Beyond the Binary: Pre-Ingest Preservation of Metadata

Jessica Moran
National Library of New Zealand
P O Box 12349
Wellington 6001
+64 4 460 2862
jessica.moran@dia.govt.nz

Jay Gattuso
National Library of New Zealand
P O Box 12349
Wellington 6001
+64 4 474 3064
jay.gattuso@dia.govt.nz

ABSTRACT
This paper describes some of the challenges the National Library of New Zealand has faced in our efforts to maintain the authenticity of born digital collection items from first transfer to the Library through ingest into our digital preservation system. We assume that assuring the authenticity and integrity of digital objects means preserving the binary objects plus metadata about the objects. We discuss the efforts and challenges of the Library to preserve contextual metadata around the binary object, in particular filenames and file dates. We discuss these efforts from the two perspectives of the digital archivist and the digital preservation analyst, and how these two perspectives inform our current thinking.

General Terms
Infrastructure opportunities and challenges; Frameworks for digital preservation; Preservation strategies and workflows; Innovative practice.

Keywords
Digital archivists, born digital preservation, technical appraisal, ingest.

1. INTRODUCTION
The National Library of New Zealand (the Library) actively collects born digital heritage collections, including unpublished material such as manuscripts, personal papers, organizational archives, photographic archives, and oral histories. In 2008 the Library went live with the National Digital Heritage Archive (NDHA). The NDHA encompasses several ingest and access delivery components and utilizes the Ex Libris digital preservation software, Rosetta. The workflow for bringing these collections into the NDHA begins with an initial technical analysis by the digital archivists, then curatorial appraisal. Once appraised, collections are arranged and described, prepared for ingest, and ingested into the NDHA. This is a collaborative workflow with input from various stakeholder groups within the Library including digital archivists, digital preservation analysts, curators, and arrangement and description librarians. In the past few years our born digital collections have grown in both size and complexity. This growth has put stress on our workflows and challenged us to develop more rigorous pre-ingest and ingest processes. For example in the last year we have received four 1TB transfers, as well as two collections with 100,000 plus files. This is in addition to a number of hybrid collections which have included 50 or more physical carrier media items each.

At the Library digital archivists are responsible for the transfer of born digital manuscript and archives collections into the Library, as well as initial technical appraisal and preparation of collection items for ingest into the digital preservation system. The digital preservation analyst is responsible for technical assessment of digital content going into the digital preservation system, and troubleshooting digital content that fails validation checks. While both roles are informed by an understanding of both archival and technical considerations, the digital archivists serve as archival and content subject matter experts, while the digital preservation analyst is subject matter expert for technical concerns. The digital archivists and the digital preservation analyst work closely together developing ingest workflows. Bringing the two perspectives together allows us to design and develop robust workflows that better meet the needs of preserving the binary object, as well as contextual metadata around the objects such as filenames and dates. Working together gives us a better understanding of each other's perspective, a more holistic view of the digital preservation challenges, and ultimately allows us to have greater confidence in our ability to assert the provenance, authenticity, and trustworthiness of the digital content we are preserving.

This paper will outline, from the perspective of the digital archivists and digital preservation analyst, some of our challenges, particularly in the areas of filename and file dates. We will look at these two seemingly simple pieces of metadata, filename and file dates, and how we track and store that metadata from initial transfer to the Library until ingest into the NDHA from the perspective of the digital archivists and the digital preservation analyst. In the library and archival digital preservation environment we are familiar with the main types of metadata necessary for digital preservation such as descriptive, administrative (technical, rights, preservation) and structural. However we are also interested in file system metadata such as filenames and dates that are not embedded with the objects themselves, but rather are stored externally in the file system and are generated to track things like name, size, location, usage, etc.[1] Filename and date metadata would seem relatively basic metadata to capture and preserve, but in our experience, these two pieces of metadata have challenged us to think critically about what constitutes acceptable, reversible, and recordable change and where and how this metadata should be stored for preservation and later for delivery to users.

