This is the description of the general shape and size of a  component, which
may be made up of "numobj" separate elements -  it is used by the
NXgeometry.xml class
-->
<NXshape name="{name of shape}">
  <shape type="NX_CHAR">
    {"nxcylinder", "nxbox", "nxsphere", ...}?
  </shape>
```

```
  <size units="meter" type="NX_FLOAT[numobj,nshapepar]">
    {physical extent of the object along its local axes (after NXorientation)
    with the center of mass at the local origin (after NXtranslate).}{The
    meaning and location of these axes will vary according to the value of the
    "shape" variable. nshapepar defines how many parameters. For the
    "nxcylinder" type the paramters are (diameter,height). For the "nxbox" type
    the parameters are (length,width,height). For the "nxsphere" type the
    parameters are (diameter).}?
  </size>
</NXshape>
```

Example 4-31. `NXsource.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXsource.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of the neutron or x-ray source, insertion devices and/or moderators.-->
<NXsource name="source">
  <distance units="m" type="NX_FLOAT">
    {Effective distance from sample}{Distance as seen by radiation from sample.
    This number should be negative to signify that it is upstream of the
    sample.}?
  </distance>
  <name type="NX_CHAR">
    {Name of source}?
  </name>
  <type type="NX_CHAR">
    "Spallation Neutron Source"|"Pulsed Reactor Neutron Source"|"Reactor
    Neutron Source"|"Synchrotron X-ray Source"|"Pulsed Muon Source"|"Rotating
    Anode X-ray"|Fixed Tube X-ray"?
  </type>
  <probe type="NX_CHAR">
    neutron|x-ray|muon|electron?
  </probe>
  <power units="MW" type="NX_FLOAT">
    {Source power}?
  </power>
  <current units="microamps" type="NX_FLOAT">
    {Accelerator proton current}?
  </current>
  <voltage units="MeV" type="NX_FLOAT">
    {Accelerator proton voltage}?
  </voltage>
  <frequency units="Hz" type="NX_FLOAT">
    {Frequency of pulsed source}?
  </frequency>
  <period units="microseconds" type="NX_FLOAT">
    {Period of pulsed source}?
  </period>
  <target_material type="NX_CHAR">
    {Pulsed source target material}
    {"Ta"|"W"|"depleted_U"|"enriched_U"|"Hg"|"Pb"|"C"}?
  </target_material>
  <notes type="NX_CHAR">
    {any source/facility related messages/events that occurred during the
    experiment}?
  </notes>
  <pulse_width units="micro.second" type="NX_FLOAT">
    {width of source pulse}?
  </pulse_width>
  <pulse_shape type="NXdata">
    {source pulse shape}?
  </pulse_shape>
```

```
  <NXgeometry name="">
    {"Engineering" location of source}?
  </NXgeometry>
</NXsource>
```

Example 4-32. `NXtranslation.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXtranslation.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This is the description for the general spatial location of a component - it is
used by the NXgeometry.xml class
-->
<NXtranslation name="{name of translation}">
  <NXgeometry name="{geometry}">
    {Link to other object if we are relative , else absent}?
  </NXgeometry>
  <distances units="" type="NX_FLOAT[numobj,3]">
    {(x,y,z)}{This field and the angle field describe the position of a
    component. For absolute position, the origin is the scattering center
    (where a perfectly aligned sample would be) with the z-axis pointing
    downstream and the y-axis pointing gravitationally up. For a relative
    position the NXtranslation is taken into account before the NXorientation.
    The axes are right-handed and orthonormal.}?
  </distances>
</NXtranslation>
```

Example 4-33. `NXuser.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXuser.xml
Editor:  NIAC
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Template of user's contact information.  The format allows more than  one user
with the same affiliation and contact information, but a  second NXuser group
should be used if they have different  affiliations, etc.
-->
<NXuser name="">
  <name type="NX_CHAR">
    {Name of user responsible for this entry}?
  </name>
  <role type="NX_CHAR">
    {role of user responsible for this entry}{Suggested roles are
    "local_contact", "principal_investigator", and "proposer"}?
  </role>
  <affiliation type="NX_CHAR">
    {Affiliation of user}?
  </affiliation>
  <address type="NX_CHAR">
    {Address of user}?
  </address>
  <telephone_number type="NX_CHAR">
    {Telephone number of user}?
  </telephone_number>
  <fax_number type="NX_CHAR">
    {Fax number of user}?
  </fax_number>
  <email type="NX_CHAR">
    {Email of user}?
  </email>
  <facility_user_id type="NX_CHAR">
```

```
    {facility based unique identifier for this person e.g. their identification
    code on the facility address/contact database}?
  </facility_user_id>
</NXuser>
```

Example 4-34. `NXvelocity_selector.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXvelocity_selector.xml
Editor:  Ron Ghosh
$Id: base_classes.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

This is the description for a velocity selector. Proposed by: Ron Ghosh-->
<NXvelocity_selector name="selector_name">
  <type type="NX_CHAR">
    {Vselect type}?
  </type>
  <rotation_speed units="rpm" type="NX_FLOAT">
    {selector rotation speed}?
  </rotation_speed>
  <radius units="cm" type="NX_FLOAT">
    {radieus at beam centre}?
  </radius>
  <spwidth units="cm" type="NX_FLOAT">
    {spoke width at beam centre}?
  </spwidth>
  <length units="cm" type="NX_FLOAT">
    {rotor length}?
  </length>
  <num type="NX_INT">
    {number of spokes/lamella}?
  </num>
  <twist units="degrees" type="NX_FLOAT">
    {twist angle along axis}?
  </twist>
  <table units="degrees" type="NX_FLOAT">
    {offset vertical angle}?
  </table>
  <height units="cm" type="NX_FLOAT">
    {input beam height}?
  </height>
  <width units="cm" type="NX_FLOAT">
    {input beam width}?
  </width>
  <wavelength units="nm" type="NX_FLOAT">
    {wavelength}
  </wavelength>
  <wavelength_spread type="NX_FLOAT">
    {% deviation FWHM /Wavelength}?
  </wavelength_spread>
  <NXgeometry name="">
    {geometry of the velocity selector}?
  </NXgeometry>
</NXvelocity_selector>
```

Chapter 5. Definitions

Monochromatic Reflectometry

Example 5-1. `monoref.xml`

```
<!--
Instrument Definition for Monochromatic Source Reflectometers.

