NEXUS—a portable, extensible, self-describing data format for neutron and X-ray scattering data.

Przemek Kłosowski (przemek@nist.gov)
NIST, Bldg. 235, Gaithersburg, MD 20899, USA

Mark Könnecke (Mark.Koennecke@psi.ch)
Paul Scherrer Institute, CH-5232 Villigen PSI, Switzerland

Jonathan Tischler (tischlerjz@ornl.gov)
Ray Osborn (ROsborn@anl.gov)
Argonne National Laboratory, Argonne, IL 60439, USA

October 7, 1996
(Revision: 1.24)

Abstract

Several portable, standard data file formats are evaluated in the context of introducing data format compatibility among the neutron scattering facilities. The HDF format is proposed as a first choice.

A need for standardizing the contents of the file is recognized, prompting proposal and definition of a scheme for classifying the data: a neutron/X-ray unified standard data format (NEXUS), defined in this document.

A companion document describes sample implementation of a program converting an existing data format into NEXUS standard data file.

Contents

1 Introduction **2**

2 File structure and contents **4**
2.1 Naming conventions 4
2.2 General conventions 5
2.3 Global attributes and data 5
2.4 Experiment data sections 6
2.5 Instrument description 7
2.6 Sample description 8
2.7 Scan data 10

3 Conclusions **13**

1 Introduction

Historically, the experimental research community did not have any great motivation to standardize the form in which data are stored on computers. In consequence, there are as many, mostly incompatible, data formats, as there are research groups working in the field.

This used to be acceptable to researchers who had a self-contained, well defined set of programs to process and display the data, with only the final result being made available to others. However, when one contemplates exchanging raw data with other groups, or using third-party programs to try novel ways of data reduction, one is faced with writing the dreaded format conversion tools. Usually, there is not much difficulty in writing any such particular tool; however, with the large and growing number of research groups, and with a growing need to exchange data with researchers from other fields (theorists, X-ray and light scatterers, computer modelers), one falls victim to a combinatorial explosion of the number of necessary converters. This state of affairs has to be considered in the context of a recently discussed "software crisis" in the U.S. neutron scattering community, caused by, among others, increasingly heterogeneous computing environments, and lack of funding and/or manpower for programming tasks.

A simple solution to the data exchange debacle is the introduction of a common, standard data format. Such a standard will improve the file exchange situation in two ways. First, for existing data files and data reduction codes, it reduces the number of converters necessary[1]. Secondly, if all newly written programs would read and write the common format, the need for data conversion would disappear altogether.

Such a common standard must satisfy the following criteria:

- implementation should be readily available and portable to all popular computing platforms
- the format should be sufficiently general to satisfy the present and future needs of its users

This last requirement basically means that the data format must be extensible and self-describing.

Since 1994, when this proposal was first formulated, parties representing nearly all the major neutron scattering laboratories in the US (Argonne, Brookhaven, Los Alamos, Oak Ridge, NIST) as well as the US synchrotron radiation community and European neutron scatterers, agreed to propose a joint data exchange standard, based on the HDF file format and library, originated at National Center for Supercomputing Applications.

The specifications of this format, named NEXUS, have been formulated; NIST and other groups plan to use it as their future data format.

Review of existing data file formats

We have considered four major existing public-domain self-describing standards for data files: ISO STEP/Express, FITS, netCDF and HDF. The International Standard Organization's STEP/Express proposal has not been used extensively for scientific data so far; we will not

[1]Indeed, if we have n distinct data formats, we need on the order of n^2 direct format-to-format conversions. Using a common format C, we could perform a double translation $A \to C \to B$, and need only $2n$ converters.

consider it in this proposal. The astronomical data FITS format is somewhat limited in its functionality, and appears to be inextensible.[2] The netCDF standard, conceived at Unidata, and the Hierarchical Data Format (HDF) developed at the National Center for Supercomputing Applications (NCSA), are much more widespread. Both netCDF and HDF have been ported to a wide variety of platforms, including PC compatible and Macintosh platforms, workstations and minicomputers, and mainframes and massively parallel environments. NetCDF has been considered for use in the European neutron scattering community [3], but we recently agreed to recommend HDF for reasons outlined below.