2. POLICY AND PLATFORMS
2.1 The Policy Context
The Library, in collaboration with Archives New Zealand (ANZ), has established a number of policies to support digital ingest and preservation activities. Of these, the Preconditioning Policy has had

the greatest impact on the pre-ingest workflows. The policy states: "the diverse nature of digital content means that there are times when it is desirable to make changes to it before it is ingested into the preservation system." These changes are classed under the term "preconditioning" and the policy describes the limits of change that can be introduced to digital content from the time it is under control of the Archives or Library to its being brought into the preservation system. These changes can include an alteration to (or addition of) file extensions and/or removal of unsupported characters in the file name. The undertaking of these preconditioning actions enables the Library to ingest more stable files into the preservation system's storage database. Regardless of the change being made during this stage, the policy demands that one be able to demonstrate that actions will not affect the intellectual content of the file, all changes are reversible, and all changes must be documented with a system-based provenance note describing any changes made. [2]

2.2 The Business and Technological Context

The Rosetta digital preservation system and some of the business and operational rules around the Library's use of the system influence our workflow decisions and requires some explanation. All files ingested into the preservation system go through the Validation Stack in Rosetta. The Validation Stack includes file-format identification (by DROID), file validation (by JHOVE), metadata extraction (NLNZ metadata extract tool and JHOVE, checksum generation (CRC32, SHA-1, MD5), and virus check (CLAM AV and F-PROT). This series of checks is run against all files in a Submission Information Package (SIP). During this validation process files within a SIP with missing or incorrect file extensions, some non-ASCII encoding of filename strings, and invalid dates will cause the SIP to fail validation and the entire SIP will be re-routed to the technical workbench area for assessment. Within this area of Rosetta there are limits on the actions that can be performed and the tools that can be used. The Library has found that when working with large unpublished collections that may require a series of fixes, it is more efficient to perform these actions outside the system prior to ingest. We also hope that ingesting files in this "stabilized" state will make them easier to report on and identify later. Further, we expect these preconditioning activities to make future operations and actions on the files, such as file format transformations for preservation, easier and less labor intensive. We are also conscious that any pre-ingest preconditioning activities we perform must be recorded systematically inside Rosetta.

For access and delivery of collection items, Rosetta acts as the link between a descriptive record maintained in a collection management catalogue system and the binary stream (digital object), or streams, that are being described. The Library's collection policy, collection plans, and business rules inform the arrangement and description and access to unpublished digital content. While a discussion of these policies is outside the scope of this paper, it is worth noting that these plans affect our practice; essentially our preservation system and collection management system work together to deliver access to digital objects. This has led to the requirement to collect and deliver discrete digital objects that are linked with a descriptive record. To that end, our usual process is to cleave any digital objects from their host file system, because we are not typically interested in preserving the file system as a collection item. To maintain the collections as any number of file system sized items would break the cataloguing and delivery mechanisms we have developed and expect to continue using to deliver access to content in the future.

While we may create disk images as part of our pre-ingest technical appraisal and processing workflow, in most instances it will not be the disk images but the individual files we preserve in Rosetta. Therefore we have developed a workflow where we are able to individually address digital objects, marking items in or out of collection scope, marking individual items as in need of special attention, and ultimately addressing the whole collection as a collection of untethered digital objects, rather than as aggregate set of items whose individual needs cannot be easily addressed.

Accepting then that we have a requirement to transfer content from the host transport or storage media, we want to ensure that the record of the items held by the media as metadata is not lost during this separation process, and that we have captured that metadata in a format that can be easily used during appraisal, arrangement and description, and ingest. We also want this metadata included as part of SIPs and therefore retained within the preservation system. While there are any number of tools that address parts of this process it has been a challenge to create a workflow that could systematically and safely maintain this metadata in a way that is useful to our various stakeholders.

3. PRESEVING FILENAME METADATA

3.1 Filename information from a digital archivist's perspective

The filenames of born digital objects, especially in manuscript and archives collections, provide us with information not about what the object is, but also what the creator might have been thinking during creation, and how the object relates contextually to other objects within a collection. For these reasons we want to be able to retain original filenames and deliver them back to researchers, even as we understand that within our preservation system Rosetta will assign new identifiers to files.

Born digital objects in manuscript and archives collections routinely come with notoriously non-standard naming conventions. Some of the issues that we encounter regularly include: older filenames with full stops in the filename, filenames with missing or incorrect file extensions, filenames with special and illegal or restricted characters in the filename, and filenames (and paths) that exceed the current Windows character limits. Other common naming issues we find include character encoding, soft hyphens, diacritics (especially in Māori language filenames), and other non-English language characters that neither our pre-ingest systems nor Rosetta can currently recognize correctly at ingest. If at all possible we will not touch filenames, however in these cases we will need to make a change to the filename in order to ingest the file into our digital preservation system.