Editor: Paul Kienzle <pkienzle@nist.gov>
Mangled_by: Nick Maliszewskyj <nickm@nist.gov>
$Id: monoref.docbook 500 2005-06-14 16:50:35Z pfp $

See http://www.neutron.anl.gov:8080/NeXus/5 for component definitions.-->
<NXentry>
  <definition URL="http://www.nexus.anl.gov/instruments/xml/monoref.xml"
      version="1.0">
    MONOREF
  </definition>
  <start_time type="ISO8601">
  </start_time>
  <tag polarization="++|+-|-+|--|+|-?" magnetic_field="NX_FLOAT?"
      userfield="..." scan="slit|background+|background-|specular|rock|...">
    {Empty.}
    <!--
    *** HDF may not allow empty blocks, so maybe a single integer
    *** to distinguish the various conditions.
    -->
  </tag>
  <NXsample>
    <rotation_angle units="degrees" type="NX_FLOAT[i]">
    </rotation_angle>
  </NXsample>
  <!--
  *** link to spectrum measurement for intensity vs. wavelength
  *** for a given slit setting

  *** warning: beam profile is not regular, but this effect is
  *** accomodated in the spectrum measurement
  -->
  <NXinstrument>
    <!-- wavelength selection -->
    <NXcrystal name="monochromator">
      <wavelength units="Angstroms" type="NX_FLOAT">
        { Wavelength of beam exiting component - take this as wavelength of
        beam incident on samples }
      </wavelength>
      <wavelength_spread type="NX_FLOAT">
        {% deviation FWHM /Wavelength - need this to compute resolution}
      </wavelength_spread>
    </NXcrystal>
    <!-- collimation -->
    <NXaperture name="pre[sample|detector]_slit[|y]#">
      <opening units="mm" filter="Qx|Qy" type="NX_FLOAT">
        <!-- *** This is not part of standard NXaperture -->
        { Slit opening; this is a scan parameter, so cannot be recorded as part
        of the aperture size. }
      </opening>
      <NXgeometry name="geometry">
        <NXtranslation name="translation">
          <distances units="mm" type="NX_FLOAT">
            { Location of slit along beamline (midway between slits if slits
            are not coplanar). This is required to compute instrument
```

```
          resolution. }
        </distances>
        <angles>
          { Only need angles if slits are not centered on beam wrt sample. }?
        </angles>
      </NXtranslation>
      <NXshape name="shape">
        <type type="NX_CHAR">
          nxslit
        </type>
        <size units="mm" type="NX_FLOAT[2]">
          { size[1] is thickness of material (actual slits can be more
          complicated in practice, being composed of different materials that
          are not aligned, but these can be simulated with sets of slits).
          size[2] is zero if slits are coplanar, otherwise it is the distance
          between slits. each half of the slit is considered to be a
          semi-infinite plane cutting neutrons in Qx if they are in the
          scattering plane, or Qy if they are normal to the scattering plane
          as defined by
          <opening filter="...">
          </opening>
          above. }
        </size>
      </NXshape>
    </NXgeometry>
  </NXaperture>
  <!--
  The polarizer-flipper-guidefield combination selects polarization vectors
  in and out of the sample. A number of scans are required to tune the
  instrument so that polarization is either 'up' or 'down' on the sample. On
  correctly tuned instruments the polarization angle selected should be
  recorded by the flipper using polar_angle relative to the surface (0/180
  for +/-, or with out of plane polarization, 90/270 for +/-). The
  polarization efficiency must be determined from a spectrum scan and the
  appropriate correction applied to the data. Raw values from the instrument,
  such as time dependent field applied to flipper coils or current on the
  current sheet can be recorded for specialized reduction programs which know
  how to handle them.
  -->
  <NXpolarizer name="presample_polarizer">
  </NXpolarizer>
  <NXflipper name="presample_flipper">
    <polar_angle units="degree" type="NX_FLOAT">
    </polar_angle>
  </NXflipper>
  <NXpolarizer name="predetector_polarizer">
  </NXpolarizer>
  <NXflipper name="predetector_flipper">
    <polar_angle units="degree" type="NX_FLOAT">
    </polar_angle>
  </NXflipper>
  <!-- detector may be protected by an attenuator and/or a beam stop -->
  <NXattenuator>
    <attenuator_transmission type="NX_FLOAT">
      { The nominal amount of the beam that gets through (transmitted
      intensity)/(incident intensity) }
    </attenuator_transmission>
  </NXattenuator>
  <NXbeam_stop name="stop">
    ? { Need all fields so that we can calculate shadow of beam stop on
    detector. }
  </NXbeam_stop>
  <NXdetector name="detector">
    <!--
    polar_angle and azimuthal_angle define the location of the detector
    relative to the beamzero
```

```
    -->
    <distance units="mm" type="NX_FLOAT">
      { distance from sample }
    </distance>
    <translation units="centimeter" type="NX_FLOAT[2]">
      { translation normal to direct beam }?
    </translation>
    <polar_angle units="degrees" type="NX_FLOAT[i]">
      { angular position of detector relative to beamzero through sample --
      known to practitioners as "A4" or "two theta" }
    </polar_angle>
    <azimuthal_angle units="degrees" type="NX_FLOAT">
      { Indicate sense of scattering: 0 is front surface of sample, 180 is
      back surface of sample. If 180, change the sign of the reflected angle
      in the data. It is also possible for the beam to enter the substrate
      from the side and reflect off the back surface of a film, in which case
      negative angles can be interpreted as inverting the scattering length
      density profile of the film (after accounting for absorption in the
      substrate. }
    </azimuthal_angle>
    <x_offset units="mm" type="NX_FLOAT[j]">
      {pixel edges in x}?
    </x_offset>
    <y_offset units="mm" type="NX_FLOAT[k]">
      {pixel edges in x}?
    </y_offset>
    <counts signal="1" axes="x_offset?,y_offset?,polar_angle"
        type="NX_INT[k,? j,? i]">
      { raw detector counts }
    </counts>
    <!--
    *** Raw counts are meaningless to the user if they are counting
    *** against detector since all values will be the same. The data
    *** only become meaningful when divided by counting time or monitor
    *** as specified by the ratio field. The generic plotting program
    *** will need to sort this out.

    *** In general, n-D data should contain k-D summary statistics
    *** for all 0&lt;=k&lt;n. That way a really dumb plotting program can
    *** still display info from a 3-D result as a line.

    *** Some control systems have data windows. Windows may be
    *** defined in terms of pixel ranges or in terms of
    *** theta_in-theta_out relationships (i.e., Qx). Each scan
    *** point should have start/end positions and summary
    *** statistics for every window defined. Window size may be
    *** dynamic. Store windows in separate data blocks.
    -->
  </NXdetector>
</NXinstrument>
<NXmonitor name="monitor">
  <mode type="NX_CHAR">
    monitor
  </mode>
  <preset type="NX_FLOAT">
    { preset value for monitor }?
  </preset>
  <data units="counts" type="NX_INT[i]">
    { record of monitor counts }?
  </data>
  <efficiency type="Nxdata">
    { Monitor efficiency as a function of wavelength }?
  </efficiency>
  <sampled_fraction units="dimensionless" type="NX_FLOAT">
    { Proportion of incident beam sampled by the monitor }
  </sampled_fraction>
```

