# NXnet proposal - Your data, my data, collected anywhere, available anywhere

**Chris Moreton-Smith**
***ISIS Facility,***
***CCLRC Rutherford Appleton Laboratory, Chilton, Oxon. UK***

## *Abstract*

***The purpose of the NeXus data format is to facilitate the exchange of Neutron, X-Ray and muon data (N/µ/X). Interest has recently been very much focused on agreeing common instrument definitions. Clearly these are key to the successful exchange of data. Also of great importance however and recently somewhat neglected is the ability to exchange the NeXus (and other data) files themselves by anything other than ad-hoc means (ftp, web download). At ISIS we have been testing a very fluid means of data distribution using the San Diego Supercomputing Centre's Storage Resource Broker (SRB) software. We believe it is now possible to create a global fully distributed storage network for Data files which can be accessed in a uniform manner from wherever the scientist is located and on whatever platform. As the technology is surprisingly simple to implement and can be added incrementally to existing facilities, this paper proposes that we adopt SRB to establish a data exchange network which links our facilities and which supports in particular the needs of neutron, X-Ray and muon science. To give it an identity associated with NeXus aims, we suggest the name NeXus-Net or NXnet for short.***

# Introduction

In a world where good networking and distributed computing are becoming commonplace one of the most exciting paradigms is that of a globally distributed file system, secure, but useable by anyone from any location on any networked computer system. One promising candidate technology for realizing this dream is the San Diego Supercomputing Centre's Storage Resource Broker[1] (SRB). In this paper, it is proposed that the neutron, muon and X-Ray community take the lead to implement an SRB system for storing and distributing their facility user's data.

The emerging NeXus[2] standard for exchanging of N/µ/X data already addresses "in file" standardisation; this paper also proposes that the process of organizing the file system and meta-data is made part of NeXus.

Sometimes the most simple concepts seem to be the ones that are hardest to see and yet once envisioned become so compelling as to be impossible to deny. The idea of a global file system in distributed computing appears to be one of these things. The aim of this paper is not to go into great detail about the workings of SRB – even a short experience of using SRB starts to make the idea of a global file system "obvious" – instead, we assume that this (or a similar system) is certain to as familiar to us in a few years as the web is now and go on to look at some of the issues which need to be addressed. In particular;

1) How the *N/µ/X* community can benefit by becoming an early adopter of this technology.
2) How we might as a community select some standards to keep the organisation of our part of this file system easy to navigate and find data in.
3) How we might as a community decide and regulate what metadata from within files is made publicly searchable.

# A Global File System for N/μ/X data

Most scientists doing experiments on instruments at different establishments will have experienced the joy of trying to copy data off a variety of different computer systems, often being forced to network a laptop machine at the last minute, write a CD or floppy disk before racing to catch a flight, manually select and copy files one at a time via ftp and/or negotiate a firewall to get at their data. If very lucky, maybe they have been able to download data from a conveniently set up web site, where remembering the password has been the only problem.

If this state of affairs wasn't bad enough, there are things that are even harder to do than read your data. For example, you might want to send a few calibration files and sample setup notes to a remote site ready for an experiment. This would normally be seen as folly unless you have a trusted colleague already on site to help you receive and look after the files. And what if you've forgotten a file you needed on arrival? Wouldn't it be nice if you could do a bit of data reduction on site and then continue at home, all the time saving the reduced files in the same directory as the raw data (and with no need to copy the data locally). And then permit a colleague to access a few of the files from their own laptop whilst at home (by setting permissions like you do on a local file system)?

These are some of the things which a shared (globally distributed) file system such as SRB could achieve. Let's examine the modus operandi of such a distributed file system by examining a scenario from the point of view of a facility user.

## *Scenario*

A user in the USA has a proposal for an experiment accepted on an instrument at ISIS in the UK. He browses through the SRB file system to locate a directory which was created for him under the data area for the instrument he is planning to work on and which has been created for his upcoming experiment. He creates a subdirectory ("…ISIS/HET/RB123456/notes") and drags a couple of documents with notes on how to perform the experiment into this area – a post doctorate and a German colleague of his will actually perform the experiment and will need to have access to these procedures for sample preparation on site.

A couple of months later, the post doctorate arrives on her own to perform the experiment at ISIS. In the cabin whilst setting up the experiment she browses the "…ISIS/HET/RB123456/notes" directory but realises that one of the calibration files referred to by the notes has absent-mindedly been left out of the area. No problem, she simply re-directs the SRB browser to her supervisors area at home in the US (which she has access to) and drags the file to her local desktop in the cabin (luckily her supervisor has for a while stored all his documents in SRB making them available to suitably privileged users.