In order to address these filename issues the Library adopted a preconditioning policy that sets the limits of acceptable change that can be introduced to digital content from the time it is brought into the control of the Library to the time it is ingested. For the digital archivists, who are responsible for the initial transfer of digital content into the Library, this gave us the framework for making acceptable changes to filenames where necessary. However, we have struggled with when, how, and where to document our changes and how to ensure that documentation makes its way into the digital preservation system. For example we create an original inventory of all files in a collection at our first contact with the material and then verify that that information is unchanged upon transfer to our pre-ingest storage location. These give us a snapshot of filenames and file path locations and establishes baseline fixity. We then identify any filename or extensions requiring preconditioning. [3]

Once we have identified filenames that need preconditioning prior to ingest, we must make those changes. In order to maintain the reliability and trustworthiness of our custodianship, we want to automate as much as possible not only the preconditioning actions we take, but also the recording of those changes, and their ingest into Rosetta as provenance metadata. For individual files or even small collections this is a relatively simple process. But once we began applying preconditioning to larger and more complex manuscript and archival collections, where for example we were processing thousands of files at a time, we discovered that we needed a better way to automate this process. [4]

3.2 Filename information from a preservation analyst's perspective

The preservation analyst provides technical assessment of digital content as it enters the digital preservation system and works especially with manuscript and archives collections when digital objects fail or are likely to fail technical validation checks. For the preservation analyst filenames pose an interesting and complex technical problem.

A filename can be reduced to a relatively arbitrary string that is used as a handle for an associated binary stream. They become a problem where an operating system (OS) or applications have reserved function for any of the characters or code points in the string that is being used. Different operating systems have different constraints for reserved characters, and over time operating systems have changed their rules.

For example in a Windows environment the characters: \ / : * ? " < > |. are all reserved and have special associated functions meaning that you cannot name files with one or more of these characters. [5]

Linux systems have a different set of constraints restricting the use of forward slash and *NULL*, Mac OS prohibits the colon. [6] It's not just the OS that requires these restrictions; it is also the underlying file system that has some reservations in character usage.

These differences start to become an issue when the Library receives files from one OS and expects them to conform to the constraints of another. The main tools we have used while working with files prior to ingest are Windows based, and the preservation application has a couple of different file systems that inform the OS based filename character constraints. This means that what may be permitted in one environment, may not be permitted in another.

OS constraints are one part of the character issue we face, but we also need to deal with "non-standard" characters, like macronised vowels (e.g. ā, ē, ī, ō and ū) found in written Māori, one of the 3 official languages of New Zealand. [7] We cannot expect all our systems, users and processes to support the use of extended UTF-8 (or other such encodings) in filenames, and in fact, we often do not know what the original encoding was that was used to label the filename. We often encounter filenames recorded as "my□file.pdf" where the "□" glyph is essentially the OS recording software unable to represent the original character. We have also observed that different tools have different ways of addressing this problem of the presence of a code point it cannot decode. Some use question marks or a symbol like the above, others simply skip the offending character. But we must stabilize these filenames to ensure we are using correct, consistent, and valid filenames that can move safely between platforms.

File extensions are another component of the file name that is of interest to us. A file extension is often (incorrectly) used as a proxy for a meaningful file format identifier. Sometimes the format type and the file extension match, other times they do not. Operating systems have different methods of associating applications to various file types, one of which, particularly in the Windows domain, is the file extension. When the file extension is incorrect to a "normal" user it generally means that the file might not open with a suitable application, if at all. They may need to take some action to ensure that the file is properly associated with a suitable piece of render software, like changing the file extension manually.

Proper handling rules means that for digital preservation we need to treat files slightly more sensitively. We might want to know what the original file extension was as it is an important part of a file's provenance. Knowing that the creator labelled a binary stream "my_file.pdf" when it ought to be "my_file.tif" for example might be useful to a future researcher, so that when a the researcher is delivered the file as "my_file.tif" the information that it was originally named "my_file.pdf" is referenced elsewhere in the metadata for their information.

