The tip of the Validation Iceberg

Addressing JHOVE-based file validation warnings

Jay Gattuso

National Library of New Zealand

New Zealand

Jay.gattuso@dia.govt.nz

0000-0002-1201-4149

Andrea Goethals

National Library of New Zealand

New Zealand

Andrea.Goethals@dia.govt.nz

0000-0002-5254-9818

Abstract – This paper describes the volume of JHOVE-based file validation warnings in the National Library of New Zealand's digital preservation repository, our motivation for addressing them, and the start of our automated workflow that seeks to significantly reduce the number of indicated files.

Keywords – File format, JHOVE, validation, preservation action

Conference Topics – Empowering Digital Preservation for the Enriched Digital Ecosystem; Building the Capacity & Capability

I. Introduction

The National Library of New Zealand {NLNZ) has been running its preservation program for around fifteen years, and its preservation system as a 'Business as usual' activity for over ten years. In that time, we have ingested and actively manage around 36 million files or around 180 different file formats in our instance of Rosetta [1] as the New Zealand National Digital Heritage Archive Collection.

A key part of our ingest process is file validation. This is a series of automatic assessments that are applied to every file coming into the repository. This provides key information for each file in our care. We get file format, file validation, file characterization and integrity/malware data. File format is provided by DROID/PRONOM [2], file validation / characterization by JHOVE [3]/NLNZMDE [4] (and other tools) and malware checks by CLAMAV [5]. All this is managed by the Rosetta application

Historically we have focused on file format information as a key area for refinement. It is important to us that all files get ingested with a single, accurate and unambiguous file format identification, and much of our research and work has gone into addressing file format identification at the time of ingest [6]. We have also logged, but mainly ignored any file validation issues noted at the time of ingest. Our ingest process has file integrity/validation checks built in through the whole process. Typically, files that fail some aspect of JHOVE validation render successfully, albeit with the occasional warning or complaint from a rendering application.

We decided this year to start to explore those validation errors and to clean up file format ambiguities in the repository. This is part of our ongoing program of work that seeks to understand, identify and mitigate risks at the file level. Our aim is to both understand the impact of those technical risks, and then reduce those risks. This will in turn serve to provide better and (more likely) successful long-term care of our collections.

This paper describes some of that activity and learning and recommendations for future work.

A. Coverage

1) Intended audience – This paper is a useful source for preservation planners, researchers, practitioners, policy makers, and strategists alike. It contains examples of real world use-cases, mitigation implementations, risk mitigations and reflections on our early practice/process.

2) Why are we sharing this paper? The work we've completed to date in this area indicates we have much more to learn about the impact of operational activities on file level objects as part of routine preservation. This learning is both at the organizational level, and as a community of practice. We are sharing this early work to

17th International Conference on Digital Preservation

iPRES 2021, Beijing,China.

DOI: 10.1145/nnnnnnn.nnnnnnn

seek feedback on our workflow and intended direction of travel, and to expose our experience to inform other organizations that have similar intentions. We are sharing both the tools [7] we are making to deliver this program of work and our experience to date.

B. Relation to prior work.

There are not many examples of technical treatments in the literature. There are some clear relationships with normalization workflows like the Danish experience [8], or the Xena tool based workflow [9]. Discussions of these processes tend to focus on the overall process, and the topic of files that trigger technical exceptions are not the primary focus nor covered at the depth required to inform our work.

There are some traces that describe mass migration projects, for example the Harvard migration framework [10] or the North Carolina experience [11]. Whilst these reports are useful as references they also don't reach the level of detail we are hoping to find.

There are many sources that describe migration activities from a theoretical vantage, like our own experience at NLNZ [12] and coverage by the DPC [13]. These don't address the topic of exception management in enough detail to inform our work.

Our project is directly addressing validation issues identified by JHOVE. There are some traces which describe work on this specific facet, for example the OPF have a growing catalogue of error messages [14]. DTH describe a positive outcome from their work on one of the JHOVE error messages [15] and Tunnat describes her work on PDF validation [16].

