THE APPLICATION OF FILE IDENTIFICATION, VALIDATION, AND CHARACTERIZATION TOOLS IN DIGITAL CURATION

BY

KEVIN MICHAEL FORD

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Library and Information Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Advisers:

Research Assistant Professor Melissa Cragin
Assistant Professor Jerome McDonough

ABSTRACT

File format identification, characterization, and validation are considered essential processes for digital preservation and, by extension, long-term data curation. These actions are performed on data objects by humans or computers, in an attempt to identify the type of a given file, derive characterizing information that is specific to the file, and validate that the given file conforms to its type specification. The present research reviews the literature surrounding these digital preservation activities, including their theoretical basis and the publications that accompanied the formal release of tools and services designed in response to their theoretical foundation. It also reports the results from extensive tests designed to evaluate the coverage of some of the software tools developed to perform file format identification, characterization, and validation actions. Tests of these tools demonstrate that more work is needed – particularly in terms of scalable solutions – to address the expanse of digital data to be preserved and curated. The breadth of file types these tools are anticipated to handle is so great as to call into question whether a scalable solution is feasible, and, more broadly, whether such efforts will offer a meaningful return on investment. Also, these tools, which serve to provide a type of baseline reading of a file in a repository, can be easily tricked. It is possible to generate files with nothing more than a proper file extension and correct magic number and have the tools "positively" identify the file. This is not the same as a file that conforms to its specification, and one that could be considered valid. The ability to manipulate the results returned by these tools raises issues of identity, trust, security and risk.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

File format identification, characterization, and validation are actions performed on data objects by humans and computers, in an attempt to identify the type of a given file, derive characterizing information that is specific to the file, and validate that the given file conforms to its type specification.[1] When formed as questions, these actions seek to answer:[2]

- Identification: Is it possible to determine what type of file this is?
- Characterization: What are common characteristics for this file type?
- Validation: Does this file conform with the characteristics of its identified file type?

These activities are extremely common and occur so often as to go largely unnoticed. Programs often create files on disk with specific file extensions that help to identify the file not only to the human that will access that file but also a computer program what will read it and a network that will transfer the data. File systems (and programs) might also record creation and modification dates that help humans and computers identify the correct version of a data object.

Characterization and validation are more important to software programs, which must know what to expect in terms of byte order and internal organization of data when reading a specific file. Typically, when a software program opens a file, it validates the file against a set of characteristics, ideally, but not always, drawn from the file type specification. Although the specifications for many file types are openly published, many more are not. A file's type specification may be proprietary to a specific manufacturer and therefore not openly published, the file's characteristics instead embedded in the compiled (and also proprietary) code of the access software itself. Software programs must also have detailed knowledge of a file format's characteristics in order to write the file to disk. Other programs exist purely to validate the file,

1 Borrowing the definition of terms from the OAIS Reference Model, about which more is said below, "data object" and "digital object" may be used interchangeably and both reference a "set of bit sequences," see CCSDS, OAIS, Blue Book: 1-9 to 1-10. However, in this present document, "data object" will never refer to a "physical object."

2 The following questions are borrowed from the JHove Project documentation, see JHOVE2 – FAQ, retrieved April 14, 2011 from https://bitbucket.org/jhove2/main/wiki/JHOVE2_Frequently_Asked_Questions_%28FAQ%29.

evaluating whether it conforms to its specification. And, problematically, software developers create programs designed to be forgiving of files that do not adhere to their standards, thereby creating a situation where the file fails to conform to its specification but is rendered acceptably, if not perfectly, to the end user by the rendering software. Web browsers are excellent examples of this, as they often render poorly formed HTML.

Beyond daily reliance on these activities, it is necessary to recognize the important role this type of information plays in digital preservation. A TIFF image file, a common and well-known image file format, provides a good example. Generally, a TIFF file will have a .tif suffix, but this supplies only a hint about the file's type. According to the published TIFF file specification, the first bytes of a properly-formed TIFF file have the following markers: bytes 0-1 are either 4949 or 4d4d, indicating endianness, and bytes 2-3 must be 42 (Adobe Systems Incorporated, 1992). In little-endian, for example, bytes 2-3 should be 002a for a conforming TIFF file. File validation tools look for, and expect to find, these bytes for a well-formed TIFF file: bytes 0-4 taken together represent the TIFF file's "magic number," a signal by which the run-time program may determine the file's type (http://en.wikipedia.org/wiki/Magic_number_%28programming%29#Magic_numbers_in_files; http://en.wikipedia.org/wiki/File_format#Magic_number). A properly-formed TIFF image will include a number of additional markers, or characteristics, including vital information important to faithfully render the file, such as the image's color model and size (Figure 1). Nevertheless, even with only the byte positions and values in this example, it is possible to

1) understand how an individual might reasonably identify the file type,
2) reference the defining characteristics of a TIFF file,
3) access information vital to the image's interpretation, and
4) validate whether a given TIFF file is, in fact, what it purports to be.

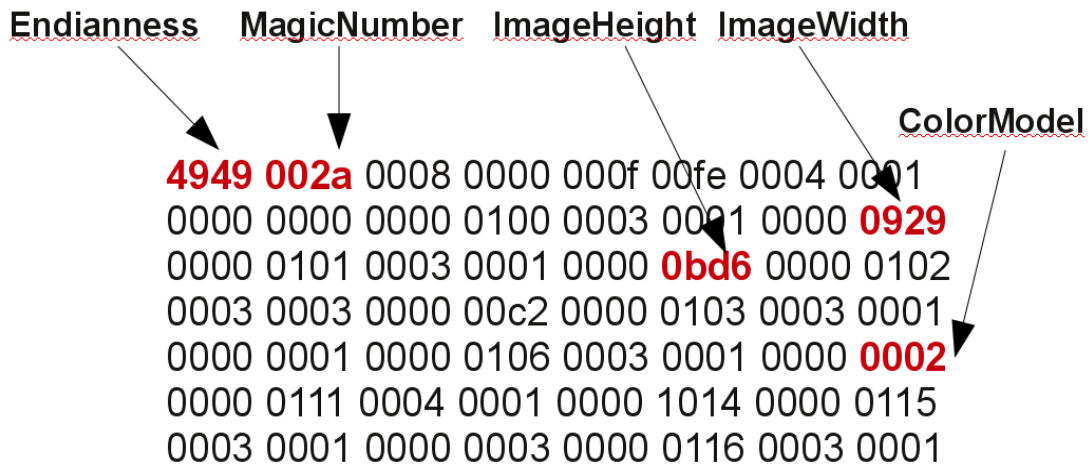

Figure 1. Initial Hexidecimal Values of a TIFF file

This technical metadata is often extracted from the data object and stored along with it in the repository, becoming part of the object's preservation metadata. This becomes important information to a system's or user's ability to successfully open the file and faithfully access its content and assess its authenticity. Technical metadata is often embedded into larger structural metadata schemas, such as within a PREMIS record (a preservation metadata schema, see Caplan, 2009), or as a standalone in-line XML stream within a METS record (METS Editorial Board, 2010).[3]

Having been captured and stored along with the data object in a repository, preservation metadata is vital to periodic evaluation of repository holdings, permitting repository managers, for example, to query all TIFF images saved using the CMYK color model for migration to a different color model should the need ever arise. Likewise, should a repository need to migrate such a TIFF file to a different file format, the technical metadata provides some information about the source file's properties that should be adequately addressed in the migrated file to ensure that the migrated file is considered authentic to the original.[4]

3 MIX (the NISO Metadata for Images in XML Schema) and TextMD (Technical Metadata for Text) are XML schemas designed for recording technical metadata. See http://www.loc.gov/standards/mix/ for MIX and http://www.loc.gov/standards/textMD/ for TextMD.