We also often encounter files missing extension altogether. In modern Windows environments this would not be common, but many manuscript and archives collections we work with were not created in a modern Windows environment. Further, some file systems and operating systems don't require file extensions and files often don't need an extension.

We made a policy choice to add or fix a file extension wherever possible, to prevent a file being delivered to a user without a valid file extension and to ensure that Rosetta has the right context clues so that the correct internal delivery mechanism is used.

This poses some challenging questions for us. What do we do when representing the original object to a user? Do we include the new file extension, or present the original string without the extension? How do we record any changes we've made and deliver that information to the user?

There is a further consideration. Rosetta has its own method of dealing with filename issues – in fact, at the first point of contact with a file Rosetta strips off the original filename, assigns the file a new filename, and retains the original filename in the Archival Information Package (AIP) metadata.

This acts as a useful normalizing process, because the files can subsequently be addressed by their new "clean" filename, but as soon as we want to deal with files in their original form, (to for example deliver to a user) we need to re-associate the binary stream with its original and potentially troublesome filename. The current method for recording these preconditioning provenance actions is to record a PREMIS provenance note that describes the state of the item before and after the intervention.

The provenance note is attached to the CREATION event, and is constructed using an in-house convention that was designed to support different types of provenance worthy interventions, including the changing of file dates, addition or change to file extensions, and the cleaning of special characters from file names. An example is offered below:-

Figure 1 - Example Provenance Note

```
<section id="event">
 <record>
  <key id="eventIdentifierType">Indigo</key>
  <key id="eventIdentifierValue">Indigo_1</key>
  <key id="eventType">CREATION</key>
    <key id="eventDescription">Provenance Note from Indigo</key>
```

<key id="eventDateTime">Wed Mar 18 12:47:24 NZDT 2015</key>

<key id="eventOutcome1">SUCCESS</key>

<key id="eventOutcomeDetail1">002_File extension was added: The file was submitted by the donor without a file extension. On the recommendation of the Preservation Analyst, a .wpd extension was added to this filename by the Digital Archivist as it has been identified as WordPerfect for Windows, version 5.1. </key>

</record>

This provenance metadata is expected to be used in the future to ensure we are able to provide both researchers and digital preservation staff with an accurate view of an object's metadata, including any preconditioning actions we may have performed on that metadata.

4. PRESERVING DATE METADATA

4.1 Dates from an archivist's perspective

Most born digital objects that come into the Library will have three dates: created date, last modified date, and last accessed date. However depending on what file system the objects were created on all of these dates may not be present. Ideally we would like to collect and preserve all three dates. It is important to note we would like to preserve these dates not so much because we believe that these timestamps gives us irrefutable information about when exactly the digital objects were created, modified, or accessed, but rather that they provide contextual information about how and when the objects were in use. They are, in essence, another piece of the puzzle used to confirm an object is what it says it is.

As the born digital collections being transferred to the Library increase in size and complexity, the amount of time from initial transfer to ingest into the digital preservation systems has also grown. This has the effect of increasing the time collections stay on pre-ingest storage, and may also increase the number of times the files are touched by digital archivists, curators, arrangement and description librarians, and digital preservation analysts. We have workflows and processes in place to mitigate risks associated with this, but we do not always have control over the underlying file system of our storage, and any changes that our information technology support might make. It is therefore important for us to understand and document both what and how the original timestamp dates were created, and how any subsequent movement of the files can effect these dates.

One of our first steps during a transfer is to view the transfer media using a write blocker and use tools to capture an inventory of the digital object's basic file system metadata including file name, file location, size, and dates available from the storage media on which the collection was transferred to us. We then must transfer the files to a pre-ingest location without affecting these dates. We found we were having to cobble together a number of tools, and therefore steps during the transfer process. We also understood that every time we wrote a file to a new file system these timestamps were subject to change, and indeed, every time we interact with a file, we run the risk of altering the timestamps.

We needed to adopt processes to ensure we did not inadvertently affect the file metadata and can ensure the authenticity and reliability of the digital objects under our control, while at the same time working in an environment that allows for multiple people to work with objects, and prepare the files for ingest into the digital preservation system.

