



EVIDENT

R&D | MAY 2025 | QUEBEC, QUEBEC

An Open HDF5-Based Format for Industrial Inspection Data with Experimental Integration of the Onion VFD for File Versioning

2025 European HDF5 Users Group Meeting

Presenter



Baptiste Gauthier

Specialist, System Engineering - R&D Evident Scientific, Quebec, QC, Canada

Working within the Ultrasonic Non-Destructive Testing & Evaluation (NDT/NDE) industry since 2016 PhD in Ultrasonic Imaging for NDT

Agenda

Who we are, what we do

The .nde File Format

File versioning

1

2

Who we are, what we do

Two Divisions

Life Science Solutions

Industrial Solutions

Developing tools that will prevent accidents





Inspections are carried out:

- During manufacturing
- During maintenance periods, shutdowns etc.



EVIDENT's Market Leading Products

Industrial Solutions



We serve key Industry markets & applications

Automotive



Electronics | Semiconductors





Metals

Oil | Gas & Chemicals





Aviation

Power generation



Environment | Natural Resources

EVIDENT's Market Leading Products

Industrial Solutions





2

The .nde Open File Format

Background and Vision

- Data is key and has a lot of value
- Increased need for collaboration and transparency
- NDE 4.0 is the next big leap



Background and Vision

- Major instrument manufacturers and solution providers dominate the market with proprietary technologies
- Smaller compagnies with specialized expertise and products must partner with the major players to **develop business systems**
- There is a need for an open development platform built on nonproprietary standards, to enable better collaboration between NDE companies and accelerate the deployment of NDE 4.0 business systems

UNIS Platform & .NDE File Format

The .nde Open File Format: introduced in October 2022 and already deployed



The .nde file format: Structure

A hybrid approach based on HDF5 + JSON





HDF5 for Heavy Data

- Storage and organization.
- Substantial datasets



JSON for Light Data

 Representation of NDT domain : physical elements and conceptual elements, relationships, etc.

Documentation and JSON Schema

- Publicly accessible on GitHub: <u>ndeformat.com</u>
- Sample files provided.



Data type

- Matrices of integers or float
- Mostly signal amplitude vs. time
- JSON as string dataset
- From few KB to GB





Why – The typical inspection scenario



1. ACQUISITON



Original file saved on instrument



XYZ.nde



Do some modifications, post-processing, annotations etc.

Keep the original file untouched and recoverable while saving modifications, annotations, analysis and post-processing.

Linear File History (no branching)



Avoid storing multiple copies of the same file, reduce storage burden.

The Onion VFD

- Introduced in HDF5 1.13.2
- Allows for creating versioned HDF5 files
- Version is defined by an open/write/close cycle
- Supports version rollback and access to previous data versions
- All modifications to the original version are stored in an ".onion" file



Oct 01, 2021		RFC THG 2020-02-10.v3
	RFC: Onion VFD	
	Songyu "Ray" Lu John Mainzer	
	Jacob "Jake" Smith	

Incre is a desire to introduce and track modifications to an HDF5 file while preserving or having access to the file as it existed prior to a particular set of modifications. To this end, this RFC proposes an Onion Virtual File Driver (VFD) as an effectively in-file revision management facility. Users will be able to open a particular revision of the file, read from and make modifications to the file, and write to file as a new revision. The name "Onion" derives from a mnemonic: the original file exists with data layered atop one another from an original file to the most recent revision.

1 Introduction

At present, the HDF5 library offers no support for version control or provenance management¹ – there is no mechanism to store unique batches of data in the same logical file space, nor to track when or by whom a particular modification was made. While this is not a significant deficit for most HDF5 applications, it is a major concern for experimental and observational data, where the original data must be preserved, and any changes tracked and attributed.

The primary reason for implementing this ourselves (as opposed to, for example, relying on an external version control program) is that doing so will allow for a significantly smaller footprint on the storage (e.g., disk). Major contenders of external programs such as SVN, Git, or Mercurial, would store each revision as a full-sized binary file of the file at the revision state – this is clearly unacceptable with large files (gigabyte-plus each) and many revisions.

An obvious and powerful way to address this use case is to implement an "infinite undo" facility along the lines of that offered by some text editors. In the context of HDF5, such a facility would allow reconstruction of earlier versions of an HDF5 file on an API call-by-API call basis, with each API call constituting a unique revision of the file. While this is doable, implementation would be a major exercise, and it would add significant complexity to the HDF5 library with the obvious implications for long term maintainability. Thus, this paper presents a sketch design for a simpler and cheaper option.

If we make the simplifying assumption that it will be sufficient to track changes on a per file openclose cycle, the problem becomes much more tractable. More precisely, it will allow us to address the revision control at a very low level of the HDF5 library, making it transparent to the vast majority of

¹ Throughout the remainder of this document, these terms – version control and/or provenance management – will be collectively referred to as revision control.

The HDF Group

Using the Onion VFD with .nde Files

Experimental implementation

Year No file selected Historic	Open fi 😒				
No dataset selected or no editor can take that data format!	File	No file selected	Historic		
		No dataset selected or no editor can take that data format!			

Weld_Plate_UT-sk90-4.1.nde
Weld_Plate_UT-sk90-4.1.nde.onion

Using the Onion VFD with .nde Files

What we learned

- The Onion VFD is still in an experimental stage
- Not a lot of documentation or examples available
- All modifications are necessarily stored within a separate .onion file

What could be improved

- Single file support
- Size of the .onion file
- Revision Metadata API
- Wider adoption and testing

File versioning: other alternative

Versioned HDF5

Community project. A versioned abstraction on to of h5py.



A_versioned_data group is added

/ versioned data/





- NDE File Format (.nde), developed by Evident, is an open, extensible data format tailored for the non-destructive evaluation (NDE) and testing (NDT) industry
- NDE is built upon the HDF5 container and augmented with JSON-based metadata
- NDT workflows would benefit from file versioning
- Experimental implementation of file versioning with the Onion VFD was explored with NDE Files
- Still need improvement but first implementation was very conclusive
- More feedback from the community on the Onion VFD would be welcomed

E VIDENT SEEING IS SOLVING