4 While the subject of significant properties, per se, is beyond the scope of this research, this notion touches on what one might consider a file's "significant properties," which is not quite the same as a file's characteristics. However, what one might consider a file's significant properties might be represented also in a file's characteristics, so there may be some overlap in data points. In many ways, a file's "significant properties" attempts to address what the file *should be* versus what it *is*. Identifying a file's "significant properties" has proven to be difficult. The properties considered significant often vary depending on who is asked the question, the file's creator or the individual tasked with preserving the file. In short, what is significant can potentially be quite subjective. Although this particular study in no way investigates the specifics of characterization in-depth

In addition to the immediate utility for repository management, many consider the ability to perform file identification, characterization, and validation actions essential to preserving data objects over the long-term. Digital librarians, archivists, and other information professionals, all of whom have played an important role identifying these activities as essential to the preservation of digital content, have been actively developing knowledge and best practices for digital preservation since the 1990s. More recently, focus has shifted from solely digital preservation (and its attendant activities of file identification, characterization, and validation) to the broader view of *digital curation*, which places greater emphasis on the life cycle of and continuing care of digital content, such as its selection, collection, archiving, preservation, and on-going maintenance (Digital Curation Centre 2010). Although there sometimes is a small tendency to favor "research" data, such as that created as a result of scientific experiments and studies (Lord, P., et al. 2004), all forms of research data, such as humanities text or digitized images of art, are in need of curation. In both cases, however, there is a basic understanding that the data be unique for research purposes. By focusing on the life cycle and continuing care of digital assets, digital curation considers all the stakeholders of digital content, from its creators, who may be scientists or historians, to those who will need future access, such as researchers or general audiences, to those who will need to ensure the content's survival (the curators themselves) so that the content remains faithful to the creators' originals and accessible to future users. Such a varied group of stakeholders equates to an equally varied type of digital content, which ranges from datasets stored in databases and spreadsheets to still images of historical documents or space imagery. Curators of this material, by definition charged with the material's preservation, will require the knowledge and means to identify, characterize, and validate a great range of content and file types from common multimedia formats to proprietary file formats.

The present study examines the current role of file identification, characterization, and validation within digital preservation, with a slight emphasis on these activities as they pertain to data curators who may be tasked with preserving material as ubiquitous as digital image and audio files to far less common material such as specially created computer programs used to access

(that is, it does not pose the question "what characteristics about a TIFF image are important to verify a migration or confirm an authentic rendering?" for example), there is substantial literature that does explore these very issues. See recent contributions Knight, G., & Pennock, M. (2009) and Hockx-Yu, H. & Knight, G. (2008). Both of these provide bibliography to older publications.

specific datasets. The next section details the motivation behind the current study, which includes the broad questions being asked and a very brief introduction to the method of testing employed to answer those questions. The following section provides not only a review of the relevant literature, but also a review of the extant tools and software developed to capture and record identification, characterization, and validation information from files. The remaining text covers in detail the methodology behind the tests conducted for this study, an in-depth analysis of the tools (and their output), and the results of the tests. The final section is devoted to a discussion of the various implications that can be drawn from the tests.

Impetus for Current Study

The projected benefits of file identification, validation, and characterization activities are considerable (Bearman 1994; Bearman and Sochats 1995; CCSDS, 2002; OCLC/RLG Working Group on Preservation Metadata, June 2002). Simply put, knowing what type of file a data object is, what version it may be, and what is characteristic about that particular file may mean the difference between accessing it in the future and being left with unintelligible bytes. As will be seen from the following literature review, the importance of these activities was determined more than a decade ago (Bearman 1994). The theoretical underpinning for these activities became official recommendations as part of the Reference Model for an Open Archival Information System (OAIS), and serves as a very influential model for digital preservation (CCSDS, 2002). While the LIS community has developed metadata formats and software tools to support these activities, formal testing of the various tools used to support these endeavors has been limited.

Testing can take various forms, from examining the accuracy of the tools' file identification capabilities to evaluating the characterization output generated by the tools. This study largely focuses on the former: how accurate are the various tools at identifying common, and not so common, file formats? To what extent can the identification be trusted?

On this basis, two approaches were taken: 1) gathering files of various formats and running the tools against those files and 2) generating specific files under very controlled conditions in an

attempt to “trick” the tools. The former strategy sought to test the coverage, reliability, consistency, and accuracy of the tools. This places a significant (and, arguably, unearned) amount of faith in the files' being of the types suggested by their extensions , which, on the surface, provide hints as to the files' types (one must assume, for example, that all TIFF files carry .tif or .tiff file extensions). To challenge the output from the tools would be to manually confirm or disprove the result for every tested file. So, based on the assumption that a file's extension accurately represents its type, the first approach tests whether the tools perform as expected and evaluates edges of their capabilities. The latter approach also challenges the idea of reliability and accuracy, but with an eye toward authenticity; this addresses the second question above concerning trust in the tools' performance.

The first question should be considered in light of the broad nature of digital curation. Generally, traditional digital collections have been composed of select file types carefully chosen by digital collections librarians with the guidance of best practices and often created under very controlled conditions. Identification, characterization, and validation tools are designed to work against file types common to digital library collections. But, when considering the broad nature of digital curation, data curators – those charged with managing datasets such as might be generated as part of a medical experiment, for example – often do not have the luxury of managing files created under their control and guidance, but instead produced by scientists and scholars whose primary objectives are their research projects and not the sustainability of their digital files. Although the term “data curation” may suggest datasets, stored perhaps as databases or XML, in reality the material in need of care is likely to range widely. Beyond database-like “datasets,” scientists capture a wide range of other data types, including imagery (such as x-rays or meteorological imagery), sound (e.g. animal communication, space noise), and video. Moreover, scientists often author very specialized computer software designed to collect, process, and evaluate data, and there is increasing interest in the preservation of this code (LeVeque 2009; McCollough 2007; Ince 2010).[5] It is more appropriate, therefore, to characterize this work more broadly, as digital curation.

As will be seen in the following section, the theoretical need to perform file type identification,

5 This has been an issue since at least Buckheit, J.B. & Donoho, D.L. (1995), which announced and introduced software, and its source code, that could be freely acquired in order to promote reproducible research on wavelets.

validation, and characterization functions has been around for some time. The current practices and software tools are the result of this theoretical research, yet few tests have been conducted that attempt to assess whether this theoretical desire is practically achievable with the current slate of software tools.

CHAPTER 2

FROM THEORY TO PRACTICE: EMERGENCE OF FILE IDENTIFICATION, VALIDATION, AND CHARACTERIZATION IN DIGITAL PRESERVATION

Historical Context and Theoretical Underpinnings

The need to correctly identify, characterize, and validate data files, specifically relational databases, dates back to at least 1986. Copeland and Khoshafian (1986), in their short piece "Identity and versions for complex objects," introduced the need for, and importance of, identifying complex objects (such as might be found within relational databases), especially of the correct version. Copeland and Khoshafian were primarily concerned with "object identity," or developing an identification schema that would, more or less, persistently identify data while accommodating a means to record versions of the data. In 1988, Margaret Hedstrom, a prominent figure in the archival community, warned about the risks of being too quick to embrace optical disks as a salve for electronic records requiring long-term preservation. She noted that many preferred to migrate the records to a "software-independent" format, but also noted that there were no "widely-adopted data and document standards" on which to rely (Hedstrom, 1988).

It wasn't until the early- to mid-1990s, however, when digital preservation began to be a topic of research interest unto itself. Foundational to the current discussion is David Bearman's 1994 publication "Reference Model for Business Acceptable Communications."[6] Bearman identified six blocks of metadata needed for each electronic record being managed. The block reserved for "Structural metadata" includes a number of sub-blocks, three of which are File Identification, File Encoding Metadata, and File Rendering Metadata (Bearman, 1994). These are basically corollaries for the three activities under discussion here and can be easily recognized as such

6 It is worth noting that the cited article must be retrieved from the Internet Archive. Bearman further developed the paper with Ken Sochats and, in 1995, both jointly published "Metadata Requirements for Evidence" (also rescued from the Internet Archive), which was a product of The Pittsburgh Project's "Functional Requirements for Evidence in Recordkeeping." The project a collection of government recordkeeping regulations, business guidelines, and other recommendations and attempted to identify a set of requirements needed for the long-term care of electronic records. Bearman goes into greatest detail, however, in his 1994 publication.

later in Bearman's paper where he specifically notes the type of information that should be recorded: File-Modality, by which he recommends values such as “text,” “numeric,” “graphic,” geographic,” “image,” “sound,” etc; File-Encoding-Base; File-Data-Encoding-Type, such as “Character,” “Vector,” “Rastor,” etc.; Data-Code, such as ASCII, UNICODE, etc; Compression-method; Rendering-rules; Dimensions; Metrics; and much more (Bearman, 1994). Notably, the tools under evaluation do, for the most part, record precisely this information.

The following year, 1995, work began on an archival standard for data by NASA representatives that would later move to the Consultative Committee for Space Data Systems.[7] The proposed standard was designed to “define an archive reference model and service categories for the intermediate and indefinite long term storage of digital data obtained from, or used in conjunction with, space missions” (Garrett, 1995). Formally published and finalized in 2002 by the Consultative Committee for Space Data Systems (CCSDS), it is known today as the Open Archival Information System Reference Model (CCSDS, 2002).[8] A set of high-level objectives and models, the OAIS Reference Model has become a foundational document for digital preservation efforts, regardless of data to be archived and technological system used for data archiving. Those creating technological solutions designed to offer some level of digital preservation know their work will be measured against the OAIS reference model by those evaluating the system (Ball, 2006; Bekaert and Von de Sompel, 2005). Knowing this, designers of such systems often measure their own efforts against the OAIS Reference Model (e.g., Tansley, Bass, & Smith 2004; Fedora Development Team, 2005). The role of the OAIS Reference Model in digital preservation and the attention given to it here are functions of the model's importance.