```
  </NXmonitor>
  <NXmonitor name="timer">
    <mode type="NX_CHAR">
      timer
    </mode>
    <preset type="NX_FLOAT">
      { preset value for timer }?
    </preset>
    <data units="seconds" type="NX_INT[i]">
      { record of times for individual points }?
    </data>
  </NXmonitor>
  <NXdata>
    <attenuator_transmission NAPIlink="NXentry/NXinstrument/NXattenuator">
    </attenuator_transmission>
    <theta NAPIlink="NXentry/NXsample/rotation_angle">
    </theta>
    <twotheta NAPIlink="NXentry/detector/polar_angle">
    </twotheta>
    <presample_slit1 NAPIlink="NXentry/presample_slit1/opening">
    </presample_slit1>
    <presample_slit2 NAPIlink="NXentry/presample_slit2/opening">
    </presample_slit2>
    <predetector_slit1 NAPIlink="NXentry/predetector_slit1/opening">
    </predetector_slit1>
    <predetector_slit2 NAPIlink="NXentry/predetector_slit2/opening">
    </predetector_slit2>
    <counts NAPIlink="NXentry/detector/counts">
    </counts>
    <count_start units="second" type="NX_FLOAT[i]">
      <!-- probably shouldn’t store any real data here, but where else? -->
      { start time of each measurement point relative to start time of entry. }
    </count_start>
    <timer NAPIlink="NXentry/timer/data">
    </timer>
    <monitor NAPIlink="NXentry/monitor/data">
    </monitor>
  </NXdata>
  <NXlog name="??">
    { Various logs for temperature, field, etc. which are assumed to be
    constant over the duration of the run. The reduction program should be able
    to display their values on a parallel graph. Note that logs are not
    necessarily sampled synchronously with the data points. }*
  </NXlog>
</NXentry>
<!--
Reflectometry requires several different data scans.  After reduction
it will all be reduced to normalized reflected intensity as a function
of Q but how it gets there depends on the nature of the instrument and
how the data are taken.

A reflectometry data point is parameterized by theta_in (the angle of
incidence relative to the sample surface) and theta_out (the angle of
the detector relative to the beam).  The reflectivity is is a count of the
neutrons reflected in the specular condition (theta_out = theta_in)
minus the background measured in an off-specular condition and
divided by the number of neutrons incident on the sample.

You also need to know the wavelength of the source in order to convert
from real space to reciprocal space and to calculate absorption in models
of the reflected signal.

The common scan types are alignment scans, intensity scans,
specular scans, and +/- offset background scans. With polarized beam
these scan types are repeated for each polarization cross-section.
```

```
Even though scans may be interleaved and the points may not be taken
in order, each logical scan will be stored in a separate entry in the
file with

Alignment scans
===============

The sample goniometer has six degrees of freedom: rotation about x,y,z
and translation along x,y and z.  Alignment consists of centering the
sample in the beam and adjusting the x,y,z angles until the reflected
signal is maximized.  At this point theta_in is defined to be theta_out.

The incident angle theta_in is undefined until the sample is aligned.
Once the sample is aligned, all subsequent theta_in values are
measured relative to those values on the goniometer.

Data in the NeXus file should record theta_in relative to the aligned
sample rather than recording raw goniometer positions.  Alignment scans
can record raw goniometer rotations.

Slit scans
==========

As the sample is rotated the portion of the beam that it intercepts
changes, with shallower angles having fewer neutrons per unit time.
A reflectometer will have slits that can be adjusted to eliminate
any beam that does not fall on the sample.  This reduces background
which is important when measuring small signals, and provides a means
for counting the neutrons hitting the sample: Simply remove the sample
from the beam and count the neutrons that come through the slits.

Some instruments have a monitor between the slits and the sample, with
a fixed ratio between detector counts and monitor counts.  For those
instruments a single point slit scan is adequate for determining the
detector to monitor ratio.

With small samples and at low angles, the slits are usually fixed, and
the beam spills over the edges of the sample.  In this case a
footprint correction is required to correct for the changing ratio
of neutrons on the sample to neutrons missing the sample.

*** Link to or include the intensity scan within every file that needs it?


Point detectors
===============

Each theta_in-theta_out pair has a single data value associated with
it.  If theta_in is equal to theta_out, then it is a specular
scan.  If it is a little bit off, it is a background scan (either
plus or minus).  There are various reasons for offsetting theta_in
vs. theta_out.  If the detector is fixed and the sample is rocked,
then it is a rocking curve, from which you might recover the structure
of the beam, or study off specular details of the reflection. It would
even be possible to do a complete Qx-Qz scan if the beam rates are
high enough, such as from an X-ray machine.

There are a variety of ways to perform background scans depending
on the source of the background.  The most common ways are to offset
from theta_in and offset from theta_out.  This is a property of the
experiment rather than the particular instrument configuration, and
so it is the responsibility of the control program to record the
correct type.  The reduction program should have the flexibility to
interpret data in either way.

Linear detectors
```

```
================

A linear detector simultaneously samples multiple theta_out values
for a given theta_in-theta_out setting of the instrument. The data
will therefore be two dimensional, with (theta_in,theta_out) on the
primary axis, and theta distances (dtheta) on the second axis.

The detector entry in the instrument definition should indicate the
location of the edges of each pixel on the detector and the detector
position.  Using this information the NeXus file writer can easily
translate this to dtheta values for the detector ( for a detector
at distance D from the sample, dtheta=atan(d/D)*180/pi ).

Specular reflection need not be centered on the linear detector.  As
a result of the alignment process, the appropriate pixel to theta
mapping must be stored in the NXdetector entry and copied to/linked
from the dtheta vector in the data block.  This will also require
details of the electronics controlling e.g., bin widths.

Area detectors
==============

Area detectors are like linear detectors except that they can also
measure angle zeta normal to the scattering plane.  This yields
3-D data, with (theta_in,theta_out) on the primary axis, dtheta on
the second axis, and zeta on the third axis.  Normally the zeta data
is vertically integrated, possibly by the DAQ software.  The reduction
software may need to compensate for rotations of the detector as
measured from a narrow beam scan.

Polarized neutrons
==================

Some instruments have neutron spin polarizers.  This is not a separate
data dimension, however, since each polarization cross section is a
separate measurement with different flipper states.  Instead each
point is characterized by the triple (theta_in,theta_out,polarization)
where polarization is ++, +-, -+, - -.  If only one polarizer is present
the polarization would be + or - depending on which is present (look
for flipper_in or flipper_out in the data block).

+ for the incident neutron is that state which comes from the polarizer
in the absence of a front flipper.  + for the reflected neutron is the
state detected in the absence of a rear flipper.

Polarization itself can, depending on instrument, be in an arbitrary
direction relative to the sample.  This direction is defined relative
to the absolute coordinate system given by NXgeometry.  NXpolarizer
class needs to be extended with this field.

It may be possible to scan polarization angle, but the current
definition doesn't support this.

Attenuators
===========

Some instruments may have well calibrated attenuators under instrument
control.  Others will have poorly calibrated attenuators, and
attenuators not under instrument control.  Values taken with different
attenuators need to be sorted into different scans so that the
reduction program can set the attenuation independent of the nominal
value.