Whilst all these sources, and the many other similar help to inform and shape our thinking, we share this paper to partially fill a gap that we believe requires attention – how do we mitigate validation errors at scale, and in bulk?

C. Resources

1. Files included

The files used in this project are all identified using searches in our Rosetta instance. A FILE level search was undertaken for all files recorded as being "Not Well-formed" or "Not Valid" as part of their JHOVE assessment completed at ingest. This surfaced about 580k files of 44 different formats. These can be grouped into sets of related formats and by their PRONOM PUID identifiers.

Table 1

Format Family	**Files**	**Formats (No. of PUIDs)**
Audio (AIFF, WAV, BWV)	7,461	12
Document (PDF)	91,113	14
Image (JPG, TIFF, GIF)	15,542	14
Structured Text (HTML, XML)	474,767	2
Text (txt)	165	2

Having identified the files with a reported issue that requires some inspection and possible mitigation, we started to export the files in a PUID set for assessment. The vast majority of these files are XML files with a simple reported namespace issue as shown in Table 1.

The assessment starts with a python script that presents each file to JHOVE and records the output to a text file. All text files are parsed as a set, or "cohort", and the tool summarizes the JHOVE reports, grouping files with similar issues. This allows us to generate cohorts of related files. Treatments are trialed against each cohort based on the reported JHOVE error, and once a suitable mitigation is found, the cohort is passed onto the comparison tool. Most of this cohort making is automated – the file identifier lists are generated by the initial analysis tool, that list is used in Rosetta to export the required files. Fresh JHOVE assessment, cohort analysis and cohort creation are all automated.

2. Roles and effort

A common topic of discussion in the broad digital preservation community of practice is focused on what resources are consumed to achieve outcomes. As this project is still in an early phase we cannot describe the resource commitment required to achieve our goals, we can however describe the effort and resources consumed so far.

The primary effort for this work is undertaken by a Digital preservation analyst who is responsible for the planning and delivery of this work. To date the project has consumed about 75% of their FTE effort for approximately three months. This work doesn't happen in isolation, and is significantly informed by previous projects, especially file format identification cleanup and processing, and ongoing mediation work for items ingested with serious technical problems.

This work is supervised and guided by the digital preservation manager. A weekly hour-long meeting is used to describe progress, inform on blockers/barriers and seek advice for next steps. As this work gathers pace, the manager role is required to sign off on proposed changes. This is a somewhat unpredictable amount of effort. As a principle, we have designed the documentation to be automated where viable, but also covering appropriate detail sufficient to allow manager sign off. As we start to document the issues facing a given format, and then related formats, the sign off effort reduces as we can reuse previously addressed justification and mitigations. New issues take some time to research both at the analyst role but also for the manager to sign off. A rough estimate would be 2 days focused on signoff and reflecting on / improving specific cohort documentation.

There has been some work for the preservation system specialist role, especially in the early steps of information gathering from the Rosetta system. This has been no more than a day since the work was initiated.

The least visible resource / activity is the implementation of any changes to files back into the repository. This is undertaken by the group's change specialist role. We do not yet have a mature set of automatable tools/API etc. to build a fully automated pipeline. Presently changes must be made manually and are the primary resource/effort bottle neck for this project. At present a rough estimate of effort required to address the volume of changes required is 1 day per month. However, we are still working on small cohorts (~30 items max). As we move into the larger sets, this will become too much work to manually achieve. The good news is that with an additional automated step (incorporating the appropriate checks and balances) this effort can be reduced to near zero.

3. Tools Used

The main tool is the comparison framework script developed in the python scripting language. This tool is the main pipeline – it accepts suitably structured inputs of a file requiring mediation, a mediated file, and some core organizational data about that file (IE/FILE/content identifiers). The tool is responsible for undertaking automated comparison of the two files, recording any assessments, signaling any identified differences between the two files and generating the automated section of the cohort documentation. Within the tool we are writing new comparison modules as they are required.