It was in the fourth draft of the nascent standard, published April 22, 1996, that its authors – Lou Reich from the Computer Sciences Corporation and Don Sawyer from NASA – introduced the role of representation information.[9] In OAIS terms, Representation Information is a combination of Structure Information and Semantic Information (CCSDS, 2002). Structure Information is

7 This work would come to conclusions similar to Bearman’s about the need to capture file characterization information, complete with complementary file identification and validation functions. It is unclear to what extent, if any, Bearman's publication influenced the NASA/CCSDS proceedings.

8 A mostly complete set of historical documentation for what would finally become the OAIS Reference Model can be found at: http://nssdc.gsfc.nasa.gov/nost/isoas/ref_model.html.

9 See section 2.2 in Sawyer and Reich, 1996.

more closely related to file characterization in that Structure Information describes "the format, or data structure concepts, which are to be applied to the bit sequences and that in turn result in more meaningful values such as characters, numbers, pixels, arrays, tables, etc.... The Structure Information is often referred to as the 'format' of the digital object" (CCSDS, 2002, 4-21). Semantic Information provides additional import to the Structure Information such as "special meanings associated with all the elements of the Structural Information" (CCSDS, 2002, 4-21).[10] The OAIS standard continues to evolve and the standard's committee has since introduced additional requirements to ensure the needed Representation Information about a data object is retained. The recent August 2009 draft for the revised OAIS standard introduces the notion of the Transformation Information Property, which is "an Information Property whose preservation is regarded as being necessary but not sufficient to verify that any Non-Reversible Transformation has adequately preserved information content. This could be important as contributing to evidence about Authenticity" (CCSDS, 2009, 5-8).[11] The Transformation Information Property mainly details semantic aspects of the original that must be retained and persist during any transformation (e.g. migration) of the digital object to ensure that the transformed object is not only faithful to the original but also authentic (Giaretta, 2009; Wilson, 2007).

The notion of Representation Information in the OAIS, which consists of Structure Information and Semantic Information, and the more recent proposal to introduce the Transformation Information Property seek to record similar, if not overlapping, information as is captured via the activities of file identification, validation, and characterization.

The LIS community responded to the OAIS Reference Model. Recognizing the vital role of documentation in digital preservation, work began immediately by OCLC, RLG, the Library of Congress, and a number of representatives from academic institutions, government

10 Early drafts of the standard sometimes provide more specificity than the published standard does and are illuminating in ways the final copy isn't. Version 8 of the pre-CCSDS draft included example scenarios that demonstrate, in detail, how Representation Information for images would be "computer-readable information describing the format of the images.... The representation information may specify that each image consists of 1000 scan lines, with each scan line containing 800 pixels, and with each pixel represented by an unsigned 16-bit integer value" (Sawyer and Reich, 1996, section 4.0).

11 Although an exhaustive search was not conducted, the notion of a Non-Reversible Transformation appears as early as the fifth CCSDS White Book version released April 21, 1999, see page 83 in Don Sawyer and Lou Reich, *Reference Model for an Open Archival Information System*, White Book, Issue 5, April 21, 1999, retrieved October 17, 2010 from ftp://nssdcftp.gsfc.nasa.gov/standards/nost/isoas/int08/CCSDS-650.0-W-5.pdf.

organizations, and national libraries that would result in a *Framework* for preservation metadata (in 2002; it was begun before OAIS was final, but based on it) (OCLC/RLG Working Group on Preservation Metadata, June 2002), a *Data Dictionary* for preservation metadata with complementary XML serialization (begun 2003, completed in 2005) (PREMIS 2005), and supporting technical metadata schemas catering to specific file types (one in late 2001/early 2002; the other in 2004) (http://www.loc.gov/standards/textMD/, http://www.loc.gov/standards/mix/). The work by these organizations and groups highlighted and strengthened the activities of file characterization, identification, and validation within digital preservation. Not only did they further define the types of documentary information that needed to be captured and recorded, but they also codified how this information could be recorded for storage in repository systems. The *Data Dictionary*, for example, supplies room for information about a data object's fixity (identifying hash values), format (of the file type), and various other "significant properties," or what OAIS would consider part of a data object's Representation Information. MIX, one of the technical metadata schemas, provides a means to record identification and characterization information for still images, such as information about an image's Color Profile, ICC Profile, YcbCr Sub Sampling, format considerations, such as Codecs and Codec Versions, and much more. Greater historical details surrounding these developments may be found in Appendix A.

CHAPTER 3

DEVELOPMENT OF SERVICES AND SOFTWARE TO ASSIST FILE IDENTIFICATION, CHARACTERIZATION, AND VALIDATION

The LIS and Digital Preservation communities have in fact developed registries in addition to other tools, services, and resources to aid in file type characterization, identification, and validation in order to capture and record the types of preservation metadata recommended by PREMIS. In the following sections, these are covered in some depth as they relate directly to this study. Table 1can be used as a general reference about each tool or service to be discussed.

Table 1: General Overview of Tools and Services

	Organization	Tool/Service	Primary Function	Started
PRONOM	The National Archives UK	Service	File Format Registry	2002
LC Dig Pres	Library of Congress	Service	File Format Registry	ca. 2004
GDFR	Harvard Univ. Libraries (HUL)	Service	File Format Registry	2002
UDFR	CDL (originally HUL), Portico, Stanford Univ.	Service	File Format Registry	2009
DROID	The National Archives UK	Tool	File Identification	2005
JHove*	HUL	Tool	File Charact., Valid., Ident.	2003
MET	Nat'l Lib. of New Zealand	Tool	File Charact., Valid., Ident.	2003
TrID	Marco Pontello	Tool	File Identification	2003

* Jhove2, begun in 2007, is associated now with CDL

File Format Registries

Established in 2002, the United Kingdom's National Archives has developed a technical registry called PRONOM (http://www.nationalarchives.gov.uk/PRONOM/) which “is an online technical registry providing impartial and definitive information about file formats, software products and

other technical components required to support long-term access of electronic records" (The National Archives[, UK], n.d.). The *Sustainability of Digital Formats* website (http://www.digitalpreservation.gov/formats/) offered by the Library of Congress (LC) is similar to the UK's PRONOM. Both websites collect detailed information about specific file types, links to relevant material, such as file specification documentation, and map the relationships between the various file types. Harvard University Libraries were also early pioneers of this work with the Global Digital Format Registry (GDFR), which was formally discussed as early as 2002 and later became a joint project with OCLC and the US National Archives and Records Administration (NARA) (Abrams, 2005).[12] GDFR was cited in the PREMIS Data Dictionary as the possible model for a format registry (PREMIS Working Group, 2005; PREMIS Editorial Committee, *Data Dictionary*, 2008). Although the project languished for many years after initial funding ended and, in the meantime, was superseded by PRONOM and the LC's Sustainability of Digital Formats website, the GDFR received renewed attention in early 2009 when it was decided to partner officially with the UK's PRONOM team and develop a new registry, the Unified Digital Format Registry (Unified Digital Formats Registry, 2009). The new registry will be based on the PRONOM database. Desiring to marshal limited digital preservation resources in a collaborative and distributed way, the partnership will leverage the existing PRONOM database and technology and marry it to the GDFR model, thereby removing the control of the database from a single entity (The National Archives, UK) and sharing it as a partnership. The data would be duplicated across several institutions.

PRONOM

Of the active format registries, PRONOM is the most comprehensive. Although not complete, PRONOM's registry contains entries for a wide variety of file types and formats: In addition to entries for MS Word and Excel files, and their OpenOffice equivalents (*.odt, *.ods), all of which are so common as to be expected, PRONOM includes entries for the SQL file type, MS Access database file types, dBase files, and FoxPro files (the latter two are older database products). But the information PRONOM does include about these files types is minimal at best, making the entries for these files types like placeholders awaiting further development. The relative dearth of information about an MS Excel 2007 file

12 For historical purposes, the GDFR website is still active at http://www.gdfr.info/.

(http://www.nationalarchives.gov.uk/PRONOM/fmt/214) is readily apparent when the PRONOM entry for the Excel file is compared to PRONOM's entry for a version 6 TIFF file (http://www.nationalarchives.gov.uk/PRONOM/fmt/10). That many of these file types are proprietary suggests why these records might be sparse – the necessary information has not been published, or cannot legally be included in the registry.[13]

In all likelihood, these issues will be resolved. Integral to the National Archives UK's digital preservation plans, Brown (2007) described a digital preservation system where file format characterization would be one of the initial actions performed on a data object, including file format identification, validation, and property extraction. In the meantime, PRONOM has become a valuable resource within the digital preservation community.