The principal concept in the HDF standard is the Scientific Data Set (SDS)—a data container that can be endowed with arbitrary attributes such as dimension scales, axes, units or comments. Additionally, a collection of SDSs can be flexibly organized into sets, called Vgroups, which help to describe hierarchies of detail in the data. This capability of organizing data sets into higher order ensembles (data hierarchy) is one of the main advantages of HDF over netCDF, which has a flat file organization.

Logical examples of such groupings are:

- a collection of linear scans, e.g. transverse and longitudinal scans through a peak, or a Q-space map made of linear scans,
- a collection of scans at different temperatures,
- empty can and sample runs
- experimental data and post-processing data (numerical simulations, computationally intensive data reductions, etc).

Yet another situation in which the hierarchical model of HDF files conveniently expresses the relevant information is in the instrument description. As one of us has noted [3], instruments are best described as a collection of "building blocks" (e.g. "monochromator, filter, sample, detector"), each block having a list of properties. This hierarchy is easily expressed using the HDF data format.

There is also the issue of support and long-term popularity. Both netCDF and HDF were written by national facilities, but netCDF is not as actively developed. HDF effort at NCSA has also been scaled back, but they still have several people working on improvements; a new release (v. 4.0) has been released in 1996.

Another important issue is support for the chosen file format in terms of available software that can understand HDF files. There are, of course, generic tools from NCSA such as Mosaic and Collage, as well as the popular `xmgr` 2D plotting program running in X windows/Unix environment. There are also a number of commercial applications, such as Spyglass, IDL and AVS which have procedures for reading HDF files. A HDF-aware program for non-linear fitting of data to arbitrary functions is being written by one of the authors (PK).

We propose thus to use HDF as the library and data format for portable data exchange within the neutron and X-ray community. In October 1994, Argonne National Laboratory organized a workshop on software support for the neutron scattering community, where interested parties representing nearly all the major neutron scattering laboratories in the US

[2]The FITS standard seems to have originated concurrently with other standards such as the high energy physics ZEBRA library written at CERN. These formats tend to address only the data portability issue and are not self-describing, i.e. they require problem-specific programs for both reading and writing the data.

(Argonne, Brookhaven, Los Alamos, NIST and Oak Ridge) as well as the US synchrotron radiation community and European neutron scatterers, agreed to propose a joint data exchange standard, based on the HDF format. This proposal has been subsequently revised, and is presented in the present document.

2 File structure and contents

Regardless of the choice of general format, there are specific issues dealing with the content of the file. There has to be discipline in organizing the data, so that data files and programs written at different locations will be fully compatible with one another. This additional structure is just an agreement on what information is included and in what order. It in no way removes the basic portability of HDF files; generic HDF-aware visualization tools, file content listing and editing programs, etc., would still read the file just fine.

Following the approach of [2] and [3], we will define attributes and variables which are valid or required in the neutron scattering context, and therefore expected to be present in NEXUS data files.

The conventions presented here do not demand that all proposed data items be found in every file. In consequence, a conforming NEXUS file might contain so little information that it would be impossible to perform many standard analyses (resolution calculation, etc.)—perhaps only plotting of the data would be possible. Such laxness in the proposed standard recognizes the reality that many pre-existing data sets may lack full experimental information, and requiring this information would make it impossible to write a fully automatic translation program.

There is of course a tradeoff between file simplicity and amount of information present, and NEXUS users must make a judgment call on this tradeoff. In our opinion it is better to err on the side of extra information in the file—HDF allows one to query the existing data by name, so that generic reading programs can simply disregard extraneous data, while more specific programs can still use it.

2.1 Naming conventions

The following naming conventions are proposed for NEXUS data and attribute names and classes:

- Lower case letters are used throughout, except for common symbols and abbreviations such as `Q_space` or `FWHM`. Note that HDF names are case-sensitive: this rule is designed to avoid confusion between, e.g. `name` and `Name`.

- Names are constructed of full words separated by the underscore character (i.e. `lattice_constant` rather than `lattconst`).

- For sequentially indexed group names, the sequential number is simply appended to the name (e.g. `filter1`, `filter2`, etc.). This convention should be used only for data group names, and then sparingly, because of a potential for confusion between indexed and non-indexed names (`filter` vs. `filter1`).[3]

[3]The data sets with multiple instances of some primitive data record should be stored in multidimensional arrays rather than in differently named HDF data elements.

- Class names are prefixed with with letters `NX` to avoid clashes with class names internal to HDF.