To date we can automatically compare and report our findings for image aspects for most mainstream image formats, metadata collected from EXIFTool [17], JHOVE, ImageMagick [18], Python PILLOW (image library) [19], PyPDF2 [20], PDFminer [21], and TIKA [22]. These tools will be added to as we identify items requiring mediation of validation issues.

We have also extensively used the python DOCX library [23] to automate report writing. This serves two purposes, we can be sure there are no human input errors - all data is drawn from system reports, and used throughout the whole project, the second being efficiency. As cohorts get bigger the amount of data needed per file to maintain an accurate record of any changes becomes problematic. Our automated reporting tool can generate an accurate set of documentation in seconds – it would take a human many hours to do the same.

4. Basic Workflow

Our process, shown in Figure 1, has a core workflow that allows us to add in new comparison process as required.

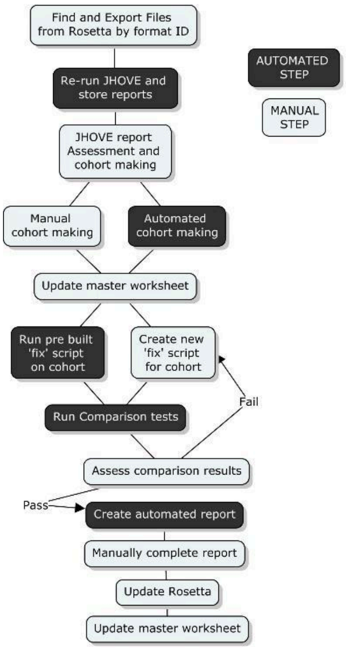
Find and Export Files from Rosetta by format ID
AUTOMATED STEP
MANUAL STEP
Re-run JHOVE and store reports
JHOVE report Assessment and cohort making
Manual cohort making
Automated cohort making
Update master worksheet
Run pre built 'fix' script on cohort
Create new 'fix' script for cohort
Fail
Run Comparison tests
Assess comparison results
Pass
Create automated report
Manually complete report
Update Rosetta
Update master worksheet

Figure 1 The workflow developed to assess and mitigate validation errors

5. Working at the sub file level

Typically, we use file fixity to demonstrate integrity after moving or touching a file. In this work, we are expecting to change a binary object, so can use fixity other than to book end our input and output conditions.

We know we need to demonstrate integrity of our treatment both to ourselves, and our organization. To achieve this, we developed a process that allows use to ringfence parts of the file and validate change or lack of change accordingly. With image files for example this typically means regarding a file as two essences – image, and metadata/technical. We can use techniques like RMSe [24] to confirm the image portion of a file found in an untreated image is the same as the image portion of a file found in a treated file. Equally, we can use tools to extract any and all metadata from both files and compare the two for differences. Any expected and unexpected changes can be reported on, so we know what requires more attention. This is an exciting development that has exciting connotations for all preservation actions.

D. Preservation Issues

This project has surfaced several interesting preservation issues that we believe should be shared and explored further.

1. Where should change be recorded?

The expected outcome of this work is changes to digital files. These changes will result in new fixity values, which requires adequate documentation to maintain an intact 'chain of custody' for collection items. We have rudimentary provenance recording mechanism inside the Rosetta system, however we lack mature tools in this space as a community. This project highlights the need for a standardised mechanism for recording technical provenance [25].

2. What should be kept as a record?

As a product of this work, for each file pair / comparison we generate a set of data assets. This includes JHOVE and EXIFtool outputs for both files, the metadata collected for both files, the files themselves, a summary of detected changes, and assets produced to illuminate detected changes. We are still working out what (if any) of these assets we need to keep once the change has been accepted and the files replaced. We expect that we'll keep more assets in the short term as we mature this process. We are not ingesting these related assets as part of the AIP, nor do we expect to. We are unlikely to expect to manage binary objects outside of Rosetta for any meaningful length of time. We have an open question on what we need to do to balance accurate and fulsome audit, sensible storage, and well-structured and documented AIPs.