Scans
=====
```

Reflectometers are scan based instrument. Rather than measuring a set of detectors for a time in an entry, they measure one point, move to another configuration, measure, and so on. The configuration space need not be measured on a mesh. For example, the specular scan may be sampled more densely than the background scans. Because of this we can't use a dense array to represent our data.

With good control software, the sequence of measurements need not correspond to logical scans being constructed. For example, it is very sensible to measure all four polarization cross sections at several different fields for a particular geometry before moving to the next geometry. It is also sensible to measure specular and +/- background interleaved rather than measuring them in different scans.

With adaptive scanning the control software may decide to fill in parts of the scan where the values are changing most rapidly meaning that the sequence of measurements is not increasing along the primary axis. Data should probably be recorded in the order it is taken otherwise we are constrained to only ever write NeXus files after the fact rather than point by point as the data arrives. Reduction and viewing software should sort the data before plotting.

The NeXus format could be organized by the physical scan. That is, an NXdata element is just a set of tuples representing the machine configuration for those tuples and the counts at those configurations. However, an automatic NeXus file plotter would not be able to make sense of the data and automatically plot something reasonable.

Instead we chose to organize by the logical scan. An NXdata element is still a set of tuples representing the machine configuration and counts, but tuples from different scans are stored in a separate NXdata blocks within an NXentry.

One consequence of this organization is that each NXdata block needs its own NXmonitor for the source monitor and NXlog to record the start time and duration of the measurement at each point relative to the NXentry.

Regardless of the file layout, each separate scan must be tagged so the user can organize their data. The tag is a set of field, value pairs which together make up the condition. When viewing their data, the user should be able to organize it by various fields. For example, by scan type, by polarization cross section, or by magnetic field if the data was taken under different field conditions. This is done with a condition block within NXdata. E.g.,

```
<tag
 scan="align|slit|back+|back-|specular|rock|..."
 polarization="++|+-|-+|- -|+|-?"
 magnetic_field="NX_FLOAT32?"
 ...
></tag>
```

An individual NXentry does not exist in isolation, but is instead part of a larger dataset. This may include more conditions such as solvent concentration or temperature which is not varied within the course of a run. Users must be free to add field,value pairs to the condition as they see fit since many of the variables important to them when organizing their data are set when they are building the sample and are not part of the instrument configuration. This is true not only for reflectivity, so this should be a generic mechanism which is part of the NXdata class, and can be used by all viewing and reduction software as part of the data selection mechanism.

Each data element has a primary axis for the data and covers a range of values along that axis. For scan instruments the primary axis

```
is the axis being scanned.  As a convenience to the data selection
software, the range of values along the primary axis should be
available as an attribute of the NXdata element rather than having
to first determine the primary axis then query it for the data range.

Motors
======

Record raw motor values as well.  These will be instrument
specific.  Where standard dimensions do not correspond to
raw motors, the standard dimension will have a
derivation="expression"  attribute with expression defined
in terms of the field names of the raw motors.  dtheta should also
have an expression based on pixel number, the bin widths stored
in entry/instrument/detector, and the total distance to the
detector.  Store the zero of the raw motor as well as the
motor value.

-->
```

Monochromatic Triple Axis Spectrometer

Example 5-2. `NXmonotas.xml`

```
<!--
URL:     http://www.nexus.anl.gov/classes/xml/NXmonotas.xml
Editor:  NIAC
NIAC Version: 0.1
$Id: monotas.docbook 500 2005-06-14 16:50:35Z pfp $

Template of a generic NeXus file containing neutron or x-ray triple-axis data.-->
<NXentry name="{Name of entry}">
  <title>
    {Extended title for entry}
  </title>
  <definition URL="http://www.nexus.anl.gov/instruments/xml/NXmonotas.xml"
      version="1.0">
    NXmonotas
  </definition>
  <start_time type="ISO8601">
    {Starting time of measurement}
  </start_time>
  <NXsample name="sample">
    <name type="NX_CHAR">
      {Descriptive name of sample}?
    </name>
    <unit_cell type="NX_FLOAT32[1,6])">
      {Unit cell parameters (lengths and angles)}?
    </unit_cell>
    <plane_vector_0 type="NX_FLOAT32[3]">
      {Reciprocal space vector of primary reflection in the scattering plane}
    </plane_vector_0>
    <plane_vector_1 type="NX_FLOAT32[3]">
      {Reciprocal space vector of secondary reflection in the scattering plane}
    </plane_vector_1>
    <polar_angle units="degree" type="NX_FLOAT32[:]">
      {Polar angle of the sample with respect to the beam incident on the
      monochromator}
    </polar_angle>
    <azimuthal_angle units="degree" type="NX_FLOAT32">
      {Azimuthal angle of the sample with respect to the beam incident on the
      monochromator}
    </azimuthal_angle>
```

```
  <rotation_angle units="degree" type="NX_FLOAT32[:]">
    {Rotation angle of the sample}
  </rotation_angle>
  <Qh type="NX_FLOAT32[:]">
    {Reciprocal space component of scan}
  </Qh>
  <Qk type="NX_FLOAT32[:]">
    {Reciprocal space component of scan}
  </Qk>
  <Ql type="NX_FLOAT32[:]">
    {Reciprocal space component of scan}
  </Ql>
  <energy_transfer units="meV" type="NX_FLOAT32[:]">
    {Energy transfer of scan}
  </energy_transfer>
</NXsample>
<NXinstrument name="{Name of instrument}">
  <NXcollimator name="premonochromator_collimator">
    <type type="NX_CHAR">
      "Soller"|"radial"
    </type>
    <soller_angle units="minute" type="NX_FLOAT32">
      {Angular divergence of Soller collimator}
    </soller_angle>
  </NXcollimator>
  <NXfilter name="premonochromator_filter">
    <description type="NX_CHAR">
      {"Beryllium" | "Pyrolytic Graphite" | "Graphite"}
    </description>
  </NXfilter>
  <NXcrystal name="monochromator">
    <type type="NX_CHAR">
      {"PG (Highly Oriented Pyrolytic Graphite)" | "Ge" | "Si" | "Cu" |
      "Fe3Si" | "CoFe" | "Cu2MnAl (Heusler)" | "Multilayer"}
    </type>
    <energy units="meV" type="NX_FLOAT32[:]">
      {Optimum diffracted energy}
    </energy>
    <d_spacing units="Angstrom" type="NX_FLOAT32">
      {The planar spacing of the nominal reflection}
    </d_spacing>
    <rotation_angle units="degree" type="NX_FLOAT32[:]">
      {Rotation angle of the monochromator}
    </rotation_angle>
  </NXcrystal>
  <NXcollimator name="presample_collimator">
    <type type="NX_CHAR">
      "Soller"|"radial"
    </type>
    <soller_angle units="minute" type="NX_FLOAT32">
      {Angular divergence of Soller collimator}
    </soller_angle>
  </NXcollimator>
  <NXfilter name="presample_filter">
    <description type="NX_CHAR">
      {"Beryllium" | "Pyrolytic Graphite" | "Graphite"}
    </description>
  </NXfilter>
  <NXcollimator name="preanalyzer_collimator">
    <type type="NX_CHAR">
      "Soller"|"radial"
    </type>
    <soller_angle units="minute" type="NX_FLOAT32">
      {Angular divergence of Soller collimator}
    </soller_angle>
  </NXcollimator>
```

