H. Perez Ponce[1], H. Görzig[1], R. Krahl[1]

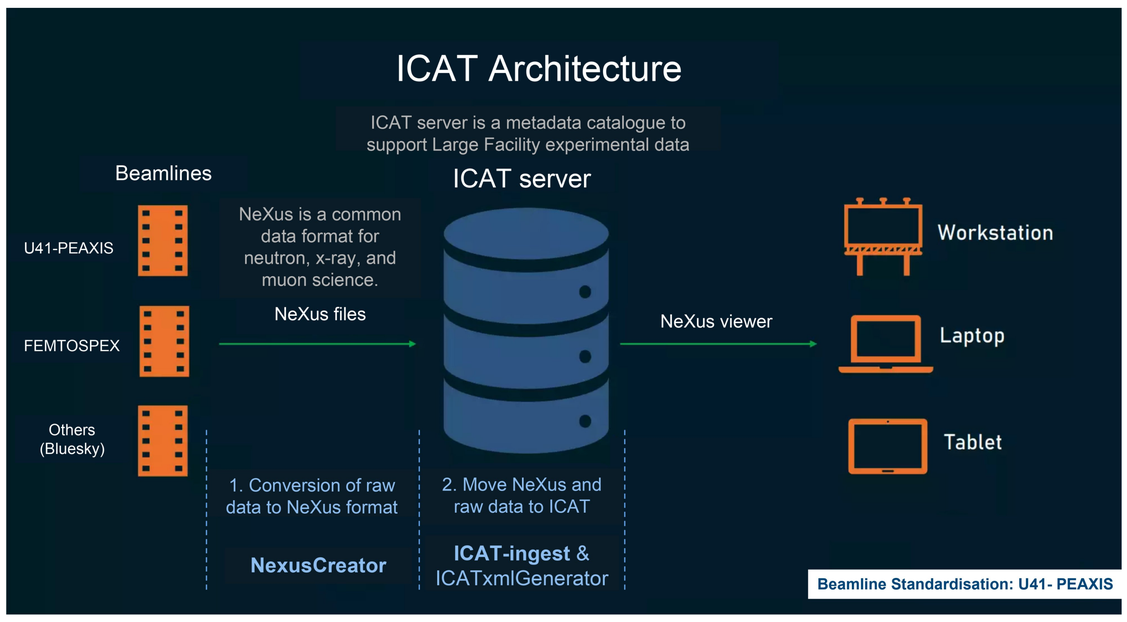
[1)] **Helmholtz-Zentrum Berlin für Materialien und Energie, Hahn-Meitner-Platz 1, 14109, Berlin**

NEXUSCREATOR & ICAT - HZB APPLYING FAIR DATA MANAGEMENT

ABSTRACT: The research data management group at Helmholtz-Zentrum Berlin is implementing FAIR data management. Data starts to be transitioned from specific file formats to standardized NeXus/HDF5 files. The standardization program encompasses both the conversion of previously generated data, and the automation of NeXus files creation for new experiments, such as those conducted with Bluesky. Our tool, NeXusCreator, allows the separation of the standardization process in two parts: 1) defining instruments and application definitions via NeXus standard, and 2) creating file converters or automated generation processes. NeXusCreator comes in two versions, Python and Javascript.

Motivation

The beamlines at BESSY-II are independently managed, with each beamline storing its data using distinct data structures and a variety of file formats, such as SPEC, SIF, DAT, XAS, among others. The goal is to standardize both the data and the metadata across all beamlines by adopting NeXus as a common data format and applying FAIR data management. Furthermore, the data will be ingested into a long-term metadata catalogue (ICAT), enabling access for authorized users to read and utilize the data efficiently.