One solution is the creation of forensic disk images as a first step in the transfer process, to essentially wrap all the digital objects in an image file that records file system and related metadata for later use. [8] At the Library we have adopted many lessons in workflow and digital handling from the world of digital forensics. [9] However, as discussed above our use case will in most instances call for extracting and loading the individual files to the digital preservation system. In these cases, even if we do have a disk image and all the relevant file system and individual file metadata, we again find ourselves in the same dilemma, that is: how do we preserve the original timestamps, even if the files themselves have been moved? And if the timestamps associated with the original digital objects are not the timestamps associated with the object at the point of transfer to the Library, how do we ensure that the original information travels with the object? While this information can be captured in a DFXML file, once we extract individual files from the disk image, that metadata becomes disassociated from the individual files and requires more steps to remarry inside the digital preservation system.

The problem of dates become further complicated when working with older born digital collections that come into the Library in less than ideal condition. In a recent example we processed a manuscript collection that came into the Library on 3.5" high density floppy disks. On our first inventory of these files we noted that the dates associated with some files appeared to be broken. That is, on about a third of the files, rather than the expected 1990-2000 era created and modified dates, we instead saw dates that ranged from 2032 to 2066. These errant dates presented a number of questions for us. First, did we want to preserve these dates as they appeared? Certainly this was how they arrived in the Library and preserving the dates, even if they were demonstratively wrong, seemed a legitimate strategy. However the initial attempts to ingest the files with these dates failed as our digital preservation system rejected the files with dates from the future as invalid.

We then investigated these files in more detail. Returning to the disk images, we noted that the erroneous dates were present on the disk images. We investigated whether it was an issue with the original file system and how the files were written to disk and then being read by our OS. However, the erroneous dates were present regardless of what OS we used.

We next asked ourselves, if we could not discover the original or "true" timestamps, how we might change those dates in a trustworthy manner, maintaining authenticity. In order to not violate our preconditioning policy, we needed to ensure any change we made did not change the intellectual content of the object, was reversible, and that we provided sufficient documentation of what changes we made to the objects and why. Finally, if we were going to have to change the timestamps, what would be the most appropriate date to change the files to? Eventually we concluded that if we were going to make changes to timestamps we should make them as transparent as possible to future users. Thus we chose to change all erroneous dates to a current 2015 date, in the hope that such dates would alert future users to the discrepancy in the purported dates in both the content and descriptive records and the file date stamps, and hopefully lead them to further investigate the provenance notes in the preservation metadata.

4.2 Dates from a digital preservation analyst's perspective

The collection described above was a particularly troublesome set of files, and worth describing in more detail.

Before we look at the collection, it's worth restating some of the principles we have adopted when handling files.

- We expect a file to arrive at the Library with accurate dates. This includes created, last accessed and last modified dates. Some files / file systems include other dates, such as last printed.
- We understand that not all OSs support all those dates, and not all files have this metadata available. For example earlier Linux versions specifically did not support the notion of a created date as it is not required by POSIX the standard (the basis of an underlying file system) [10]
- We expect to be able to handle files without changing those dates – we consider them to be a valuable part of the item.
- We understand there is a particular issue around the "created" date of a file in the archival context. (Should this date reflect the date a clone of a bitstream was made, or the original bitstream?)
- We understand that different types of date / time have different resolution, ranging from milliseconds to whole days. [11]
- We understand that moving a file changes the OS metadata differently to copying a file.
- We have a practice of only working on copies of original files wherever possible, especially when testing procedures.
- We have a practice of touching the original file as little as possible and only as much as needed to get the file into the preservation environment
- We find that at present there is no perfect tool or process that we can use that ensure we can operate with absolute comfort.

These rules are not all set in stone, and we have been working on our business practices and tools to help us align our practice with our policy as much as possible. Each time we encounter a problematic collection, our model shifts slightly to accommodate new learnings or specific nuances of a file or set of files.

In the example discussed above, we knew we had a problem ingesting these files when one of our automated tools returned an error.

In this case we were using the Python ZIP library [12] to coerce the collection into a single zip file, ready for automated ingest into Rosetta (via a process we call the csv ingest method).[1] The scripts we had written returned an error because it could not handle file objects created after the current date.