```
      <NXfilter name="preanalyzer_filter">
        <description type="NX_CHAR">
          {"Beryllium" | "Pyrolytic Graphite" | "Graphite"}
        </description>
      </NXfilter>
      <NXcrystal name="analyzer">
        <type type="NX_CHAR">
          {"PG (Highly Oriented Pyrolytic Graphite)" | "Ge" | "Si" | "Cu" |
          "Fe3Si" | "CoFe" | "Cu2MnAl (Heusler)" | "Multilayer"}
        </type>
        <energy units="meV" type="NX_FLOAT32[:]">
          {Optimum diffracted energy}
        </energy>
        <d_spacing units="Angstrom" type="NX_FLOAT32">
          {The planar spacing of the nominal reflection}
        </d_spacing>
        <polar_angle units="degree" type="NX_FLOAT32[:]">
          {Polar angle of the analyzer with respect to the beam incident on the
          monochromator}
        </polar_angle>
        <azimuthal_angle units="degree" type="NX_FLOAT32">
          {Azimuthal angle of the analyzer with respect to the beam incident on
          the monochromator}
        </azimuthal_angle>
        <rotation_angle units="degree" type="NX_FLOAT32[:]">
          {Rotation angle of the monochromator}
        </rotation_angle>
      </NXcrystal>
      <NXcollimator name="predetector_collimator">
        <type type="NX_CHAR">
          "Soller"|"radial"
        </type>
        <soller_angle units="minute" type="NX_FLOAT32">
          {Angular divergence of Soller collimator}
        </soller_angle>
      </NXcollimator>
      <NXdetector name="detector">
        <counts signal="1" axes="energy_transfer|Qh|Qk|Ql" type="NX_INT32[:]">
          {Integer counts}
        </counts>
        <polar_angle units="degree" type="NX_FLOAT32[:]">
          {Polar angle of the detector with respect to the beam incident on the
          monochromator}
        </polar_angle>
        <azimuthal_angle units="degree" type="NX_FLOAT32">
          {Azimuthal angle of the detector with respect to the beam incident on
          the analyzer}
        </azimuthal_angle>
      </NXdetector>
    </NXinstrument>
    <NXmonitor name="monitor">
      <mode type="NX_CHAR">
        "monitor"|"timer"
      </mode>
      <preset type="NX_FLOAT32[1]">
        {preset value for time or monitor}
      </preset>
      <data type="NX_INT[:]">
        {Monitor data}?
      </data>
    </NXmonitor>
    <NXdata name="data">
      <Qh NAPIlink="NXentry/NXsample/Qh">
      </Qh>
      <Qk NAPIlink="NXentry/NXsample/Qk">
      </Qk>
```

```
    <Ql NAPIlink="NXentry/NXsample/Ql">
    </Ql>
    <energy_transfer NAPIlink="NXentry/NXsample/energy_transfer">
    </energy_transfer>
    <counts NAPIlink="NXentry/NXinstrument/detector/counts">
    </counts>
    <energy NAPIlink="NXentry/NXinstrument/analyzer/energy">
    </energy>
  </NXdata>
</NXentry>
```

Time of Flight Neutron Direct Geometry Spectrometer

Example 5-3. `NXtofndgs.xml`

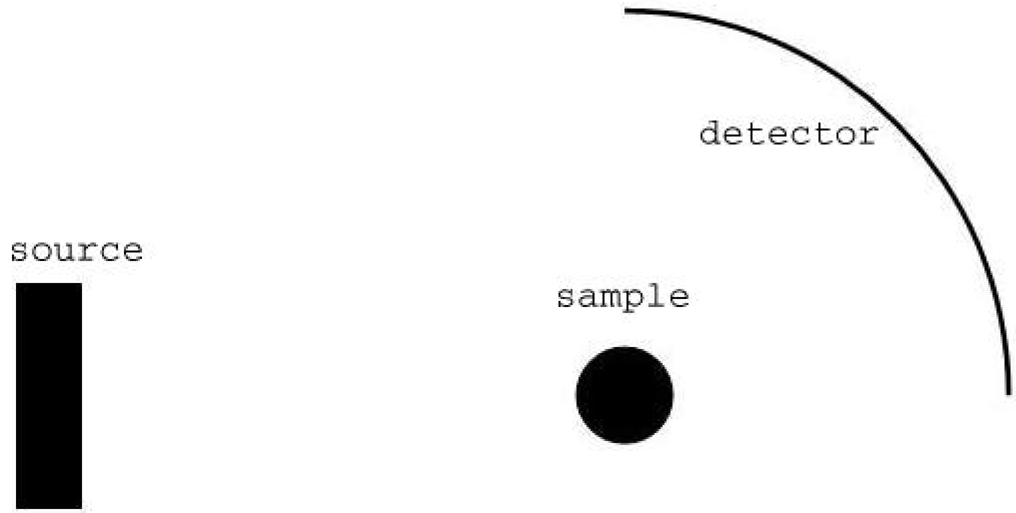
```
<!--
URL:     http://www.nexus.anl.gov/classes/xml/NXtofndgs.xml
Editor:  NIAC
NIAC Version: 0.1
$Id: tofdgs.docbook 500 2005-06-14 16:50:35Z pfp $

Template of a generic NeXus file containing data from a direct geometry
time-of-flight spectrometer.
-->
<NXentry name="{Name of entry}">
  <title>
    {Extended title for entry}
  </title>
  <definition URL="http://www.nexus.anl.gov/instruments/xml/NXtofndgs.xml"
      version="1.0">
    NXtofndgs
  </definition>
  <start_time type="ISO8601">
    {Starting time of measurement}
  </start_time>
  <NXsample name="sample">
    <name type="NX_CHAR">
      {Descriptive name of sample}?
    </name>
    <unit_cell type="NX_FLOAT32[1,6])">
      {Unit cell parameters (lengths and angles)}?
    </unit_cell>
    <sample_orientation type="NX_FLOAT[3]">
      {This will follow the Busing and Levy convention from Acta.Crysta v22,
      p457 (1967)}?
    </sample_orientation>
    <orientation_matrix type="NX_FLOAT[3,3]">
      {Orientation matrix of single crystal sample}{The is the orientation
      matrix using Busing-Levy convention}?
    </orientation_matrix>
    <mass type="NX_FLOAT">
      {Mass of sample}?
    </mass>
    <NXgeometry>
      <NXshape>
        {Shape of sample}
      </NXshape>
    </NXgeometry>
  </NXsample>
  <NXinstrument name="{Name of instrument}">
    <NXmoderator name="{Name of moderator}">
      <distance type="NX_FLOAT">
        {Effective distance as seen by measuring radiation}?
```