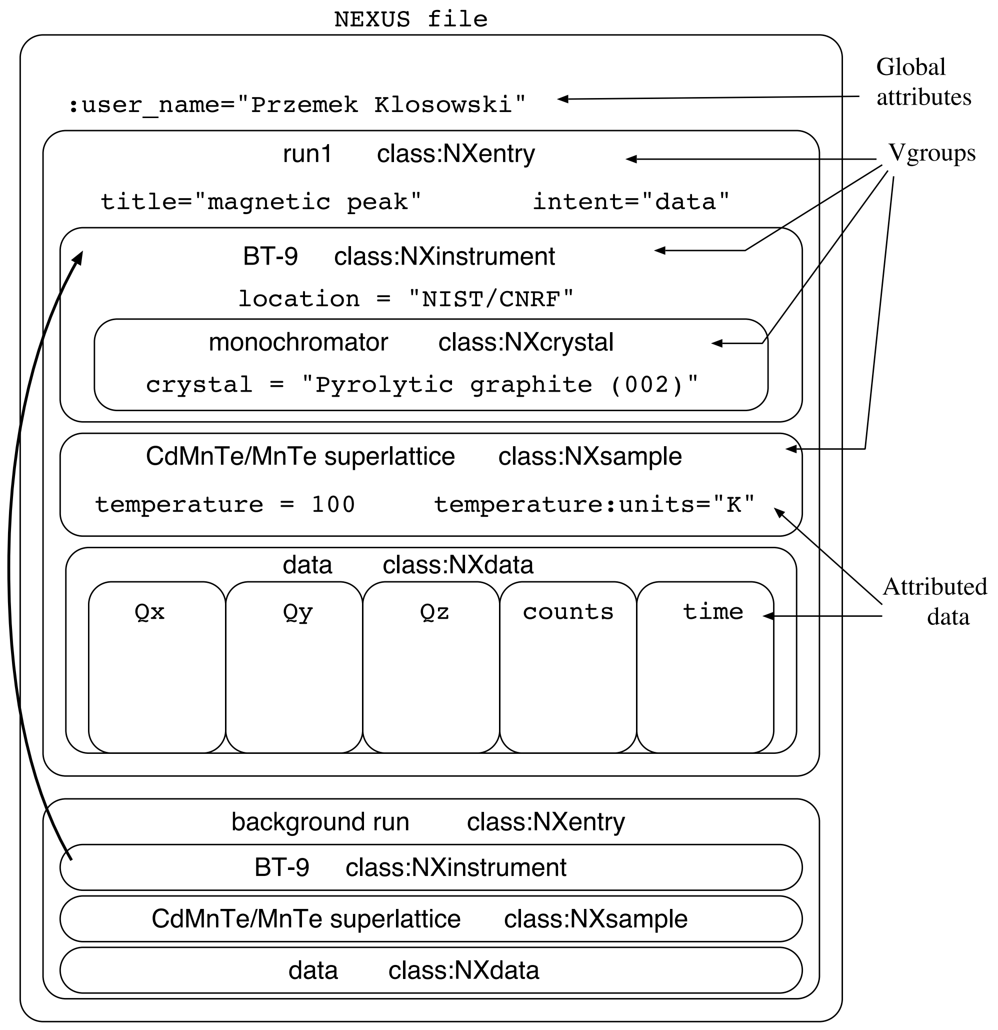
NEXUS file
Global attributes
:user_name="Przemek Klosowski"
run1 class:NXentry
Vgroups
title="magnetic peak" intent="data"
BT-9 class:NXinstrument
location = "NIST/CNRF"
monochromator class:NXcrystal
crystal = "Pyrolytic graphite (002)"
CdMnTe/MnTe superlattice class:NXsample
temperature = 100 temperature:units="K"
data class:NXdata
Qx
Qy
Qz
counts
time
Attributed data
background run class:NXentry
BT-9 class:NXinstrument
CdMnTe/MnTe superlattice class:NXsample
data class:NXdata

Figure 1: Proposed structure of a NEXUS file. The file is composed of class `NXentry` Vgroups, each of which contains lower level data groups describing various aspects of the data. As explained in the text, fixed file-wide information can be multiply linked instead of being duplicated (e.g. the `instrument` group).

2.2 General conventions

It is very important that all relevant attributes be provided for the variables included in a NEXUS file. This primarily means that data units should be correctly specified.