Whilst running the experiment, the NeXus data files are stored automatically by the instrument software to the "…ISIS/HET/RB123456/data" directory. After a few runs some results give the experimenter a problem, she can't understand a particular feature of the data and really needs to talk to her German colleague who was going to do the experiment with her at ISIS. Unfortunately, he is at home in Germany, his wife gave birth unexpectedly and so he was unable to visit the UK as planned. When he gets a phone call, he logs onto his home machine and starts his SRB browser. He quickly locates the raw files where they are being collected in the data directory and offers to run some data reduction and analysis codes on the data from home. The results are stored in the "…ISIS/HET/RB123456/data/reduced" area. An hour or two later, both communicate on the phone again and they spend some time visualising the data together and finally resolve the problem. The German colleague copies a script he used last year at ISIS into the current "…ISIS/HET/RB123456/scripts" directory and modifies it to run overnight (it was stored with the data from his last experiment at ISIS and remains in the SRB system so he can find it easily).

# *A proposed architecture*

So far we have talked about SRB somewhat in abstract. In this section we examine how to go about setting up NXnet, a global file system shared between the major world facilities of the neutron, muon and X-ray communities. Does this sound like a major undertaking? Not necessarily. When the world wide web first came into being, all that was really necessary was to set up a web server and to get a client (NCSA Mosaic mainly) to view it. The high energy physics community set up web servers at their sites and looked at each other's sites with the browser.

SRB is not that different, there are clients and servers and both are quite easy to set up. To implement a common file system however, it is important that the servers themselves are linked to each other explicitly (not a requirement of web servers).

Here we look briefly at the pieces of the system in a little more detail, then propose a basic architecture.

## Clients

SRB is essentially a client/server system, there a several clients for accessing the file system. Those of most immediate interest are:

1) Command line – a set of command line file system operations "S commands" based on the Unix command set. For example:

| | | |
|---|---|---|
| Sls, Spwd, Srm, Smv | - | Similar functions to Unix but within SRB file system |
| Sget, Sput | - | Exchange data files in SRB with local files |
| Sreplicate | - | Functions specific to SRB |

2) PC browser Windows (INQ) - Mimics explorer drag/drop interface on the PC.

3) Web interface (MySRB) - Allows access into the SRB system from anywhere without the other clients.