```
  </distance>
</NXmoderator>
<NXchopper name="monochromator">
  <distance>
    {Distance of the centre of the chopper to the sample.}
  </distance>
  <energy type="NX_FLOAT">
    {Optimum energy transmitted by the chopper.}
  </energy>
  <type type="NX_CHAR">
    {fermi|disk|counter-rotating|statistical}
  </type>
  <rotation_speed type="NX_FLOAT">
    {Chopper rotation speed}
  </rotation_speed>
</NXchopper>
<NXcrystal name="monochromator">
  <distance>
    {Distance of the centre of the crystal monochromator to the sample.}
  </distance>
  <type type="NX_CHAR">
    {"PG (Highly Oriented Pyrolytic Graphite)" | "Ge" | "Si" | "Cu" |
    "Fe3Si" | "CoFe" | "Cu2MnAl (Heusler)" | "Multilayer"}
  </type>
  <energy units="meV" type="NX_FLOAT32[:]">
    {Optimum diffracted energy}
  </energy>
  <d_spacing units="Angstrom" type="NX_FLOAT32">
    {The planar spacing of the nominal reflection}
  </d_spacing>
</NXcrystal>
<NXdetector name="{Name of detector bank}">
  <data signal="1" axes="x_angle:y_angle:time_of_flight"
      type="NX_FLOAT[i,j,k]|NX_INT[i,j,k]">
    {Data values}?
  </data>
  <time_of_flight type="NX_FLOAT[k+1]">
    {Total time of flight}
  </time_of_flight>
  <distance type="NX_FLOAT[i,j]">
    {distance from the sample
  </distance>
  <data_errors type="NX_FLOAT[i,j,...]|NX_INT[i,j,...]">
    {Data errors}
  </data_errors>
  <x_offset type="NX_FLOAT[i]">
    {offset from the detector center in x-direction}?
  </x_offset>
  <y_offset type="NX_FLOAT[j]">
    {offset from the detector center in the y-direction}?
  </y_offset>
  <x_angle type="NX_FLOAT[i]">
    {angle of detector in x-direction with respect to unscattered beam}?
  </x_angle>
  <y_angle type="NX_FLOAT[j]">
    {angle of detector in y-direction with respect to unscattered beam}?
  </y_angle>
  <polar_angle type="NX_FLOAT[i,j]">
    {polar angle of a detector pixel}
  </polar_angle>
  <azimuthal_angle type="NX_FLOAT[i,j]">
    {azimuthal angle of a detector pixel}
  </azimuthal_angle>
  <solid_angle type="NX_FLOAT[i,j]">
    {Solid angle subtended by the detector pixel at the sample}?
  </solid_angle>
```

```
      <x_pixelsize type="NX_FLOAT[i,j]">
        {Size of each detector pixel}?
      </x_pixelsize>
      <y_pixelsize type="NX_FLOAT[i,j]">
        {Size of each detector pixel}?
      </y_pixelsize>
      <gas_pressure type="NX_FLOAT[i]">
        {Detector gas pressure}?
      </gas_pressure>
      <type type="NX_CHAR">
        "He3 gas cylinder"|He3 PSD"|?
      </type>
      <NXgeometry name="geometry">
        {Position and orientation of detectors}?
      </NXgeometry>
    </NXdetector>
  </NXinstrument>
  <NXmonitor name="whitebeam_monitor">
    <distance type="NX_FLOAT">
      {Distance of monitor from sample}
    </distance>
    <time_of_flight type="NX_FLOAT[i]">
      {Time-of-flight}
    </time_of_flight>
    <data signal="1" axes="time_of_flight" type="NX_INT[i]">
      {Monitor data}
    </data>
  </NXmonitor>
  <NXmonitor name="presample_monitor">
    <distance type="NX_FLOAT">
      {Distance of monitor from sample}
    </distance>
    <time_of_flight type="NX_FLOAT[i]">
      {Time-of-flight}
    </time_of_flight>
    <data signal="1" axes="time_of_flight" type="NX_INT[i]">
      {Monitor data}
    </data>
  </NXmonitor>
  <NXmonitor name="beamstop_monitor">
    <distance type="NX_FLOAT">
      {Distance of monitor from sample}
    </distance>
    <time_of_flight type="NX_FLOAT[i]">
      {Time-of-flight}
    </time_of_flight>
    <data signal="1" axes="time_of_flight" type="NX_INT[i]">
      {Monitor data}
    </data>
  </NXmonitor>
  <NXdata name="{Name of data bank}">
    <data NAPIlink="NXentry/NXinstrument/NXdetector/data">
    </data>
    <x_angle NAPIlink="NXentry/NXinstrument/NXdetector/x_angle">
    </x_angle>
    <y_angle NAPIlink="NXentry/NXinstrument/NXdetector/y_angle">
    </y_angle>
    <x_offset NAPIlink="NXentry/NXinstrument/NXdetector/x_offset">
    </x_offset>
    <y_offset NAPIlink="NXentry/NXinstrument/NXdetector/y_offset">
    </y_offset>
    <time_of_flight NAPIlink="NXentry/NXinstrument/NXdetector/time_of_flight">
    </time_of_flight>
  </NXdata>
</NXentry>
```

Time of Flight Neutron Powder Diffractometer



Schematic diagram of the generic time of flight neutron powder diffractometer.

The time of flight powder diffractometer (TOFNPD) is an instrument used with a couple of different types of analysis. For that reason the composite TOFNPD definition is made up of three separate definitions.

TOFNPD:Time Focus

To time focus data there is little information required. The parameters needed in the file are

1. unique detector pixel identifier
2. primary flight path
3. detector pixel position
4. detector pixel solid angle covered

In addition, the software needs to have some additional information that is specified by the user.

1. mapping of detector pixel identifiers to focused detector pixel identifier
2. focused detector position
3. unique focused detector identifier

Example 5-4. `TOFNPD:time_focus.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXtofnpd.xml
Editor:  NIAC
$Id: tofnpd.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Instrument definition for a time-of-flight neutron powder diffractometer that
can be time focused.
-->
<NXentry name="{Entry Name}">
```

```
  <definition URL="http://www.neutron.anl.gov/nexus/xml/NXtofnpd-timefocus.xml"
      version="$Revision: 1422 $" type="NX_CHAR[]" instrument="TOFNPD">
    Time Focus
  </definition>
  <NXsample name="">
     <chemical_formula type="NX_CHAR"></chemical_formula>
  </NXsample>
  <NXinstrument name="{name of the instrument}">
    <name long_name="{full name of instrument}?" type="NX_CHAR[]">
      {Abbreviated name of instrument}
    </name>
    <NXmoderator name="">
      <distance units="metre" type="NX_FLOAT">
        {distance from the sample (should be negative)}
      </distance>
    </NXmoderator>
    <NXdetector name="{Name of detector bank}">+
      <time_of_flight link="{absolute path to location in NXdetector}"
          units="10^-6 second|10^-7 second" type="NX_FLOAT[i+1]">
        {Total time of flight}
      </time_of_flight>
      <pixel_id link="{absolute path to location in NXdetector}"
          type="NX_INT[j]">
        {Identifier for detector}
      </pixel_id>
      <counts signal="1" axes="[time_of_flight,pixel_id]"
          link="{absolute path to location in NXdetector}"
          units="number" type="NX_FLOAT[i,j]">
        {Data values}
      </counts>
      <distance axes="pixel_id" type="NX_FLOAT[j]">
      </distance>
      <polar_angle axes="pixel_id" type="NX_FLOAT[j]">
      </polar_angle>
      <azimuthal_angle axes="pixel_id" type="NX_FLOAT[j]">
      </azimuthal_angle>
      <solid_angle axes="pixel_id" type="NX_FLOAT[j]"></solid_angle>
      <TOF_to_d method="linear|quadradic" type="NX_FLOAT[j,k]">
         {Calibrated conversion factors to be used for time focusing}
      </TOF_to_d>
    </NXdetector>
  </NXinstrument>
  <NXdata name="">
      <time_of_flight
          NAPIlink="/NXentry/NXinstrument/NXdetector/time_of_flight"/>
      <pixel_id
          NAPIlink="/NXentry/NXinstrument/NXdetector/pixel_id"/>
      <counts
          NAPIlink="/NXentry/NXinstrument/NXdetector/data"/>
  </NXdata>
</NXentry>
```