This is not really surprising as conceptually the files should not exist if the file creation date is assumed to be true, however there are no technical controls over the setting of such date, and various methods can be used to change any dates associated with a binary stream in a file store. [13]

In our example set, we found that a third of the dates we could find were incorrect, and about half the number of dates we thought we ought to have (1 x created date, 1 x last modified and 1 x last accessed date per file, for 1,101 files) was missing from the source set.

We noted that the missing and incorrect dates were scattered amongst the collection, which was originally housed on seven 3.5" high density floppy FAT12 formatted disks.

The incorrect dates fell into two groups, a set that ranged from 2032 to 2035 and a set that ranged from 2062 to 2066. The original set ranged from 1999 to 2003.

It was notable that the original dates spanned a four-year period, as did the wrong dates, just with a relatively consistent offset.

Our working assumption for some time was that if we could find the offset, we could assert the original date with some confidence. Being copies of emails and travel schedules many of the affected files actually had dates as part of the intellectual content. This meant we could tell if we were in roughly the right time period.

That approach held for a while, at which point a discovery that the time portion of the date was equally affected.

This left us in some doubt that the original date could be recovered, and in conclusion we returned to the idea that we would set the dates to something obviously (to us) wrong and ingest the files with provenance information that documents the decision and justification for our actions.

This problem is not dissimilar to those we face with filenames, and it follows that we might want to use a similar approach. If we capture the filename and dates that are available to us at the first point of contact with a file, and store them in a suitably convenient way, we can act with more confidence in changing this type of metadata.

5. DEVELOPING BETTER TOOLS

5.1 One possible solution

One general characteristic of digital curation work is finding oneself in a situation where there is both a preponderance of tools and no one tool or even suite of tools, that meets all our needs. In the Library's case what we needed was a tool to help us automate the original and any subsequent transfers of born digital content, ensure the capture of original filename and date metadata and any preconditioning actions we performed, and at the same time create a log of that activity that is auditable and both human and machine readable.

5.2 Our requirements

Working together, the digital archivists and digital preservation analyst developed the Library's requirements. These requirements included needing a way to create an inventory of each file on a piece of media. This inventory should include the original filename, file path location, an MD5 hash of the binary object, the file date created, date last modified, date last accessed, and the file size. But we did not just want that information, we wanted the ability to take all that information, move all the files to a new storage location, recheck everything and confirm that all the data points were the same and that no changes were made in the course of the transfer, or if changes were made, to record those changes. Finally, we

[1] CSV ingest is a method of ingest in which we create a ZIP of all the files to be ingested, as well as a csv manifest of all the SIP metadata to support the ingest directly to Rosetta. Rosetta then builds the SIP and ingests the files automatically.

needed a way to compare these inventory logs if a set was moved more than once, as is required from time-to-time.

Based on previous experiences with other file copying tools that either failed or silently made changes to filenames with reserved, and non-standard characters, and our prior transfer experience, we knew we would continue to regularly receive files with these issues. Therefore we also needed a way to automatically remove any restricted characters and record those changes at the time of the initial transfer.

Essentially what we wanted was a tool that can meaningfully capture this metadata, keep that metadata linked to the original objects, and be integrated into our existing workflows and systems. By bringing our archival and technical perspectives together, we have begun working on the early stages of a tool that can do this for us in a way that meets both our needs and the needs of our systems. [14]

The underpinning requirements that we are attempting to address cover the following:

- Must be easy for relatively non-technical users to deploy
- Should result in the transfer of all files found in a mounted file store, starting at a nominated folder (recused through the extents of all child folders to a nominated location elsewhere
- File system structure should be mirrored
- File fixity is recorded and maintained in the copied item
- File dates are recorded and maintained in the copied item
- Filename and paths are allowed to be sanitized as long as done in accordance with preconditioning policy.
- Boolean comparisons of source and destination are recorded for all data elements (file names, paths, fixity and dates)
- Elements are recorded in a way that allows accurate metadata and provenance information to be captured with binary objects and used in creation of ingest SIPs.

Most of the above is present in the script we developed, and we have used an early version of this script to confidently move several terabytes of data, and further to assist in the automatic ingest of complex collections into Rosetta.