By being a rich, publicly-accessible source of information for the community, PRONOM has made itself available as a component to be integrated with other digital preservation tools or services. Brody et al. (2007) describe a service created as part of the *Preserv Project* that uses OAI-PMH to access repository content so that it could be evaluated using the National Archives's PRONOM database and software (see DROID below). Ferreira et al. (2006; 2007) discuss the potential use of PRONOM as part of format detection and format evaluation services within a preservation system. These studies demonstrate how identification, characterization, and validation activities have been integrated into digital repository workflows. And they also reveal the breadth of information needed to preserve digital information. The studies reveal how a central, networked resource not only reduces the considerable efforts required for good digital preservation but also underscore that sustainable digital preservation is a community effort from developing high-level services (such as developed by Ferreira et al.) to recording detailed information within the PRONOM database.

Whereas PRONOM seeks to provide file format information for any and all types of data objects, the LC's *Sustainability of Digital Formats* database is strongly biased toward images, audio, video, and some text-based file formats. It should be stipulated, however, that the LC website is

13 It may also be due to a lack of manpower, as suggested by the fact that newer MS file format specifications are available as ISO standards and Microsoft has published information about its older Office file formats. For the newer, ISO documentation see http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51463. For the older (and newer) MS file formats, see http://msdn.microsoft.com/en-us/library/cc313105.aspx.

basically publicly available information that exists to address internal policy needs (Arms & Fleischhauer, 2005). It therefore does not seek to be a comprehensive database of all existing (or historical) file formats. As such, not only does it not contain entries for MS Excel spreadsheets (.xls or .xlsx), but it does not even include entries for the ubiquitous MS Word formats (.doc or .docx). Although the LC, at least in 2005, was working closely with those developing the GDFR at Harvard University (Arms & Fleischhauer, 2005), the *Sustainability of Digital Formats* website appears to be a rarely-updated resource designed for human, versus machine, consumption.[14] Unlike PRONOM, there appears to be no associated services, overarching strategy for the database, or complementary software.

File Format Identification, Characterization, and Validation Tools

Beyond format registry databases, the digital preservation community has also created software for use with file format characterization, identification and validation. The UK's National Archives has contributed DROID (http://droid.sourceforge.net/), which relies on the information in the PRONOM database. DROID – Digital Object Record Identification – automates file identification and can be integrated into digital preservation workflows, as described by Brody et al. in 2007 as part of the PRONOM-ROAR project, "a file format profiling service that uses the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) standard and the PRONOM-DROID tool from the UK National Archives" (Brody et al., 2007). When given a list of files, DROID takes each in turn and attempts to determine the file type by taking technical characteristics of the file and querying, essentially, the PRONOM database. To do this, the DROID software automatically detects and downloads an up-to-date file format signature file from the PRONOM registry against which files are identified and validated. The signature file includes characterization information – file extensions, magic numbers, and other specific markers – that the software uses to identify a given file (Brown, 2007). DROID's strength is the

14 It has been difficult to establish whether and how often the LC format database is updated. Although few updates to the LC's *Digital Formats* website were detectable earlier in 2010, update activity has increased significantly since March 2010. Google reports more than 40 updates to the database since this time. Although not conclusive (a recent "last modification" date can easily overwrite a "last modification" date of indeterminable date), Google reports only 4 updates to the site during 2009, 12 for 2008, 22 for 2007, 8 for 2006, 39 for 2005, and 64 for 2004, the first year for which Google has data. The relatively large numbers for the 2004 and 2005 combined with the significantly smaller numbers for 2006-2009 suggest few updates were made to the database the last few years, or the LC regularly modified existing records, annually. The previously noted numbers were current as of Summer 2010.

PRONOM database and its leveraging of networks to maintain an up-to-date signature file.

At the time of this writing (Summer 2010), and used in the tests discussed below, the current signature file is version 35; the signature file contains more than 700 file format types, 213 of which have a corresponding internal signature component containing at least one byte sequence to validate a file's type. Appendix B reviews how DROID uses the signature file to identify a file and the confidence with which it makes that determination. DROID comes with a graphical user interface, permitting evaluation and use by non-programmers, and it also includes a command line executable, facilitating integration to systems as a service, as noted above. Integrated within a preservation system, DROID provides a useful, self-updating component that repository managers can integrate into their own systems and workflows, as demonstrated by a number of projects that have taken advantage of DROID's architecture (Brody et al., 2007; Ferreira, Baptista, & Ramalho, 2006; Ferreira, Baptista, & Ramalho, 2007). DROID is a model component demonstrating how digital preservation actors can collaborate on individual required aspects of digital preservation.

Where DROID seeks simply to identify a file type, others have developed tools that gather extensive technical metadata about data objects. JSTOR and Harvard University Libraries have developed JHove (JSTOR/Harvard Object Validation Environment), another tool for file format identification and validation (http://hul.harvard.edu/jhove/) that was first released in 2004 (http://hul.harvard.edu/jhove/oldnews.html). JHove comes with a graphical user interface or it can be used from the command line, which is conducive to integrating it with digital repository workflows. Designed to be extensible, JHove uses specially created modules programmed to evaluate specific file types. The original developers created modules for a variety of image file types (TIFF, GIF, JPEG, DNG, JPEG-2000), audio types (AIFF, WAV), some markup and text-based types (XML, HTML, UTF8, ASCII), PDF, and a generic bytestream (http://hul.harvard.edu/jhove/documentation.html). Users have contributed a few third-party modules: one for MP3 files and one for Zip and Gzip files (http://hul.harvard.edu/jhove/thirdpartymodules.html). JHove is able to export the technical metadata, which includes file identification and validation information and detailed characterization information for the given file, in XML format for storage alongside the data object in the repository. A detailed discussion, with examples, of the JHove output is discussed

in Appendix B.

JHove's modular design exemplifies a strength and weakness of the software. Being extensible, anyone can create a module for use with the tool. On the other hand, this type of flexibility permits end users to circumvent what might be a larger issue: the creation of a non-conforming, non-standard file. For example, the Library of Congress uses the software for its National Digital Newspaper Program (Littman, 2006) for which it was necessary to modify the basic TIFF module to accommodate the special TIFF files generated by the Library that JHove was initially unable to handle (Littman, 2006).[15] Naturally, however, this raises very real questions, no less of which is, "What does it mean when, in the future, a repository manager tries to validate the file against a new version of JHove, or a different tool altogether, and the tools is unable to identify, validate, and/or characterize the file as expected?" This may be one isolated case, and to the implementers it may appear this way, but it also establishes a procedure that deviates from expected practice.

In the period following the official release of JHove1 in mid-2005, a number of other technical issues and ideas have arisen, which, if addressed, could strengthen JHove. In response, development of JHove2 began in 2007, originally as a joint project between Harvard University, Portico, and Stanford University (Harvard University would later exchange its position with the California Digital Library) (http://www.jhove2.org). These new efforts are outlined in Abrams (2009). One of the primary items on the list is to redesign JHove to sufficiently handle complex objects, such as a TIFF image with embedded XML metadata and embedded color profiles (Abrams, 2007). A significant modification is that JHove2 will also separate the functions of file identification and validation. This will permit JHove2 to more reliably *identify* a file even if JHove2 is unable to *validate* it (Abrams, 2009). It will make JHove2 more flexible than its predecessor and allow JHove to more easily integrate other, complementary tools such as DROID.

The National Library of New Zealand has developed and published a tool very similar to JHove.

15 Littman (2006) states in full: "It is the NDNP team's experience that many TIFF generators and most TIFF renderers ignore the word offset rule, which is ambiguous in the TIFF specification." The issue of 'strict adherence to published file specification documentation' versus a more passive and forgiving relationship with file specifications is discussed below.

Begun in 2003 as an internal tool, The Metadata Extraction Tool was published as an open source offering in 2007 (http://meta-extractor.sourceforge.net/) (Knight, 2005). Predicated on an extensible modular design, it, too, is capable of identifying and validating a variety of image file formats (TIFF, JPEG, GIF, BMP), audio files (WAV, MP3), and markup documents such as HTML and XML. And like JHove, it has a graphical user interface and can also be integrated with server applications from the command line. The Metadata Extraction Tool, though, also has support for MS Office documents, such as Word, Excel and Powerpoint, and OpenOffice documents. Until June 2010, when a new release of the Metadata Extraction Tool was published, the last official release had been 2.5 years earlier, in December 2007 (http://sourceforge.net/projects/meta-extractor/files/).[16] The new release made a number of changes, corrected a number of bugs, and added three new adapters. Metadata Extraction Tool serializes the results as XML, though in a format different than JHove's or DROID's formats. Appendix B includes some discussion about the data produced from the Metadata Extraction Tool.