Example 5-5. `TOFNPD:rietveld.xml`

```
<!--
URL:     http://www.neutron.anl.gov/nexus/xml/NXtofnpd.xml
Editor:  NIAC
$Id: tofnpd.docbook 1422 2010-02-10 13:59:27Z Pete Jemian $

Instrument definition for a time-of-flight neutron powder diffractometer that
can be time focused.
-->
<NXentry name="{Entry Name}">
  <conforms_to>
    <definition URL="" version="">TOFNPD:Time Focus</definition>
```

```
  </conforms_to>
  <definition URL="http://www.neutron.anl.gov/nexus/xml/NXtofnpd-timefocus.xml"
      version="$Revision: 1422 $" type="NX_CHAR[]" >
    TOFNPD:Rietveld
  </definition>
  <NXcharacterization
     name="isotropic_scatterer"
     NXS:location="" instrument="TOFNPD" version="" URL=""
     definition="Time Focus">{Should be same scales as this entry}
  </NXcharacterization>
</NXentry>
```

Time of Flight Neutron Reflectometer

Example 5-6. `tofnref.xml`

```
<!--
Instrument Definition for a Time of Flight Neutron Reflectometer

Editor: Robert Dalgliesh <r.m.dalgliesh@rl.ac.uk>
Initial version: October 2004
$Id: tofnref.docbook 500 2005-06-14 16:50:35Z pfp $

See http://www.neutron.anl.gov:8080/NeXus/5 for component definitions.
2005-04-28 Paul Kienzle <pkienzle@nist.gov> * Only include information that may
be used for reduction/analysis * Make consistent with monoref and monotas
-->
<NXentry>
  <definition URL="http://www.nexus.anl.gov/instruments/xml/tofnref.xml"
      version="1.0">
    TOFNREF
  </definition>
  <start_time type="ISO8601">
  </start_time>
  <tag polarization="++|+-|-+|--|+|-?" magnetic_field="NX_FLOAT?"
      userfield="..." scan="spectrum|background+|background-|specular|rock|...">
    {Empty.}
    <!--
    *** HDF may not allow empty blocks, so maybe a single integer
    *** to distinguish the various conditions.
    -->
  </tag>
  <NXsample>
    <rotation_angle units="degrees" type="NX_FLOAT[i]">
    </rotation_angle>
  </NXsample>
  <!--
  *** link to spectrum measurement for intensity vs. wavelength
  *** for a given slit setting

  *** warning: beam profile is not regular, but this effect is
  *** accomodated in the spectrum measurement
  -->
  <NXinstrument>
    <NXmoderator name="{Name of moderator}">
      <distance units="metre" type="NX_FLOAT">
        { Distance from T_o to sample along beam-path. To calculate wavelength:
        L[i] = wavelength at time T[i] T[i] = time of flight for point i. d1 =
        distance from moderator to sample along beam path d2 = distance from
        detector to sample along beam path h = Planck's constant m_n = mass of
        the neutron L[i] = h/m_n * T[i]/(d1+d2) }
      </distance>
```

```
  <pulse_shape type="NXdata">
    { Find the center of mass of the pulse shape and use that as the T0
    offset with respect to the protons hitting the target. The TOF from
    target (which is the real T0) to the moderator is insignficant compared
    to the uncertainty from the pulse shape and so can be ignored. }
  </pulse_shape>
</NXmoderator>
<NXguide name="{Name of guide section}">
  { Guides in total or in segments thgrough to sample position; may be
  interspersed between other components - Check component index. Can be
  nested for guides with multiple straight segments. Affects wavelength
  spectrum, both in divergence and intensity. The spectrum scan will
  automatically compensate for intensity effects. To compute divergence
  effects, detailed information about the guide geometry will be required.

</NXguide>
<!--
Some instruments will require gravitational corrections. Neutrons travel on
a parabolic trajectory. For long wavelength neutrons this changes incident
and reflected angle and results in the neutron appearing on a lower
detector pixel than expected. The information required for these
corrections comes from the instrument geometry.
-->
<NXchopper name="[T0_chopper|frame_overlap_chopper]">
  <wavelength_range units="Angstrom" type="NX_FLOAT[2]">
    { Reduction software needs to ignore Q values outside the range defined
    by the choppers. The T0 chopper is phased to the source to block fast
    neutron and gamma flash. The frame overlap chopper is set to select low
    wavelength neutrons (those from the current pulse) or high wavelength
    neutrons (those from the previous pulse. On a properly tuned
    instrument, the time bins recorded in the detector will reflect the
    actions of the choppers and these fields can be ignored. }
  </wavelength_range>
</NXchopper>
<NXmirror name="frame_overlap_mirror">
  <cutoff_wavelength mode="above|below">
    <!-- *** This is not part of standard NXmirror -->
    { The frame overlap mirror is used to eliminate very long wavelength
    neutrons from previous pulses. Together with the choppers, this helps
    to choose which pulse to use in the TOF calculations. On a properly
    tuned instrument the time bins recorded in the detector will account
    for the actions of the mirror. There will be some attenuation but this
    will be compensated for when correcting for the spectrum scan. For an
    ab initio calculation, you would need to store the angle wrt the beam
    to compute the cutoff angle but often this will not be explicit since
    the instrument is simply tuned to have the correct cutoff. } }
  </cutoff_wavelength>
</NXmirror>
<!-- collimation -->
<NXaperture name="pre[sample|detector]_slit[|y]#">
  <opening units="mm" filter="Qx|Qy" type="NX_FLOAT">
    <!-- *** This is not part of standard NXaperture -->
    { Slit opening; this is a scan parameter, so cannot be recorded as part
    of the aperture size. }
  </opening>
  <NXgeometry name="geometry">
    <NXtranslation name="translation">
      <distances units="mm" type="NX_FLOAT">
        { Location of slit along beamline (midway between slits if slits
        are not coplanar). This is required to compute instrument
        resolution. }
      </distances>
      <angles>
        { Only need angles if slits are not centered on beam wrt sample. }?
      </angles>
    </NXtranslation>
```

```
        <NXshape name="shape">
          <type type="NX_CHAR">
            nxslit
          </type>
          <size units="mm" type="NX_FLOAT[2]">
            { size[1] is thickness of material (actual slits can be more
            complicated in practice, being composed of different materials that
            are not aligned, but these can be simulated with sets of slits).
            size[2] is zero if slits are coplanar, otherwise it is the distance
            between slits. each half of the slit is considered to be a
            semi-infinite plane cutting neutrons in Qx if they are in the
            scattering plane, or Qy if they are normal to the scattering plane
            as defined by
            <opening filter="...">
            </opening>
            above. }
          </size>
        </NXshape>
      </NXgeometry>
    </NXaperture>
    <!--
    The polarizer-flipper-guidefield combination selects polarization vectors