Example use case: A TIFF file is found to be using invalid date/time separators in its date/time metadata objects. The fix with the least amount of change to a file is to identify the byte position/offset of the invalid separator token in the file, edit the byte-stream to use a valid separator token, and then save the now edited binary object. Post change validation with JHOVE indicates when we have successfully completed this operation. The new file replaces the old file in Rosetta. What do we record as part of the file/AIP that documents what we did, why we did it, and the byte locations we changed?

3. What do we expect of internal metadata?

A key mitigation tool for image files it to use EXIFtool to 'remake' files that have some structural deficiency. We have a scripted process that makes a binary copy of a given file, exports its metadata with EXIFtool, and uses EXIFtool to reassert the metadata. This process is extremely useful because we know it doesn't touch the content (e.g. image) part of the file. In our current usage EXIFtool is updating some parts of the files metadata where it has been designed to do so but we weren't expecting, for example the file's XMP data objects. We are unsure what we should do about this changed metadata.

Example use case: A TIFF file reports out of sequence tiff tags via JHOVE. TIFF files are expected to present all tiff tags in a proper numerical order. Some TIFF creation software does not always achieve this, and as a result, tags are written out of sequence. As a result of reasserting the metadata via EXIFtool these sequence errors are resolved. EXIFtool reorders the metadata before it saves it to the file. A by-product of this activity is that EXIFtool updates a few key fields, such as last modified date. Whilst not incorrect – these changes are being made to the metadata and we need to intentionally decide whether to accept these new data.

4. What about masked errors?

An unexpected outcome of this work has been the exposure of other potential errors in a file. We have found multiple instances of previously unknown errors in files becoming apparent once we have started a treatment / manipulation on a file for a known error. This raises some questions about our current risk / issue measuring tools. We cannot be sure that the issues we have recorded for a given file are the sum of all its issues. Can we be sure that a "good" file wouldn't display the same behaviour when subjected to this metadata re-assertion process? This raises some obvious questions about how to plan and prioritize effort – do we know where our main areas of concern are?

Example use-case: A TIFF file presents with a JHOVE warning of values not presented at word boundaries. This is a similar problem to the sequence error. We can use EXIFtool in the same way to correct this issue, however, in some cases our post change summary shows that several other metadata fields also change. Analysis of these fields indicate that more of the file metadata was incorrect than originally detected. We are yet to resolve how we can ensure we are working on the most accurate version of all metadata exposed from a file.

5. What is the factual record anyway?

Following on from the above example, we have open questions around which set of metadata is the correct metadata. There is a good justification for both the original, and the new metadata to be the primary data found in the preservation master file. The original is the file metadata when it was ingested, the new is the file metadata as it is freshly assessed. We are unsure what the impact might be for deciding on this question. One side results in an item that has had original (but inaccurate) data removed, the other is a file that may be unsuitable or unusable for forensic (e.g. photogrammetry) or technical research. We do know this change is currently 'lossy', Once we have reasserted the metadata with EXIFtool we are not able to easily revert the changes to the file. Ideally, we would like this to be a lossless change – something we could undo if we had the correct metadata recorded in a structured technical provenance note.

6. How do we record business rules?

Having completed an assessment of a file, we may decide that we disagree with the JHOVE report and do not find an error in a file. It may be that having investigated the file we are comfortable that the error being reported is not consequential to the successful long-term preservation of the file. Presently we do not have a mechanism that allows us to indicate that we have assessed a file, and actively chosen to ignore the error. We also cannot easily document our finding to future technical consumers of the file (including our future selves) so they can benefit from our research.

We also need to be able to exclude files that we described above from risk reporting / technical assessment. If we are not able to mark a file as “safe”, we will have to spend a great deal of effort tracking individual files outside of Rosetta, so they are removed from our automated risk reporting and cohort analysis.