Two additional tools, both of which were not developed specifically for digital preservation purposes and by LIS community members, deserve mention: the UNIX file() command and TrID. The file() program, which was originally developed for UNIX but was reimplemented as an open-source offering for Linux and BSD (Berkeley Software Distribution), is a powerful file identification tool available on nearly all Unix- and Linux-based systems (http://www.darwinsys.com/file/). Just as DROID relies on a signature file, which contains the magic number byte sequences for files it can identify, the file() program uses a magic numbers database to assist its file identification actions. Although it is difficult to determine how many types of files the file() program is capable of identifying, the magic number database contains more than 1,900 zero-position offset checks. It is tempting to think that file() therefore tests for more than 1,900 different file types (or, more accurately, versions of file types), but it is likely that a portion of those 1,900 tests represent variations of offset checks for the same version of the same file type, and not distinct file types. The file() command also has the capability to execute semantic checks within a given file in order to more accurately identify its type. For example, the file() program can look into text file types, such as computer code, and attempt to determine

16 The tests in this paper, having been conducted prior to the June 2010 release, relied on the 2007 version of the Metadata Extraction Tool.

more precisely the specific type of computer code file, such as Ruby code, Perl, and Python. The file() command design is flexible enough, and with a simple yet customizable file type definition syntax, that it would be possible to perform even more sophisticated file type checks than file() already does. Such is the breadth of the file() command's magic number database that the JHove2 project members have noted that it could be leveraged to expand JHove2's coverage of file types for identification (on top of those identifiable by DROID, which JHove2 will also leverage) (Abrams, 2009). The file() program, which was used in the present study, can be configured to stop after the first match or to continue looking, presumably, in an attempt to find the best match.

TrID is a non-open source file identification software (http://mark0.net/soft-trid-e.html). It is free for "personal/non-commercial" use. It is also does not appear to be a product of the LIS community. A one-man operation, its creator, Marco Pontello, appears to have made it primarily for use when attempting to recover from data loss, during which recovery software may be able to salvage files but only as blocks of unknown data. TrID attempts to identify those blocks of unknown data. According to the documentation, TrID is capable of identifying 4,173 file types. Like DROID and the file() command, it appears to rely on a substantial database of file type information (presumably magic numbers), though it appears to have some support to detect more specific semantic information between file types, such as strings that might be found globally within a file (http://mark0.net/soft-tridscan-e.html). TrID provides a command line and graphical interface; the latter is designed only for a MS Windows environment. Interestingly, and promisingly, TrID provides an additional tool created to scan a group of files with the intent of generating an identification profile (http://mark0.net/soft-tridscan-e.html) that can then be merged with the larger identification database. Like DROID, TrID's primary objective is file identification, with validation only being a residual benefit and then only as rigorous as the process of identification. However, TrID routinely returns multiple results per file, each with a percentage value indicating how confident TrID is in its identification. Although not formally part of the results presented here, a broad test suggests that TriD is frequently less than 70% positive about an identification.

Recent Developments - Bundling of Digital Preservation Tools

A recent trend in developing file format identification, validation, and characterization services for digital preservation has centered on bringing the disparate resources and tools together under one umbrella software offering. One such offering is Harvard University Library's File Information Tool Set (FITS) (http://code.google.com/p/fits/). FITS integrates DROID, JHove, the Metadata Extraction Tool, and a few other tools not under review here because of their limited applicability.[17] The future version of JHove, JHove2, will also integrate DROID (Abrams, 2007). The PLANETS project (Preservation and Long-term Access Through Networked Services) is another such offering (http://www.planets-project.eu/).

But, more than a software platform, PLANETS attempts to provide an entire practical and implementation-ready *framework* for digital preservation. PLANETS, composed of a matrix of software, provides assistance with digital preservation planning and decision making. This includes help with evaluating all of the data objects in need of preservation, the level of preservation those objects require, and finally, a program to best implement the digital preservation requirements identified for those data objects. The planning and services offered by PLANETS includes identifying the data objects, providing migration strategies depending on the file type, assistance with identifying what the preservationists believe to be file characteristics that must be retained in migration, and, finally, the migration services, complete with a check to see if the migration preserved the characteristics originally identified by the preservation manager. To achieve these tasks, PLANETS has integrated a number of the existing tools and services into its offerings, such as JHove, DROID, PRONOM, and the Metadata Extraction Tool. PLANETS has also developed a few of its own, such as SIARD (Software Independent Archiving of Relational Databases) (Swiss Federal Archives, n.d.). SIARD was created by the Swiss Federal Archives, a member of the PLANETS consortium, as "a sustainable solution for the long-term preservation of relational databases" by representing the data in an open, non-proprietary format (Swiss Federal Archives, 2088). The eXtensible Characterization Language (XCL) (http://planetarium.hki.uni-koeln.de/planets_cms/about-xcl) is another tool developed by PLANETS. Described by Becker et al. (2008), XCL strives "to express the complete

17 Exiftool, which extracts embedded Exif data from digital images, is one such tool. FITS also integrated a component that "normalizes" the results from the three tools for easy comparison purposes.

information content about a file in a format independent model" (Becker et al. 2008, 406). In other words, XCL seeks to semantically characterize file format information across different file formats that are of essentially of the same type (JPEG, TIFF, PNG are all images, and so on) by basically abstracting the characteristics of a number of similar file types. It can then compare XCL data from the original file to the XCL data of the migrated one.

Community Evaluation of Services, and Tools

Beyond being used in production systems, the tools have undergone little formal testing. The most recent, extensive formal test appears to have been conducted in 2009, by Artefactual Systems, which has performed one of the few tests to look at the Metadata Extraction Tool (Artefactual Systems, Inc., *DROID, JHOVE, NLNZ Metadata Extractor*, 2009). In addition to comparing the features of DROID, and the Metadata Extraction tool, Artefactual took a sample of files, one each representing a different file type, against which to run the tools (Artefactual Systems, Inc., *Test File Results*, 2009). Of the 24 file types Artefactual tested, 17 (71%) were positively identified by DROID (29% "not identified"); Metadata Extraction Tool identified 18 files (75%); and JHove identified 10 files (42%). In 2006, DROID and JHove were studied as part of the University of London's Computer Centre Digital Asset Assessment Tool Project (University of London Computing Centre, Dec 2006). For DROID, the University of London report noted a number of issues with false-positive identifications and, ultimately, of 77 files tested, DROID was unable to identify nearly 50% of the sample. The sample included a wide variety of file types, from common image types (JPG, PNG) to system specific files such as Windows system file types (SYS, CAT, MST) (University of London Computing Centre, Dec 2006, Appendix 1). The test performed on JHove was predictable: those types JHove was programmed to identify and characterize, it did; those it was not programmed to identify, it did not. The false-positive file identification issues with DROID have been the subject of recurring criticism (e.g. IDEALS, 2008; Brody et al. 2007). A study of the performance of JHove and DROID noted that DROID and JHove performed "equally well," "statistically" for file format types *JHove* was programmed to identify (Nguyen, 2008). Ending in 2009, a study of the University of Southampton's Preserv2 EPrints toolkit (which integrates DROID) demonstrated a 93.1% "classification rate" (taken to be the accurate identification of a file type), which is

corrected to 92.75% when including wrongly classified files (Preserv.co.uk, n.d.).[18] A number of other reports also discuss the quality of the information generated by the various tools (Anderson et al., 2006; Prom, 2010; IDEALS, 2008; Nguyen 2008). While the samples of files examined by Artefactual and University of London were small, those tests examined a broad selection of file types. Conversely, although the Preserv2 Eprints toolkit examined a substantial number (2,144) of files from a number of repositories, these files were overwhelmingly common files, the vast majority being PDF documents (for some repositories, PDF files made up more than 95% of the collection) (http://www.preserv.org.uk/testing/repositories/).

As will be seen in the coming pages, identification, validation, and characterization of common file types, such as PDF files or JPEG images, are very well supported by the tools. Therefore, the very high accuracy reported in the Preserv2 Eprints study could be anticipated, as is the much lower numbers of the other two studies. The Preserv2 Eprints study is also interesting because it compares two different versions of the DROID signature file, versions 12 (released August 2006) and 13 (dated September 2007). Embedded in the results are a number of instances wherein the classification changed. For example, in at least two instances, version 12 of the DROID signature file identified a file as a PDF but version 13 of the signature file identified these same files as HTML files (http://wiki.preserv.org.uk/index.php/TestDataResults#Soton_.2897_Files.29; http://wiki.preserv.org.uk/index.php/TestDataResults#Tartu_.2898_Files.29). Other tests revealed that, of 944 files examined, 2 were newly "unknown" by version 13 of the signature file (http://wiki.preserv.org.uk/index.php/TestDataResults#Typical_Repository_Outcomes_.28994_Files.29). That such inconsistencies might be introduced from version to version highlights the potential fragility of these tools, and accentuates the need for these tools to be nearly perfect in a digital preservation environment. For identification and/or validation results to change when the software performing the identification and/or validation is updated, inconsistency is introduced into a process whose very legitimacy is predicated on regular, consistent results over time. Without human interpretation of the results, it becomes an enormous challenge to program the machine to reliably determine that the PDF file is an HTML file or vice versa.