    in and out of the sample. A number of scans are required to tune the
    instrument so that polarization is either 'up' or 'down' on the sample. On
    correctly tuned instruments the polarization angle selected should be
    recorded by the flipper using polar_angle relative to the surface (0/180
    for +/-, or with out of plane polarization, 90/270 for +/-). The
    polarization efficiency must be determined from a spectrum scan and the
    appropriate correction applied to the data. Raw values from the instrument,
    such as time dependent field applied to flipper coils or current on the
    current sheet can be recorded for specialized reduction programs which know
    how to handle them.
    -->
    <NXpolarizer name="presample_polarizer">
    </NXpolarizer>
    <NXflipper name="presample_flipper">
      <polar_angle units="degree" type="NX_FLOAT">
      </polar_angle>
    </NXflipper>
    <NXpolarizer name="predetector_polarizer">
    </NXpolarizer>
    <NXflipper name="predetector_flipper">
      <polar_angle units="degree" type="NX_FLOAT">
      </polar_angle>
    </NXflipper>
    <!-- detector may be protected by an attenuator and/or a beam stop -->
    <NXattenuator>
      <attenuator_transmission type="NX_FLOAT">
        { The nominal amount of the beam that gets through (transmitted
        intensity)/(incident intensity) }
      </attenuator_transmission>
    </NXattenuator>
    <NXbeam_stop name="stop">
      ? { Need all fields so that we can calculate shadow of beam stop on
      detector. }
    </NXbeam_stop>
    <NXdetector name="detector">
      <!--
      polar_angle and azimuthal_angle define the location of the detector
      relative to the beamzero
      -->
      <distance units="mm" type="NX_FLOAT">
        { distance from sample }
      </distance>
      <translation units="centimeter" type="NX_FLOAT[2]">
        { translation normal to direct beam }?
```

```
    </translation>
    <time_of_flight units="10^-6 second|10^-7 second" type="NX_FLOAT[l+1]">
      { Total time of flight }
    </time_of_flight>
    <polar_angle units="degrees" type="NX_FLOAT[i]">
      { angular position of detector relative to beamzero through sample --
      known to practitioners as "A4" or "two theta" }
    </polar_angle>
    <azimuthal_angle units="degrees" type="NX_FLOAT">
      { Indicate sense of scattering: 0 is front surface of sample, 180 is
      back surface of sample. If 180, change the sign of the reflected angle
      in the data. It is also possible for the beam to enter the substrate
      from the side and reflect off the back surface of a film, in which case
      negative angles can be interpreted as inverting the scattering length
      density profile of the film (after accounting for absorption in the
      substrate. }
    </azimuthal_angle>
    <x_offset units="mm" type="NX_FLOAT[j]">
      {pixel edges in x}?
    </x_offset>
    <y_offset units="mm" type="NX_FLOAT[k]">
      {pixel edges in x}?
    </y_offset>
    <counts signal="1" axes="time_of_flight,x_offset?,y_offset?,polar_angle?"
        type="NX_INT[l,k,? j,? i]">
      { raw detector counts }
    </counts>
    <!--
    Time bins are logarithmic, but identical for each pixel on the detector
    this keeps constant resolution in Q. If not, then save things in
    different bins. Time bins are set for the lowest angle and with
    resolution improving at higher angles. Since dtheta/theta dominates,
    there’s little benefit to changing time bins at higher angles.
    Regardless, lots of rebinning is required because the Q steps are too
    fine.

    *** Raw counts are meaningless to the user if they are counting
    *** against detector since all values will be the same. The data
    *** only become meaningful when divided by counting time or monitor
    *** as specified by the ratio field. The generic plotting program
    *** will need to sort this out.

    *** In general, n-D data should contain k-D summary statistics
    *** for all 0&lt;=k&lt;n. That way a really dumb plotting program can
    *** still display info from a 3-D result as a line.

    *** Some control systems have data windows. Windows may be
    *** defined in terms of pixel ranges or in terms of
    *** theta_in-theta_out relationships (i.e., Qx). Each scan
    *** point should have start/end positions and summary
    *** statistics for every window defined. Window size may be
    *** dynamic. Store windows in separate data blocks.
    -->
  </NXdetector>
</NXinstrument>
<NXmonitor name="monitor">
  <mode type="NX_CHAR">
    monitor
  </mode>
  <preset type="NX_FLOAT">
    { preset value for monitor }?
  </preset>
  <data units="counts" type="NX_INT[i]">
    { record of monitor counts }?
  </data>
  <efficiency type="Nxdata">
```

```
        { Monitor efficiency as a function of wavelength }?
      </efficiency>
      <sampled_fraction units="dimensionless" type="NX_FLOAT">
        { Proportion of incident beam sampled by the monitor }
      </sampled_fraction>
    </NXmonitor>
    <NXmonitor name="timer">
      <mode type="NX_CHAR">
        timer
      </mode>
      <preset type="NX_FLOAT">
        { preset value for timer }?
      </preset>
      <data units="seconds" type="NX_INT[i]">
        { record of times for individual points }?
      </data>
    </NXmonitor>
    <NXdata>
      <time_of_flight units="second"
          NAPIlink="entry/instrument/detector/time_of_flight" type="NX_FLOAT[k]">
      </time_of_flight>
      <attenuator_transmission NAPIlink="NXentry/NXinstrument/NXattenuator">
      </attenuator_transmission>
      <theta NAPIlink="NXentry/NXsample/rotation_angle">
      </theta>
      <twotheta NAPIlink="NXentry/detector/polar_angle">
      </twotheta>
      <presample_slit1 NAPIlink="NXentry/presample_slit1/opening">
      </presample_slit1>
      <presample_slit2 NAPIlink="NXentry/presample_slit2/opening">
      </presample_slit2>
      <predetector_slit1 NAPIlink="NXentry/predetector_slit1/opening">
      </predetector_slit1>
      <predetector_slit2 NAPIlink="NXentry/predetector_slit2/opening">
      </predetector_slit2>
      <counts NAPIlink="NXentry/detector/counts">
      </counts>
      <count_start units="second" type="NX_FLOAT[i]">
        <!-- probably shouldn't store any real data here, but where else? -->
        { start time of each measurement point relative to start time of entry. }
      </count_start>
      <timer NAPIlink="NXentry/timer/data">
      </timer>
      <monitor NAPIlink="NXentry/monitor/data">
      </monitor>
    </NXdata>
    <NXlog name="??">
      { Various logs for temperature, field, etc. which are assumed to be
      constant over the duration of the run. The reduction program should be able
      to display their values on a parallel graph. Note that logs are not
      necessarily sampled synchronously with the data points. }*
    </NXlog>
  </NXentry>
```