The data values should follow applicable standards:

Dates and times use ISO 8601 format (e.g. `1996-07-31 21:15:22+0600`)[4]. Because civil time zones, such as US Eastern Daylight Savings (EDT), aren't unambiguously defined,

[4]`mailto://sales@isocs.iso.ch, http://www.ft.uni-erlangen.de/ mskuhn/iso-time.html`

use of Universal Coordinated Time (possibly with offset, as shown here) is recommended. Note also the use of four-digit year.

Physical units use official SI unit names with fully spelled prefixes and unit names[5].

The HDF Vgroups are identified by both individual name and a class name. The former identifies a particular data group, while the latter can be used as a general functional description for this group[6].

Data sets in HDF can be of arbitrary dimensionality. Reading programs can find out what are the dimensions of data in a particular SDS. This feature allows flexibility in providing data. For instance, to specify a monochromator, one could provide a nominal value as an one-dimensional, single-element SDS; alternatively, a $2 \times n$ data array would provides a wavelength distribution.

Another application of flexibility in data sizes is to describe multiple values of some quantity, such as unit cell data for a multi-phase sample. A single phase would be described by a six-element array giving the dimensions and angles of the cell; the multi-phase sample would use a 2-D array, with consecutive 6-element rows containing data for respective phases.

2.3 Global attributes and data

Global attributes provide information pertaining to the file itself. Here is a list of the proposed attributes[7]; the underlined items are mandatory, others are optional:

Attribute name	Type	Description
<u>`file_name`</u>	CHAR8	Original file name under which the data file was opened.
<u>`file_time`</u>	CHAR8	Original date and time when data file was opened.
`file_history`	CHAR8	Extendable string; appending/modifying the data should also append an explanation line to this attribute.
<u>`NEXUS_version`</u>	CHAR8	Revision of the NEXUS format used by the file (this document describes Revision: 1.24).
`experiment_description`	CHAR8	Bibliographic reference to a description of the experiment
`owner_name`	CHAR8	Name of the principal user who owns the file, e.g. `John Q. Public`
`owner_mail`	CHAR8	A traditional mail address of the principal user, e.g. `Chemistry Dept., Berkeley U., CA 11001`
`owner_email`	CHAR8	e.g. `joe@pdp8.phys.mit.edu`
`owner_phone`	CHAR8	e.g. `900-555-1212`
`owner_fax`	CHAR8	e.g. `900-555-1213`

[5]`http://physics.nist.gov/Pubs/SP811/contents.full.html`

[6]The Vgroup class name is supposed to imply what kind of information will this Vgroup contain. For instance, a Vgroup "`IN5`", with class name "`NXinstrument`" is expected to describe a measuring device, and will contain appropriate data items describing the instrument's components.

[7]It would be more elegant to create an SDS called `owner` and encode the data as attributes of this SDS; however, there is no way to conveniently access such SDS (HDF doesn't provide a way to request top level data sets that aren't connected to any Vgroup).

2.4 Experiment data sections

The main part of a NEXUS file is constructed as a hierarchy of data groups, using the structuring facilities of the HDF format (c.f. Fig 1). The top-level components are data Vgroups with the class name `NXentry` and appropriate specific names (e.g. "`Run 123`", "`LaBaCuO`" or "`Background run`"). Each such group contains full and complete information about the respective scan. We are aiming to specify the data that are commonly believed to be necessary for an exhaustive analysis of a neutron scattering experiment. At the same time, we realize that there may always be something that the standard omits: fortunately, the extensibility of the HDF format means that such a variable can always be added without having to modify the existing reader programs. We would, however, encourage people who find it necessary to add their private variables to announce this fact and to seek inclusion of their additions into the officially recognized list; otherwise these additional data items will not be recognized nor used by other people's programs; additionally, their names may conflict with names introduced by others.

The actual number of data groups contained in a NEXUS file is to be determined locally by the implementers of the writing program. Some people may decide to bunch together many groups corresponding to consecutive scans, but most likely implementations will include no more than a small number of scans per file—presumably because such scans are closely associated[8].

The complete scan data should include the instrument and sample information, and the actual scan parameters and scan data—c.f. Fig.1. Since NEXUS aspires to be an exchange format, the information that is "well-known" at some site, and therefore traditionally not included in data files, should still be included in a NEXUS file.[9] To avoid the unnecessary duplication of such information in files with multiple data groups per file, HDF elements (SDSs and Vgroups) can be cross-linked to multiple parent Vgroups. Thus, a proposed `instrument` Vgroup, describing the parameters of the instrument, is included only once and then linked to all data Vgroups, with minimal overhead (c.f. Fig. 1).