Good, theoretically-scalable tools and services have been developed and continue to grow, but

18 See also, http://wiki.preserv.org.uk/index.php/TestDataConclusions

tool and service development is slow and expensive. (PRONOM, with its trove of file format information - down to the byte level in many, many cases - stands alone among these tools, though DROID, which relies on PRONOM, is a close second.) Yet, despite these strides, tools remain error prone, inaccurate, too strict, and too narrowly scoped. And, new file formats are continuously introduced, creating a game of constant catch-up for tool and service developers. The scale of the problem has raised questions in the community about the usefulness of these tools, and has motivated others to modify the tools to accommodate files that may deviate from a file's published specification. As noted above, tests of these tools have mainly focused on a broad spectrum of file types while testing only a few files; some others have examined larger numbers of files with limited variation in file formats tested. While published results have noted inaccuracies in the output from these tools, none have tested the degree to which those results might be manipulated by examining a file intentionally engineered to trick the tool. The following examination looks at a very large sampling of files of various file types created under a number of different conditions designed to bring the strengths of the previous test together under one cover. At the same time, a few tests are performed aimed at testing how easy it is (or is not) to manipulate the results from the tools.

CHAPTER 4

METHOD

Three tools – JHove, DROID, and the Metadata Extraction Tool – were selected for testing. All are employed by repositories and digital archives; all are open-source software tools developed by LIS community members in support of digital preservation activities. They exist for the express purpose of identifying files and extracting technical metadata for use in the long-term care of digital objects. A fourth tool, the Local Tool, was developed. It relies largely on the Unix/Linux file() command.

Two different experiments were run in order to evaluate the tools. One attempted to manipulate the results returned by the tools. The other turned the tools on a large sampling of files to evaluate their coverage of file types.

File and Tool Manipulation

Files were specially created to test what factors might lead to a "positive," "tentative," or "not identified" identification. To this end, three files were created in a hex editor: a TIFF file, a WAV file, and a PDF file. These file types were chosen in part because they are in wide use and also because their file specifications have been published. As such, the files were created in consultation with their specifications so that they contained the correct magic numbers (and sequence) and to determine what else might be necessary for a file to be considered valid. As for data, the only content in these files were their magic numbers. The files were saved with the appropriate extension according to their type. The TIFF file is 4 bytes; the WAV file is 16 bytes; the PDF file is 15 bytes. These files and the results from the tools are available at http://www.3windmills.com/thesis/.

Data Collection and Tool Configuration

From a high-level view, the work proceeded as follows: A collection of digital content was

identified and harvested by downloading from an internet location to a local workstation for testing. Each of the tools was configured and run against each harvested file. The resulting XML output from the tests was saved to disk and later analyzed.

The data and preliminary analyses are available at http://www.3windmills.com/thesis/

Files for evaluation were harvested from five distinct projects:

1) Internet Archive
2) Chronicling America, Library of Congress
3) Performing Arts Encyclopedia, Library of Congress
4) Prints and Photographs, Library of Congress
5) Personal User Files

A brief description of each group follows including the general rationale for inclusion.

Internet Archive

The Internet Archive (IA) seeks to preserve information from the World Wide Web and other born-digital content to protect against their loss and save them for future reference and study (http://www.archive.org/about/about.php). The harvested items from the Internet Archive (1,135 files) represent material purposefully uploaded to the IA, versus material collected by IA, such as archived websites.[19] A strict date-based from 1 Jan 2010 to 1 Feb 2010, which reflects when material was added to IA, was performed. Because a date-based search was executed, it is presumed the file types represented in the hits do not give preference to one particular file type over another. However, it is reasonable to assume that the hits reflect files created recently, with recent software, and employing now-common file formats.

The 130 items harvested from the IA yielded 1,135 individual files.[20] Each item

19 IA permits anyone to upload material, providing the have the rights to, and, although IA documentation makes specific mention of media file formats, the IA can and does accept any type of content (http://www.archive.org/about/faqs.php#Uploading_Content).

20 The IA collection was originally intended to be larger. It was initially decided to capture 2,000 items from the Internet Archive, but after hours of operation the harvest was halted, having only downloading 130 items, for reasons of time (the slowness of the harvest raised questions about the efficiency of the method) and space (the 130 items were averaging 500MB each; they consume 37 GB of disk space).

contains two XML files, one representing the item's metadata and the other the individual files associated with each item. The remaining 875 files are of various formats, and represent various parts of the 130 items, including multiple video parts, multiple audio parts, thumbnails, etc. Part of the ingest process includes creating derivative files (http://www.archive.org/help/derivatives.php). For video, this might mean creating a video for delivery for the World Wide Web. Additionally, the Internet Archive creates animated GIF files for videos. These function as preview images for the video, flashing a frame capture of the video every few seconds. For images, thumbnails and other size representations may be created. In addition to capturing audio, video, and image files, which constitute the bulk of the content captured, Microsoft PowerPoint files and other documents were also harvested.

Library of Congress Collections
Prints and Photographs (390 files evaluated)
Chronicling America (600 files evaluated)
Performing Arts Encyclopedia (1,324 files evaluated)

Files from three distinct Library of Congress collections were harvested. Given their source (the LC), the collected files are assumed to adhere to the standard and accepted digitization best practices that were in place at the time of their creation. Despite the presumed uniformity of their creation, the files that constitute the collections were generated by three distinct divisions of the Library, each of which used its own equipment and personnel. These three file sets are meant to function, therefore, as a point of comparison to file sets that come from sources with less controlled conditions, such as those from the Internet Archive and the Personal User Files. The bulk of the file types harvested from the Library of Congress are text files, XML files, images, audio, and video.

The files from the Prints and Photographs Online Catalog (PPOC) (http://www.loc.gov/pictures/) are all images. Based on the creation time of some of the files, some are more than a decade old, dating to 1998. Given the fact that all of the digital items in PPOC are images and created by the same division in the Library, only 60 items were harvested. These were divided between two sub-collections: Baseball

Cards and Civil War Photographs. The 60 harvested items yielded 390 files for evaluation (TIFFs, plus JPEG and GIF derivatives).

The files from Chronicling America (CA) (http://chroniclingamerica.loc.gov/) were harvested based on a simple word query search ("overcoats"). The first 100 items were selected. Each CA item captured for study consisted of six files, for a total of 600 files for analysis: one JPEG thumbnail, one JPEG2000 file, one RDF/XML file, one ALTO XML file, one simple text file, and one PDF file. The CA project is a relatively recent project that is still on-going. The files are fairly evenly divided between those in XML format, master and derivative images, and text files containing the output of images passed through an optical character recognition program. Chronicling America is part of the National Digital Newspaper Program (NDNP), a joint partnership between the National Endowment for the Humanities and the Library of Congress. The Library has published strict digitization guidelines for any grant awardee submitting image files to the project (National Digital Newspaper Program , 2010).

Most of the items from the Performing Arts Encyclopedia (PAE) (http://www.loc.gov/ihas/) were created by the music division though it is possible some were created by other units in the Library. PAE holds a mixture of resource types, from video to XHTML, which was its attraction. The list of 111 items harvested was created by repeatedly searching the PAE website and selecting items based merely on compiling a list representing a variety of resource types.

Personal User Files (6,109 files evaluated)

Files were harvested from an anonymous individual. This yielded more than 15,000 files. System-created back-up files were removed, as were all XML and HTML files, which reduced the number to around 9,000 files. The list was further reduced by 3,000 files by removing those deemed duplicative in so far as they would add little to the sample; that is, they were redundant based on format and character. For example, of twenty MP3 files representing 20 tracks from an audio CD, 15 may have been deleted

since all 20 files were created within the same 10 minute span by the same audio software. Thousands of computer source files were deleted based on the same criteria.

The remaining files represent the widest spectrum yet for analysis. Although a large number of the files are images, the others might be computer programs, source code, databases, spreadsheets, office documents, all of which are in a variety of formats, including different versions of the same file type (MS Word files created by different product versions of Microsoft Office; different versions of PDF files).