We have encountered a number of interesting occasions where it was not possible to reassert some of the original metadata on the copied file. The script uses a metadata preserving method of copying a binary from A to B. It collects accessed date, modified date, and created date as applicable and records these in a log for that item. Further, it checks that the dates found on the copied item are the same as the original item. If they are different, it attempts to assert the original dates onto the copied file. It finally checks the two items again, and if they are still different this discrepancy is recorded as another data element. In the case of dates failing to be maintained for the copied item, the original dates are captured before we try and copy the item, and the Boolean flag is set to clearly indicate we need to create a provenance event for changes in dates to ensure an integrious record is maintained.

The steps enumerated above are an improvement on our existing practice, and result in data that can be used to confidently move content from location A to location B, capturing accurate metadata from any operating system / file system that it touches along the way, and specifically helping to drive an increasingly automated pre-ingest workflow where applicable.

Our ongoing questions concern the extent to which delivery of objects from the digital preservation system should include not just a stated assurance that this is an authentic binary object, but also proof of that integrity and authenticity through delivery of the associated metadata. Attempting to answer these and other questions that are sure to arise will require us to continuing working together and learning from each other's perspectives.

6. ACKNOWLEDGMENTS

Our thanks to Leigh Rosin (Digital Archivist, National Library of New Zealand) who was part of the many conversations that informed this paper.

7. REFERENCES

[1] Rogers, C. 2015. Diplomatics of born digital documents – considering documentary form in a digital environment. *Records Management Journal* 25, 1 (2015), 6-20. DOI= http://dx.doi.org/10.1108/RMJ-03-2014-0016

[2] Joint Operations Group, Policy (JOGP), Department of Internal Affairs, 2012, "Digital content preconditioning policy," pp.1-2. http://ndha-wiki.natlib.govt.nz/assets/NDHA/About-Us/Strategic-Partnerships/Digital-Preservation-Policy-Manual.pdf .

[3] Rosin, L. 2014. Applying theoretical archival principles to actual born-digital collections. *Archive Journal 4* (Spring 2014). http://www.archivejournal.net/issue/4/notes-queries/applying-theoretical-archival-principles-and-policies-to-actual-born-digital-collections/ [accessed 1 July 2015].

[4] Rosin, L. and Smith, K. 2014. Then and now: the evolution of digital preservation and collecting requirements over a Decade. In *Proceedings of the 11th International Conference on Digital Preservation* (Melbourne: State Library of Victoria, 6-10 October, 2014) https://www.nla.gov.au/sites/default/files/ipres2014-proceedings-version_1.pdf .

[5] Naming Files, Paths, and Namespaces, https://msdn.microsoft.com/en-nz/library/windows/desktop/aa365247%28v=vs.85%29.aspx, [accessed 17 April 2015]

[6] File Naming Conventions in Linux, http://www.linfo.org/file_name.html [accessed 17 April 2015].

[7] Korero Maori, Tohuto – Macrons, http://www.korero.maori.nz/resources/macrons.html [accessed 17 April 2015].

[8] Lee, C.A., Woods, W., Kirschenbaum, M., and Chassanoff, A. 2013. *From Bitstreams to Heritage: Putting Digital Forensics into Practice in Collecting Institutions*. White Paper. September 30, 2013.

[9] Kirschenbaum, M.G, Ovenden, R. and Redwin, G. 2010. *Digital Forensics and Born-Digital Content in Cultural Heritage Collections.* Council on Library and Information Resources, Washington, D.C. http://www.clir.org/pubs/reports/reports/pub149/pub149.pdf

[10] Unix &Linux, How to find creation date of file? http://unix.stackexchange.com/questions/91197/how-to-find-creation-date-of-file [accessed 17 April 2015].

[11] Corbet, J. (2007) "Once upon a time." http://lwn.net/Articles/244829/ [accessed 17 April 2015].

[12] Python Standard Library, 12.4 Work with ZIP archives. https://docs.python.org/2/library/zipfile.html [accessed 17 April 2015].

[13] Python – change file creation date, http://stackoverflow.com/questions/887557/python-change-file-creation-date [accessed 17 April 2015] and How do I change the file creation date of a Windows file from Python, http://stackoverflow.com/questions/4996405/how-do-i-change-the-file-creation-date-of-a-windows-file-from-python [accessed 17 April 2015].

[14] Safe_mover.py, https://github.com/jayGattusoNLNZ/Safe_mover [accessed 17 April 2015].