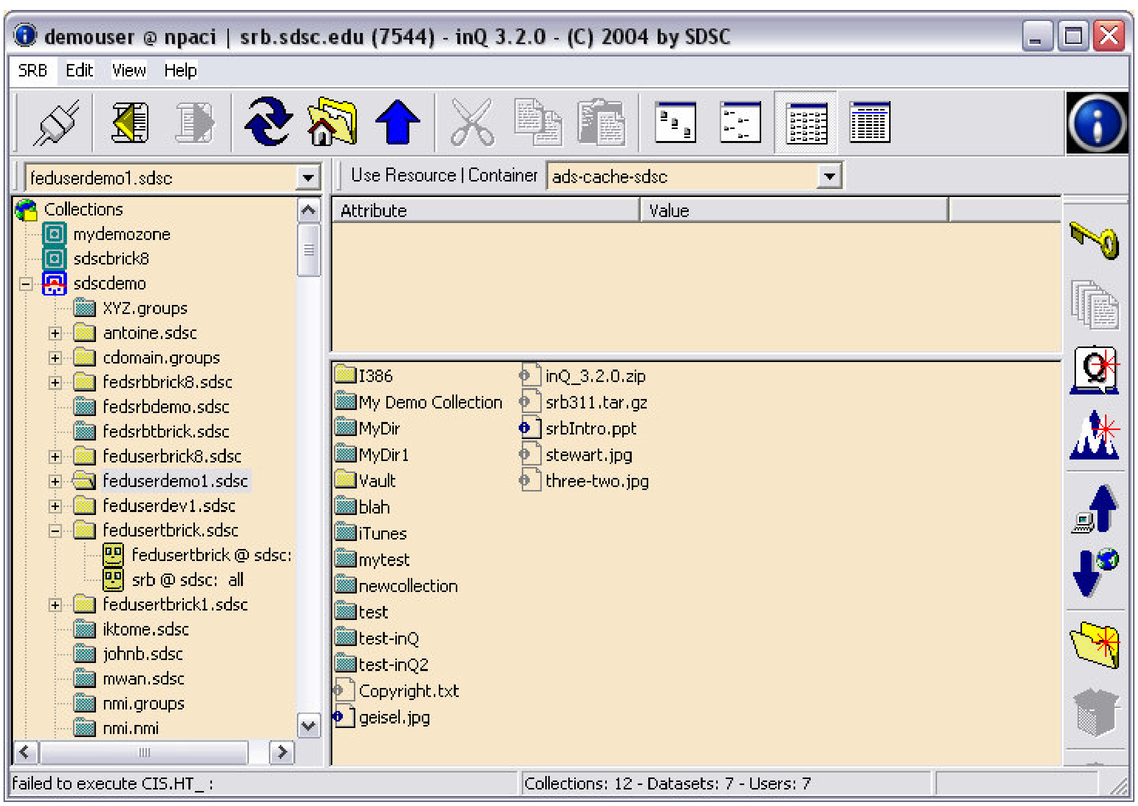


**Figure 1 - INQ browser interface**

## Servers

Servers are generally run on machines local to *data resources* which are to be served as part of an SRB file system. For example, if I have a server running on Linux, I can serve a Linux disk resource. Likewise, a server on NT can serve an NT disk resource. From the SRB client point of view, these two disk resources are simply two alternative locations where I can put files interchangeably. A server may also be linked to a driver specialised for a particular resource, for example a tape drive or optical disk jukebox system.

For each area or "zone" in which SRB is being used, there must be at least one "MCAT" enabled server. This server interfaces with an MCAT database which is normally maintained locally. It is in this database that all file system metadata, user accounts and permissions are stored.

Once identified to each other, SRB servers communicate with each other to provide access to data as requested by a client. The geographic location of servers is only important in terms of the efficiency of data transfer. For example, if an SRB server is running in Australia sharing a zone with the UK, files could be requested through any server in the zone. For efficiency, it would make most sense to request files to be used in Australia and residing on Australian data resources via a server in Australia. For files residing on Australian resources to be used in the UK it would not be too important which server they were requested from as there would be a transfer of files to the UK at some point. By requesting from an SRB server in the UK, files would be transferred within the SRB system, and, if this state of affairs was to continue, a *replica* could be generated of these files on a UK SRB server.

## NXnet – how it might be structured

The discussion about SRB servers leads on to how we might structure a network of SRB servers to usefully link the major neutron, muon and X-ray facilities globally.

One of the most important facilities added in the version 3 release of SRB was the ability to federate between multiple zones[3]. This means that each facility can set up an SRB zone to store its own user's data on its own storage resources. By running a single MCAT database for the facility the facility implements a unified SRB file system over machine architectures and storage resources ranging from major data stores to local IDE disks. Not only this, but data can be provided seamlessly from SRB to any facility users wherever they are in the world via one of the clients.

This is good but only half the story, how much better might things be if we were to work within a *unified* SRB file system created by federating with the zones of our sister facilities. With an infrastructure such as this, it then becomes possible for us to envisage the sort of example scenario discussed earlier. All this can occur without a facility having to surrender control or management of its own data resources. ISIS data files for example would remain on ISIS file servers in the UK but might be listed next to files from the APS in the US – in the same directory - which likewise would remain on APS servers within the US.

Because SRB allows us to set up servers incrementally, this paper proposes that one of the things we do is set up a test network between a few interested establishments who already have or will have a need to provide user's data online. Each participating establishment as it joins creates their own zone federated by MCAT replication with NXnet and runs one or more SRB servers on their local resources. If the test network behaves successfully, this can then be moved to a production system. During testing, data should be considered volatile and users warned of this fact.

# SRB and NeXus

## *NeXus Standard for storing/locating files*

The scenario in the previous section describes the sort of benefits which can be realised when using a global file system, and hopefully demonstrates some of the power of this sort of system. With power comes the need to give some sort of guidance in terms of organisation so that people are still able to find things.

In terms of facilities wishing to make data areas available before and after experiments, it greatly helps to agree on some conventions which would allow a user to find data easily (browsing access between different facilities becomes much greater). These can be quite simple and a few suggestions might be:

1) Store all data in NeXus files and use the extension ".nxs" to denote them.
2) File names should consist of a six character facility id, six character instrument id, run number and date in ISO format.
3) Ensure that for each facility, data is stored within a path of this form /facility/instrument/experiment-number/rawdata.
4) Calibration data to be stored in /facility/instrument/calibrations/year/data

These are just examples but it can be seen that this sort of standardisation is similar to the effort put into defining common structures within NeXus files. As such it seems logical to charge the NeXus International Advisory Committee (NIAC) with involving and guiding the community to workable standards.