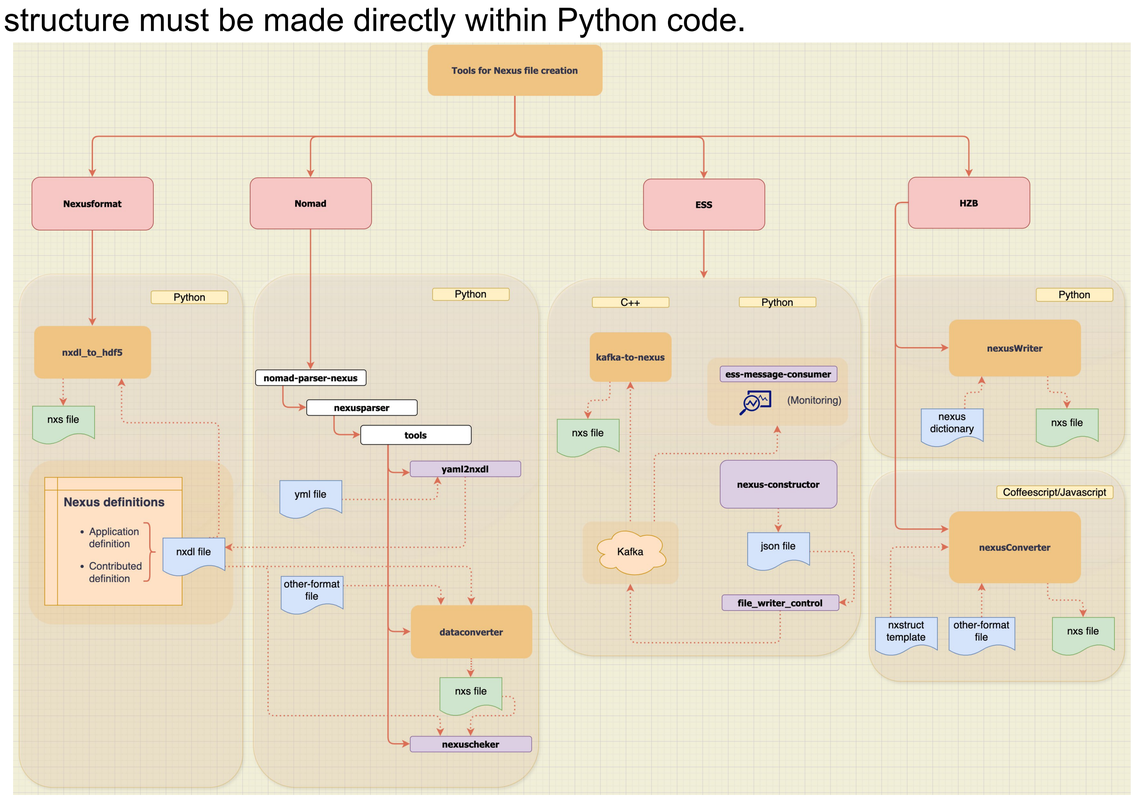


Existing Tools

A study was conducted to evaluate existing software for the generation of NeXus/HDF5 files. Four software applications were assessed based on their capacity to generate NeXus files without requiring programming. These software application are as follows:

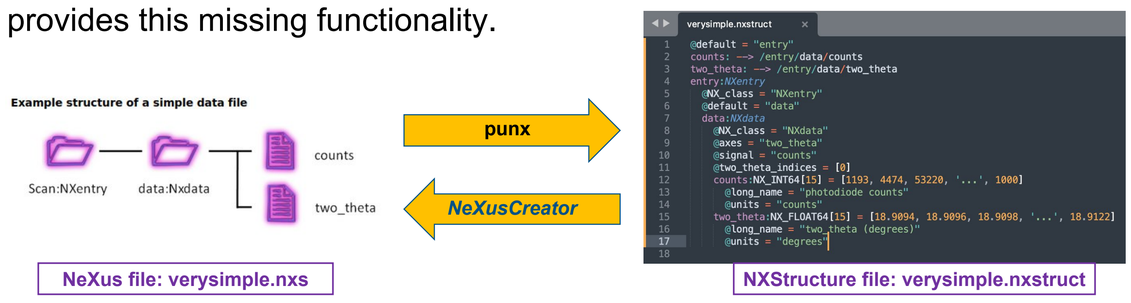
Software		Input file format	Advantage
nxdl_to_hdf5	nexusformat	NXDL	ASCII file as input.
yaml2nxdl / dataconverter	NOMAD	YAML	Import of datasets from other file formats.
nexus-constructor / kafka-to-nexus	ESS	JSON	Nice graphical view of NeXus structure syntax.
NexusWriter	HZB	Python dictionary	Simple operation.

Tools **nxdl_to_hdf5** and **yaml2nxdl / dataconverter** provide the capability to generate NeXus files from NXDL and YAML scripts. These tools facilitate users without programming expertise to work with NeXus files, with YAML being simpler than NXDL. However, the use of YAML is not entirely straightforward. **NeXus-constructor** is an excellent tool for designing the NeXus structure, although editing an existing structure is somewhat challenging, as moving elements is not supported; users must delete and recreate them. **NeXusWriter** offers a simple operation, but modifications to the structure must be made directly within Python code.