For all but the Personal User Files, the capture included a metadata record that contained pointers to the various files comprising the complex digital object. For example, a Chronicling America item might contain a thumbnail image, two forms of XML data, a PDF file, and a JPEG2000 image. Except for the metadata file, the other files were captured using *wget*, a command line program designed to download data over HTTP from the World Wide Web. *Wget* is eminently suited for such a task because of particular features, including protection against timeouts, downloading files of unlimited size, and an option to preserve a file's created and last modification times; these are important as they facilitate the accurate capture of files' characteristics.

Once a harvest was complete, each captured file was analyzed in turn by the three tools. JHove could be run as is, or with an option that attempts to identify the file based purely on the file's internal characteristics (such as its “magic number”). Therefore, JHove was run over the files twice, once using the default options and a second time instructed only to evaluate a file's internal characteristics. For JHove-with-S-option the output is very similar to what DROID produces.

The Metadata Extraction Tool also came with two options. One that attempted to identify the item based on modules created by the National Library of New Zealand, hereafter referred to as MET-1, and another option that sought to not only identify the file but also record detailed provenance information pertaining to its location in the file system, hereafter referred to as MET-2. Like JHove, the Metadata Extraction Tool was run over the files twice, once for each option.

DROID has no options that might result in a meaningful difference, and was therefore run over the files just once. All three tools provide a way to save the program output serialized as XML. All of this work was executed on the command line using a number of specially crafted scripts – the graphical user interfaces included with each tool were not used.

Finally, a very simple script was created – hereafter referred to as the Local tool – that attempted to replicate the (identification) work of DROID, JHove, and the Metadata Extraction Tool but using common functions part of most programming languages (such as the ability to derive a file's most recent modification time) and programs common to most Linux operating systems (primarily the file() command, which exists to identify file types).

Many more details about data normalization decisions, based on the varied output of the tools under investigation, may be found in Appendix B. But briefly: after processing the files through the various tools, the data were normalized to facilitate comparison of the results. XSL transformations were created for these purposes, including ones that crunched the output resulting in aggregated, quantitative data. Finally, the data and files were uploaded and a website created to assist with further analysis by providing a way to view the aggregated results and underlying source data in an organized fashion.

CHAPTER 5

RESULTS AND ANALYSIS

In general, test files created under controlled conditions were consistently and "positively" identified by the tools; that is, they were reported to be of the types expected based on file extension. If, for example, a file had a JPG extension, then the tools generally identified it as a JPEG image file. If it had a WAV extension, then a WAVE audio file, and so on. One must also assume that the files' internal characteristics not only matched their specifications but also that the tools were correctly programmed to identify files based on their published characteristics, such as appropriate magic numbers.

Such a positive identification should not necessarily be interpreted as "correct." DROID, JHove-with-S-option, the Local tool, and, to lesser degrees, MET-1 and MET-2 can be "tricked." The tests performed by these tools quickly check one or two data points per file: the magic numbers (if present and applicable) and file extension. If they align, the tool "positively" identifies the file as a specific format. Because the tools check so few data points, when these tools are fed files that have been specially crafted to contain only the correct magic numbers and carry the correct file extension, they are likely to "positively" identify the file. A "positively" identified file is not the same as one that is valid and conforming to its specification, and, above all, one that can be opened.

File Manipulation Results

The tools – JHove, MET, DROID, and the Local Tool – identified the specially created files as follows.

> TIFF: Size, 4 bytes. Per the identification matrix developed for this project (see Appendix B), all tools – JHove, JHove-with-S-option, MET-1, MET-2, DROID, and the Local tool - positively identified the file as a TIFF file. DROID's identification was "generic," meaning it identified it was a TIFF file, but was unable to determine the

TIFF version. Interestingly, MET-1 identified it as version 6. JHove exhibited conflicting information: JHove indicated a signature match with the TIFF-hul module, but ultimately used the ASCII-hul reporting module (versus the more generic BYTESTREAM). The GIMP and Gwenview, two image programs, reported an error in the file when failing to open the image.

WAV: Size, 16 bytes. The results were nearly identical to those for the TIFF file. JHove, JHove-with-S-option, MET-1, MET-2, and the Local tool positively identified the file as a WAV audio file. DROID's identification was "tentative," meaning it identified it was a WAV file based on file extension alone, but was unable to more precisely determine the WAV version. JHove exhibited conflicting information: JHove indicated a signature match with the WAV-hul module, but ultimately used the ASCII-hul reporting module (versus the more generic BYTESTREAM). Notably, two audio programs, VLC and Amarok, opened the file without complaint.

PDF: Size, 15 bytes. JHove-with-S-option, MET-1, MET-2, the Local tool, *and* DROID positively identified the file as a PDF file. JHove, however, did not determine a signature match, but it still reported it a well-formed and valid ASCII file. Desktop software was not able to open the file.

File Manipulation Analysis

When reading the specifications, it can be difficult to determine what aspects are necessary for the file to be considered "valid," or what is needed for successful and correct "identification." The TIFF specification states that "a TIFF file begins with an 8-byte image file header that points to an image file directory (IFD)" (Adobe Systems Incorporated, 1992, 13). Not having an image file directory, the above TIFF image is only 4 bytes. It seems safe to conclude that the tested file does not meet the minimum requirements. The documentation for the PDF specification is less definitive (Adobe Systems Incorporated, 2008). The specification employs the word "shall," noting that "the first line of a PDF file shall be a header consisting of the 5 characters" (Adobe Systems Incorporated, 2008, 39). The text continues that "the body of a PDF file shall consist of

a sequence of indirect objects representing the contents of a document" (Adobe Systems Incorporated, 2008, 40). While it might be safe to assume that the header of a valid PDF *must* begin with 5 specific characters, it is less clear whether a "body" section is required. As for the WAV file, it is missing at least one but possibly two vital chunks. The WAV specification requires that each WAV file contain a format and data chunk: "<fmt-ck> must always occur before <wave-data>, and both of these chunks are mandatory in a WAVE file" (IBM Corporation and Microsoft Corporation, 1991, [56]). Although the presence of the string "fmt" in the header *might* constitute the beginning of the <fmt-ck> chunk, it clearly does not have a <wave-data> chunk. This file does not meet the minimum requirements.

The positive identification of a file's type – suggested by the results from testing the manipulated files – is a low bar. Indeed, it is so low as to cast doubt on all positive identifications while spotlighting the scale of the issue when considering that current tools might report a "positive" identification for less than 60% of the material in a given collection. These tools do not generally look past the first few bytes of a file before making an identification determination. And, although that successful "identification" may be adequate for some purposes – as a first pass in data collection or for repository analysis, for example – it hardly constitutes "validation." On this last point, it is particularly bothersome that the MET tool returned such confident identifications as it is ostensibly designed to more robustly analyze a given file. As for JHove's validation and characterization capabilities, not only did it fail to correctly identify the contrived files, but it also returned an equally incorrect identification, versus a more neutral "unknown."

Tool Analysis

Both JHove and MET, by virtue of positively identifying a file (but not as a generic "bytestream"), presumably effectively validate the file. For validation, tools must look beyond the file extension and magic numbers; they must use a matrix of data points, all of which may be dispersed throughout the file. In this way, identification becomes a by-product of validation, if not synonymous with it. JHove and MET reach their determinations by looking at external and internal file markers, including byte-level patterns such as magic numbers. Beyond superficial identification based purely on magic numbers, as seen here, it would be difficult to mimic a file's (many) internal markers and have that file actually be a different file type than what it purports to

be internally, though not impossible. Examining JHove's code base, JHove does appear to rely on its own logic to make file identification (and validation) determinations. But, after reviewing MET's code base, an act sparked by the frequent reporting of mimetypes MET was not known to support, MET appears to rely on a third-party library to determine a file's mimetype.[21] The third-party code determines mimetype solely on the file's extension.[22] This also underscores the importance of these projects being open-source. Without being able to examine the code base for these projects, it is impossible to sufficiently audit how these tools come to the conclusions they do. In the case of MET, mimetype is based on a rather tenuous data point. This transparency is essential to the preservation process.