# *NeXus Standard for metadata*

SRB makes provision for storing metadata along with the file itself; it even provides tools to add standard metadata such as Dublin core[4] to any file. The metadata is generally not expected to be a very large amount of data (and need not be present at all). It is likely though to be data which a search engine could pick up without searching through gigabytes of raw data. Possibly critical or very useful data from within the file might be added to the metadata as well as general book keeping information. There are three particular issues with metadata for consideration, content, security and format.

## Content – What count's as metadata?

Generally, metadata is information which is handled in some different way from the normal data content of the file. For example, NeXus files would normally contain some user information and information about the title and nature of the experiment such as samples and conditions which pertained during the experiment. Some of this is numerical in the NeXus file but would be better formatted as a string and duplicated in the meta-data for a search engine or human readability.

Defining what sort of items should be stored "externally" to the NeXus file as searchable meta-data for that file is another issue for the NIAC. It is important that some standardisation occurs again to make working across different facilities and different branches of science possible.

## Security – What metadata should be searchable?

Given that useful meta-data content can be defined, another sensitive but important issue is what and how much meta-data should be made available for public consumption and searching. If we look at the world of publishing, quite a lot of metadata is always made available for books and reports (Title, author(s), dates of publication, ISBN). The Dublin core is one standardisation effort in this area and the implication is that this data is always publicly available, even when searching the content of books themselves might not be.

It is not guaranteed that a user performing an experiment can identified in meta-data or even in the file currently. Ultimately, some of these things will need to be written in the data policies of each establishment. Again, a lead from the NIAC would be extremely helpful here.

## Format – How should metadata be represented

There are several choices as to how metadata can be formatted. It may be represented as ASCII text pairings (keyword with value), free form text, some form of XML or in an XML based language specifically designed for representing metadata. One such language is RDF[5] (Resource Description Framework), this is seen as forming the basis of the "Semantic Web" by the originators of the World-Wide-Web and is perhaps one of the best ways currently to represent meta-data.

# The Proposal

The motivation for NeXus is to allow the exchange of data between programs, instruments, establishments and branches of science where the neutron, muon and X-ray technologies are the common theme. Recently the NeXus community has been greatly (and crucially) occupied with choosing what data to store in a file and how it should be described in a common fashion.

This proposal tackles a complementary facet of the broader problem; namely the exchange of the data files themselves between the programs, operating systems, instruments and establishments.

It is proposed that the NIAC look at this area of standardisation, particularly the file/path naming and meta-data issues that will arise so that establishments can work to some common conventions for locating/naming files.

The proposal is in three parts.

1) We adopt SRB as a working system on which to experiment with building a global integrated network for sharing Neutron, Muon and X-Ray data between our establishments and our users. We do this pragmatically (like we have done with HDF) because it currently seems to do the job and support is what standards like this need to develop. As we have a reasonably small number of facilities, creating a network of this sort is quite a plausible undertaking.

2) More fundamentally, we extend our remit of defining and organizing data types within the NeXus file to also giving some sort of standardisation to the organisation and location of data within a global file system. Quite simply, this just avoids things being lost by everyone storing things under different names and in different places (for example, a naming convention for raw files

3) Even more fundamentally, we spend some effort defining the sort of meta-data which we might associate with each file (possibly not contained in the NeXus file itself). This meta-data would enable a data portal style search engine, just like a super data-Google quickly to find relevant data by searching throughout this global file system. Some of this sort of work is already underway but some sort of standardization of the type and contents of this metadata is very close to the sort of standardization we are aiming at with the NeXus file contents and would greatly ease the ability to search and find relevant information.

## *Conclusion*

SRB Provides one example of a paradigm which is going to become common place, that of a global file system in which anyone can store/retrieve data without regard to the current file system of the local machine. Because of the real benefits provided by this sort of system to our users, the paper proposes that we leap on the opportunity provided and implement an SRB system between our establishments. We also ask the NIAC to look at some broader "out of file" issues which are in many ways similar standardisation issues to those the community is already working on.

---

[1] Storage Resource Broker main web site for documentation (see www.npaci.edu/DICE/SRB/ for more information).
[2] NeXus data format for storing neutron, X-Ray and muon data (see www.nexus.anl.gov).
[3] An Overview of the SRB 3.0: the Federated MCAT. Michael Wan, Arcot Rajasekar, Wayne Schroeder (see www.npaci.edu/DICE/SRB/FedMcat.html).
[4] The Dublin Core Metadata Initiative (dublincore.org).
[5] Resource Description Framework (RDF) W3C specification see (www.w3.org/RDF).