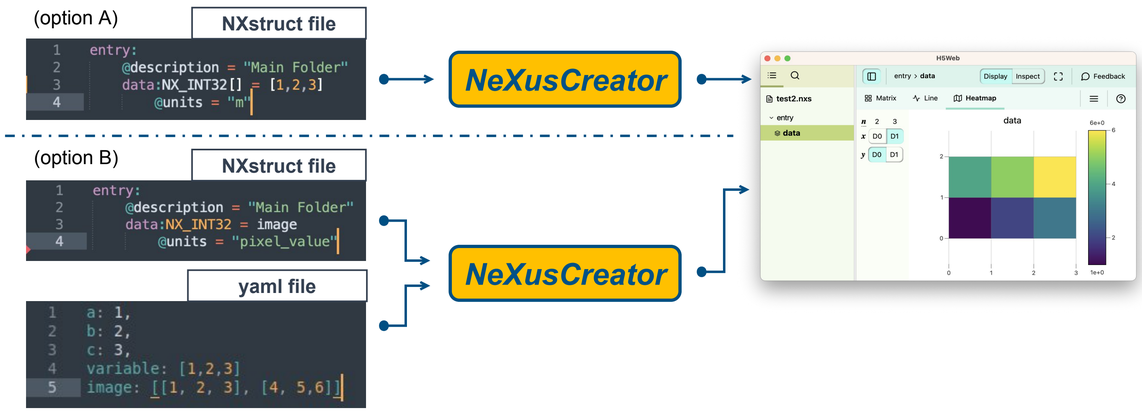


NeXusCreator

In the tutorial available at www.nexusformat.org, examples of NeXus files are presented, where the structure is shown using a simplified ASCII tabular syntax, which I will hereafter refer to as “NXstruct”. Unfortunately, no tool currently exists that allows this syntax to be used as input for generating a NeXus file, even though it would be more straightforward than using NXDL or YAML. Conversely, a tool called **punx** does exits, which enables the extraction of the content of a NeXus file into NXstruct syntax. To address this gap, we have developed a tool at HZB, named ***NeXusCreator***, which provides this missing functionality.



NeXusCreator reads an NXstruct file and generates a corresponding NeXus file. Datasets can be included directly within the NXstruct file (Option A in following figure); however, it is more practical to use variables instead (Option B in following figure) and generate a dataset dictionary from any other source. This source can be an external file, such a YAML, SPEC or any other file format, or a connection to another system.

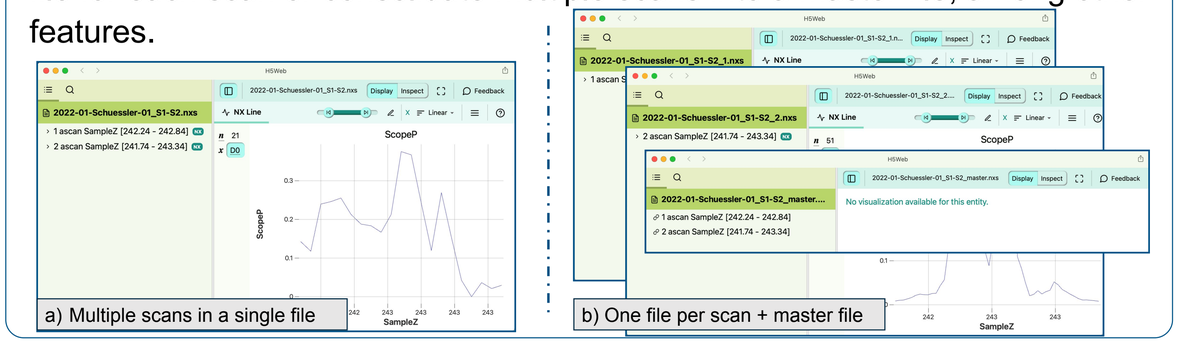


NeXusCreator is composed of independent readers and converters, each with similar underlying structure. This modular design allows ***NeXusCreator*** to read files in various formats or to integrate with automated infrastructure, such as **Bluesky** or **SECoP**. The converters can be customized for each beamline to utilize multiple readers as needed, thereby generating a dataset dictionary that provides the necessary datasets for the final NeXus file.

NeXusCreator also enables the processing of directories containing subfolders or files, handling each file based on its extension. Additionally, it can organize the output NeXus files into the required folder structure, facilitating seamless import into **ICAT**.

Results & Conclusions

NeXusCreator is currently being used to convert data from five beamlines, including the ROCK-IT project. Its feature of managing the NeXus structure independently from the data conversion or acquisition process has enabled Data Stewards to focus solely on the creation of the Instrument Definitions and Application Definitions using the NeXus ontology. The conversion of SPEC files into NeXus format has made it possible to use analysis software, such as **PyMca**, to work with the data in a consistent manner. Furthermore, due to the versatility of ***NeXusCreator***, it is possible to generate a separate file for each scan or consolidate multiple scans into a master file, among other features.



MORE INFORMATION

Hector Perez Ponce, PhD
hector.perez_ponce@helmholtz-berlin.de

NeXusCreator
https://codebase.helmholtz.cloud/hzb/research_data_management/nexuscreator
https://codebase.helmholtz.cloud/hzb/research_data_management/nexuscreator-js