Those who are worried about duplicating bulky information in separate data files, must decide a tradeoff between simplicity and efficiency. The price of data storage seems to be in a never-ending fall, and is currently much less than a dollar per megabyte; thus it makes sense to include all possibly relevant data in the data file.

2.5 Instrument description

The experimental apparatus is described by a Vgroup of class `NXinstrument`, named after the instrument itself. This Vgroup is composed of some general-purpose descriptive SDSs (`location`, etc.), and of Vgroups describing individual parts of the instrument. The general SDS are:

[8]The advantage of many small files is that they are easier to manage: the computer file system provides "database record" (i.e. individual file) manipulation capabilities such as deletion, copying, listing, etc. If a NEXUS file contained many data records, programs to perform these record management facilities within the file would have to be written.

[9]An example might be the distance from a chopper to the sample; even though it may be constant and easily accessible locally, a portable NEXUS file might end up in a place where this quantity is unavailable.

SDS name	Type	Description
location	CHAR8	Location of the instrument, e.g. `NIST/CNRF`
description	CHAR8	Bibliographic reference to a description of the instrument
type	CHAR8	Description of instrument type, e.g. `TOF`, `trash`, `TAS`
local_contact	CHAR8	Name of the collaborating instrument scientist

The instrument components are described by Vgroups with class names describing the component. If there are two units of the same type in the system, e.g. two blocks of collimators, they would be called `collimator1` and `collimator2`, although descriptive names such as `collimator_before_sample` and `collimator_after_sample` would be preferred.

All instrumental components may have the following SDS members:

`name`

`equipment_identification` describes the particular unit (e.g. an X-ray tube might be described as `Siemens 3487 SN # 12345678`).

`distance` distance(s) from the instrument element preceding the current one

`position` element-specific position information, e.g. angular positions of a multi-detector setup

After [3] and [2], we define the following components:

Vgroup class and an example name	member SDS
`NXreactor NIST reactor`	`power, flux`
`NXX-ray_tube Rigaku Cu`	`current, beam_size`
`NXaccelerator IPNS`	`energy, current`
`NXundulator U-5`	`energy, gap`
`NXfermi_chopper monochromator`	`wavelength, diameter, rotation_speed, phase`
`NXdisk_chopper monochromator`	`wavelength, diameter, rotation_speed, phase`
`NXcrystal_filter monochromator`	`wavelength, crystal, mosaic, reflection, d-spacing`
`NXBeryllium_filter analyzer`	`wavelength, mosaic, temperature`
`NXsample_table sample`	`elevation, temperature, pressure`
`NXsoller_collimator collimator`	`collimation, height, length, widthn`
`NXpinhole_collimator collimator`	`collimation, height, length, width, start_diameter, end_diameter`
`NXmonitor beam_monitor`	`number, size, transmission, efficiency`
`NXdetector detector`	`number, size, efficiency, pixel_size, efficiency`

2.6 Sample description

The sample is described by a Vgroup, named after the specimen, and with a class name - `NXsample`. This Vgroup's contents describe the physico-chemical nature of the sample. Only the values that remain constant throughout the whole scan should be entered in this Vgroup.

SDS name	Type	Description
description	CHAR8	Bibliographic reference to a description of the sample
form	CHAR8	One of `single crystal`, `polycrystal`, `powder`, `thin film`,`multilayer`
shape	CHAR8	One of `sphere`, `plate`, `cylinder`
unit_cell	6*FLOAT32	Crystalline cell parameters (dimensions and angles)
unit_cell_error	6*FLOAT32	Error of crystalline cell parameters (dimensions and angles)
structure_factor	2*FLOAT32	Complex structure factor for the unit cell
orientation	9*FLOAT32	Orientation matrix
absorption	FLOAT32	Sample absorption.
temperature	FLOAT32	The sample temperature, `units=Celsius`
pressure	FLOAT32	Pressure in a sample cell, `units=Pascal`
magnetic_field	3*FLOAT32	Magnetic field applied to sample (in the same co-ordinate system in which the orientation matrix is expressed) e.g. `0.5, 0., 0.` with `units=Tesla`

As mentioned earlier, if there are multiple values of some quantity (e.g. a multi-phase sample with several unit cells present), the corresponding SDS will have an additional dimension and multiple values will be stored in consecutive slices indexed by this dimension.