E. Outcomes for NLNZ

This is a relatively new piece of work. At the time of writing we are approximately three months into it and have processed around 3,000 individual files. We recognise that pace initially is slow, especially where we are processing new file formats, or researching newly identified problems. Our focus on automated tools is already paying dividend, and we are much further along than we might be without that extra processing support.

We have reached a good position on image files - we have a few different processes we can use, high quality assessment / comparison routines and a growing understanding of the types of issues we find in this type of format. We are exploring how we expect to address PDF files and are leveraging tools and techniques previously established - such as converting a PDF in to pages, saving those pages as individual images, and running the pairs of images (original page an image, new page as an image) through the image comparison process to check for differences.

Our key new conceptual tool is to explore a file’s integrity beyond a traditional fixity. We are splitting the item into ‘fingerprintable’ blocks and operating with precision on those blocks alone. These blocks might be a files’ core essence (the image, the audio), the files metadata (its EXIF or XMP components, or its structure (its tag sequence, or logical positions within files). Each block has its own validation process, and we can return high confidence indicators of parts of a file that have been changed and more importantly, those that have not.

We expect this work to achieve for our collections:

1. Reduce the “risk” found in the collections. By making individual files standards adherent we reduce the number of technically ‘risky’ files that we hold.

2. Identify and mitigate broken files. By developing methods of checking files at a deeper level, we can detect issues that have gone previously undetected. To date, we have discovered around ten files out of the 3,000 we have processed that are damaged beyond our ability to fix them. These files were ingested in that state.

3. Increase and enhance our understanding of individual formats. This work demonstrates the level of technical understanding required to be able to process items that report errors. In many cases once we have identified the source of the error and developed a solution that fixes the error to our satisfaction we are able to mechanize that solution, and repeat it as needed. It is our expectation and obligation that we share these techniques with our community.

REFERENCES

[1] https://exlibrisgroup.com/products/rosetta-digital-asset-management-and-preservation/

[2] https://www.nationalarchives.gov.uk/PRONOM/Default.aspx

[3] https://jhove.openpreservation.org/

[4] https://jhove.openpreservation.org/

[5] https://www.clamav.net/

[6] McKinney, et al. 2014, Reimagining the Format Model: Introducing the Work of the NSLA Digital Preservation Technical Registry (pdf, 2MB), New Review of Information Networking, Vol 19 (2). p 96-123

[7] https://github.com/NLNZDigitalPreservation/file_validation_checking_tools

[8] Strategy for archiving digital records at the Danish National Archives, 2014

[9] Taming digital records with the Warrior Princess: developing a Xena preservation interface for TRIM, O’Donnell, 2010

[10] Digital Preservation Roadmap for Harvard (2014-2020)

[11] The “M” Word: What Works and What Doesn’t in File Format Migration

[12] Preservation Actions; where we started, and where do we go from here?

[13] https://www.dpconline.org/handbook/organisational-activities/preservation-action

[14] http://wiki.opf-labs.org/display/Documents/JHOVE+issues+and+error+messages

[15] https://heritage-digitaltransitions.com/adobe-fixes-bug-uncovered-by-jhove/

[16] https://openpreservation.org/blogs/jhove-the-one-and-only-pdf-validator/

[17] https://exiftool.org/

[18] https://imagemagick.org/index.php

[19] https://pillow.readthedocs.io/en/stable

[20] https://pypi.org/project/PyPDF2/

[21] https://pypi.org/project/pdfminer/

[22] ttps://tika.apache.org/

[23] https://python-docx.readthedocs.io/en/latest/

[24] https://www.sciencedirect.com/topics/engineering/root-mean-square-error

[25] https://natlib.govt.nz/files/digital-preservation/Proposal-for-new-PREMIS-entity-provenanceNote.pd, Gattuso and McKinney 2012