Analysis mode and scan intent Two useful SDS that should be present in each `NXentry` Vgroup are `analysis` and `intent`. The `analysis` SDS is designed to help in automatic analysis or display of data, although, of course, it can be overridden by explicitly specifying different operations. For instance, the reflectivity data could be automatically reduced and fit using the appropriate program; clearly, a different procedure should be applied to crystallography or TOF data. Values suggested for neutron scattering use are:

Value	Description
`general`	general Q-E space scan
`diffraction/misc`	for elastic ($\delta E = 0$) data scans along theta,hkl,Q.
`diffraction/peak`	like diffraction/misc, but requests peak analysis
`diffraction/powder`	only for powder diffraction scans
`trash`	trash mode data
`TOF`	time-of-flight data
`reflectivity`	reflectivity data (implies monitor scans)
`reflectivity/PSD`	reflectivity data with off-specular component from a PSD counter
`crystallography`	mostly integrated intensities, scans may or may not be stored
`NONE`	status of something, no analyzable information (e.g. slit status)

The `intent` SDS categorizes the data by the purpose of each run. Again, this is for informational purposes only; it makes it easier to, e.g. select all adjustment runs for plotting. Suggested values are:

Value	Description
`alignment`	alignment scans, not to be processed for data.
`calibration`	calibration information
`data`	actual data that is to be analyzed and used in research
`status`	status of something, no analyzable information (e.g. slit status)
`misc`	miscellaneous
`testing`	generic instrument testing
`centering`	centering reflections to obtain orientation matrix

The `intent` SDS may have an attribute `index`, describing current data in general physical terms. For example, it could contain values of h, k, l around which the scan is being done.

Other SDS in the `NXentry` Vgroups might be:

SDS name	type	Description
start_time	CHAR8	ISO time when scan was started, e.g.`1996-07-31 21:15:22+0600`
end_time	CHAR8	ISO time when scan has ended
title	CHAR8	A title, something you would like to see on a plot. A good thing to put here is the reason for this particular scan. e.g. `measure mosaic width`
program_name	CHAR8	Name of program writing this data. Use attributes for version. e.g. `ICP`
command_line	CHAR8	Command line used to initiate the current scan e.g. `scan energy from 10 to 10.3 keV, 10 steps`
abort	CHAR8	Flags a scan that was aborted. This is mandatory for incomplete/aborted scans. It should not be present for properly completed scans. Use the text part to indicate what was wrong, e.g.. `forgot to open shutter`, or `hard limit on theta`, or `hard limit on monochromator`
constraint	CHAR8	Describes what is held constant by the scan (useful on plots) e.g. `chi=25` or `Hkl=(0 0 1.2), Ef=30 MeV`
plot_type	CHAR8	Flags preference for plottable data. Proposed types are `line`, `3D`, `contour`, and `image`. This tag is not required since the structure of the data itself suggests a default. i.e. a raster image should default to a picture, a 1-d scan defaults to line plot, etc.

2.7 Scan data

Actual scan data are contained in a separate Vgroup containing data sets with scanned variables. All variables changed or measured in the scan (plus perhaps some derived variables, if desired) are included in this section, represented by vectors of values. Those SDS shall be described by attributes such as units, description of physical meaning of the variable, comments for plotting and other automatic manipulation of data.

Note

Since each element of a data vector corresponds to one data point, all vectors will have the same length, equal to the number of measured data points.

Note that if the appropriate variable does not change during the scan (i.e. it is neither scanned nor nominally measured at each scan point), the corresponding SDS should be in the `entry`N Vgroup rather than in `data`.

Some SDS names that might be present in `data` are:

SDS name	Type	Description
temperature	FLOAT32	The sample temperature `units = Kelvin`
pressure	FLOAT32	Pressure in a sample cell. `units=Pascak`
magnetic_field	3*FLOAT32	Magnetic field at sample e.g. `0.5, 0., 0.` with `units=Tesla`
Q	3*FLOAT32	Q space position e.g. `0.5, 0., 0.` with `units=2pi/d`
counts	FLOAT32	neutron counts in the detector

Besides physical attributes, such as units, the data in the `data` Vgroups may be annotated with the following attributes:

Attribute name	Type	Description
primary	INT16	Indicates importance of column for plotting. 1 is most important. Not all columns should have a value. Only ONE SDS can be primary=1.
axis	INT16	Indicates the default independent axes for plotting (1=x-axis, 2=y-axis, ...,). Values between 1 and rank of data must be defined; additional values may be used to define, e.g. additional axis on plots. Only one SDS can have each value.
count_time	NONE	Attached to the column which contains the time counted (in some form). Only its presence counts. Only ONE SDS can be `count_time`. Be sure to set units to be able to get count time into seconds.
equation	CHAR8	Its presence indicates a derived quantity. the text gives the equation used.
monitor	NONE	Attached to the column with best intensity measurement before sample. Only its presence counts. Only ONE SDS in an entry may be `monitor`.
flipper	CHAR8	Status of flipper (only used for neutron data) e.g. `on` or `off`
FWHM	2*FLOAT	FWHM and error of SDS with axis=1 as x-axis (=2 for y-axis)
integral	2*FLOAT	Integral and error of SDS with axis=1 as x-axis (=2 for y-axis)
plot_expression	CHAR8	Expression that can optionally be plotted together with data (e.g. a guide for the eye, or a pre-fitted model curve).
voigt	14*FLOAT	Fit to a Voigt function with axis=1 as x-axis. Parameters are: `background_offset, slope, amplitude, x_position, shape, FWHM, integral`. Integral followed by errors in each of these values.
gaussian	14*FLOAT	As above, for a gaussian peak
lorentzian	14*FLOAT	As above, for a lorentzian peak

Here is how [2] describes the plotting attributes `signal, primary and axis`:

> These named attributes are used to identify the role of individual SDS's. This provides the plotting and analysis programs a means to process the data. This indirect reference method (using signal, primary, axis) was chosen since there may many input channels that are collected in each scan, and the `primary signal` may change depending upon what the user is interested in for that scan. For example, the user has both a scintillator and a PIN diode operating. `signal` will be `scintillator` or `PIN` depending upon which detector is in use at the moment. Thus when `signal` is `PIN`, the PIN column should be plotted. The same reasoning applies to the independent variables. For a scan in reciprocal space along the [010] direction where h,k,l,theta,2theta,chi,phi are all collected, it is necessary to identify

that the k column is best choice for x-axis. If the scan were along the [011] direction, then a computed axis (not actually measured) would be the best choice.

Attribute defaults The multiple linking of SDS/Vgroups takes care of most cases where redundant data storage is to be avoided. One exception is when multiple `entry`N Vgroups contain different information but with similar attributes. For example, a NEXUS file might contain many short $\theta - 2\theta$ scans, with different values of θ in each scan. The attributes of θ (i.e. units, scale, etc.) would, nevertheless, be the same, and would have to be stored each time, possibly resulting in a significant overhead.

In such cases, a special *optional* mechanism is proposed, whereas a `defaults` Vgroup would be created (once) and linked in to every entry Vgroup that needs default attributes. This Vgroup would contain dummy SDSes with no data, that only provide default attributes for identically named SDS in this entry. This mechanism was suggested by Tischler [2], who describes it thusly:

> HDF does not include a provision for default attributes (or any other kind of defaults). However, their use is desirable, especially when the data file consists of many small scans where recording a large number of attributes for each one would noticeably increase the file length. Although HDF does not explicitly provide for default attributes, it is possible to include them while maintaining a legal HDF file. Since this default information is not part of the HDF standard, generic HDF programs (such as Mosaic, Spyglass, ...) will not recognize the meaning of this default information. However, these generic programs will still be able to process the files, since they were written using standard HDF routines. As a result of all this, defaults will be most useful for attributes that are defined in this standard (e.g. diffract_axis), and least useful for attributes that are generally recognized by all HDF programs (e.g. units).

3 Conclusions

There is still some work to do before this proposal is formalized. We invite comments and suggestions from all interested parties.

Certain trade names and company products are mentioned in the present paper. In no case does such mention imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

References

[1] "HDF Reference Manual", NCSA, February 1994, available electronically from `ftp.ncsa.uiuc.edu:/Documentation/HDF3.3`

[2] Jonathan Tischler, ORNL & UNI-CAT, "Proposed Data Standard for the APS (v4)", June 6, 1994

[3] Mark Könnecke, Rutherford Appleton Laboratory "Proposal for a European Neutron Scattering Data Exchange Format (v4.2)", June 17, 1994