Validation is a high bar, and difficult to achieve. For JHove and MET to try and provide this level of identification and validation service for a given file type, each type receives its own module, which requires extensive programming. Unless a module exists for a particular file type, it is simply impossible for JHove and MET to identify and validate a file of that type. Therefore, when these tools fail to identify a file type and that file type does not have a corresponding module, it is not a revelatory result. Correctly identifying the file (and validating it) also greatly depends on how strictly (or not) the module is programmed to evaluate the file type according to the file type specification. Some file types do not need to conform perfectly to their specification in order to be well-formed and function files. Although one might reasonably argue that such files are not "valid," this might be a distinction that is detectable only when comparing a file to its specification. Many programs will open and provide access to them as if they were conformant. Also, if a module looks strictly for an internal marker where the file specification permits variation, the identification may fail not because the file does not conform to its specification, but because the module was not programmed to allow for the variation. On the other hand, because of JHove's and MET's architectures, when they are able to make a positive identification, they also provide some form of characterization information about the file. The information captured depends on the module programming. Therefore, if the characterization information is insufficient to the individual or organization using the tool, it falls to the user to either write a new module or modify (and recompile) the existing modules – this is a non-trivial undertaking. These are considerable shortcomings of this approach, and one that

21 http://meta-extractor.cvs.sourceforge.net/viewvc/meta-extractor/metadata-extractor/src/java/nz/govt/natlib/xsl/XSLTFunctions.java?revision=1.1&view=markup

22 http://www.docjar.com/docs/api/sun/net/www/MimeTable.html#findByFileName%28String%29

only very robust module development can mitigate.

The form and level of information in the output from the tools varied from tool to tool, as did the semantics employed. In the absence of some standard for this type of information, this is to be expected. But, for repository managers seeking to employ more than one tool as a type of check on the others, the lack of uniformity in the results complicates comparison of the results. FITS, mentioned above, exists in part to address this problem.

The tool results, therefore, needed to be normalized for comparison purposes. For example, warning information was not normalized across all three tools, nor did the tools report a type of confidence measure for the identification. Regarding an identification confidence measure, DROID conveniently assigned a measure – either positive, tentative, or not identified – but such a basic metric had to be developed for the other tools based on their output. As for how DROID made such a determination, and the classification matrix designed for the other tools, this is covered in depth in Appendix B. As for warnings, DROID reports warnings, if any, in its output. "Possible file extension mismatch" was the only warning reported by DROID. For both JHove and MET, warnings were basically extensions of the file identification determination and, like the identification confidence measure for JHove and MET (see Appendix B), what constituted a "warning" was determined during the normalization process. If a file was "Tentatively" identified or "Not identified," it received a warning. There could be a few additional warnings when using MET. One was "Program error," a determination made when the XML output was missing a File element.[23] The error that led to the XML output omitting a File element was observable during tool evaluation and recorded in the logs created during the evaluation. In a few instances, the XML output generated by MET was discovered to be malformed. These types of errors were reported as warnings.

A little more documentation was desired, therefore, describing the output of these tools. DROID is the exception, as the developers publish very detailed information about the tool's output format, particularly the semantics behind the element and attribute names in the XML (Brown, 2006). JHove and MET, however, do not provide any substantive documentation about their output. Such documentation would include detailed explanations of how the programs arrived at

23 http://www.3windmills.com/thesis/data/ia/met-1/AH-hpc_1-14-10/hpc_1-14-10.gif.xml

specific determinations and the definitions of some elements (DROID's documentation includes this information). Although most XML elements are named in such a way as to be self-explanatory, clear documentation detailing the XML semantics is lacking. For example, the meaning of JHove's "sigMatch" element is a little unclear, at least until one finds mention of it buried among the release notes beta 3 release of JHove 1 in 2005 (it is understood to mean "signature match" of a particular file format module) (http://hul.harvard.edu/jhove/releasenotes-1_0b3.html). Still, under what conditions, for example, might a file with an AVI extension have a "reportingModule" of "bytestream" but a "sigMatch" module of "WAV-hul" and a mimetype of "application/octet-stream"?[24] Why, for example, is the "reportingModule" different than the "sigMatch" module?

Data Analysis

Turning to the harvested files – the LC, IA, and Personal User files – as noted above, files created under controlled conditions were generally consistently and "positively" identified by the tools, based on matching file extension and magic numbers. But, it bears repeating: identification determinations have not been independently verified – one must take the tools' conclusions at face value. While this is not unreasonable, it does underscore the faith one must place in the results reported by these tools.

Nevertheless, inspection of the results from the three Library of Congress collections compared to the results from the Internet Archive files and Personal User Files supports the idea that files created under controlled conditions were generally consistently and "positively" identified. The LC files, a mixture of common files types produced in and by libraries during digitization projects, were consistently and correctly identified. Tables 2-6, which detail the percentages of "identified," "tentative," and "not identified" by collection and tool, provide high-level views of this phenomenon for each collection. Appendix B provides in-depth treatment of the three identification classifications; it also includes an identification matrix for reference purposes (Table 9).

24 http://www.3windmills.com/thesis/data/ia/jhove/10.1.10/10110mestica.avi.xml

Table 2:LC Chronicling America

LC Chronicling America

Tool	Positive	Tentative	Not Identified
Droid	500 (83%)	100 (17%)	0 (0%)
Jhove	600 (100%)	0 (0%)	0 (0%)
Jhove-with-s	600 (100%)	0 (0%)	0 (0%)
Met-1	499 (83%)	0 (0%)	101 (17%)
Met-2	399 (67%)	200 (33%)	1 (0%)
Local	600 (100%)	0 (0%)	0 (0%)

Table 3: LC Prints and Photographs

LC Prints and Photographs

Tool	Positive	Tentative	Not Identified
Droid	368 (94%)	17 (4%)	5 (1%)
Jhove	368 (94%)	0 (0%)	22 (6%)
Jhove-with-s	390 (100%)	0 (0%)	0 (0%)
Met-1	390 (100%)	0 (0%)	0 (0%)
Met-2	368 (94%)	22 (6%)	0 (0%)
Local	390 (100%)	0 (0%)	0 (0%)

Table 4: LC Performing Arts Encyclopedia

LC Performing Arts Encyclopedia

Tool	Positive	Tentative	Not Identified
Droid	1292 (98%)	6 (0%)	26 (2%)
Jhove	1201 (91%)	0 (0%)	123 (9%)
Jhove-with-s	1324 (100%)	0 (0%)	0 (0%)
Met-1	1225 (93%)	0 (0%)	99 (7%)
Met-2	1183 (89%)	106 (8%)	35 (3%)

Table 5: Internet Archive

Internet Archive

Tool	Positive	Tentative	Not Identified
Droid	723 (64%)	319 (28%)	93 (8%)
Jhove	432 (38%)	0 (0%)	703 (62%)
Jhove-with-s	1135 (100%)	0 (0%)	0 (0%)
Met-1	484 (43%)	0 (0%)	651 (57%)
Met-2	421 (37%)	488 (43%)	231 (20%)
Local	1135 (100%)	0 (0%)	0 (0%)

Table 6: Personal User Files

Personal User Files

Tool	Positive	Tentative	Not Identified
Droid	3915 (64%)	860 (14%)	1333 (22%)
Jhove	3472 (57%)	0 (0%)	2637 (43%)
Jhove-with-s	6105 (100%)	0 (0%)	4 (0%)
Met-1	3534 (58%)	0 (0%)	2577 (42%)
Met-2	2876 (47%)	3019 (49%)	221 (4%)
Local	6113 (100%)	0 (0%)	0 (0%)

Specific analysis of the results from each collection follow, beginning with the digital library collections from the Library of Congress.

LC Chronicling America (see Table 2 above)

The tools generally reported uniform results for the CA material with the exception of text files.

JHove (without the "s" option) identified all 600 files, though 100 files (all text files) were classified as "Tentative," a determination made because there was no "sigMatch" element identifying a matching module despite the fact that: 1) the "reporting module" used was UTF8-hul, 2) the file's format was correctly identified as "UTF-8," and 3) the file was considered to be "well-formed and valid." This may be as expected for this file type, but without clear documentation as to the significance of the "sigMatch" property (versus the reportingModule property), it remains unclear whether to trust the determination fully.[25] The characterization information captured by JHove recorded the various types of Unicode blocks (such as the presence of Basic Latin characters, characters from the Latin-1 set, and geometric shapes), the number of characters in the document, and the types of line endings. Although seemingly innocuous, all of these attributes could be essential in the future to properly understanding one of these files.[26]

25 It may be that, because there is no reliable way to identify a text file confidently, i.e. with some form of signature match, the sigMatch property is not included as part of the output. Again, only documentation can definitively answer this.

26 Consider, for example, the frequency one can encounter a problem with a text file created in a Microsoft Windows environment when accessed in a Unix/Linux environment - if it is a batch file it may not run because of the presence of Windows line endings and if it is a generic text file, the Unix/Linux system may interpret every line break as two lines.

JHove with the "S" option (meaning JHove attempts to identify the file based purely on internal markers) returned similar results, with differences in a few key areas. Unlike JHove without S option, JHove-with-S-option positively identified all 600 CA files and reported no warnings. And, indeed, inspecting the JHove-with-S-option output of the problematic text files mentioned above reveals that JHove-with-S-option output reported a "sigMatch" for those text files. Closer inspection, however, shows the results were inconsistent between the two ways to operate JHove. Output from one of the JHove-with-S-option files shows it to be a well-formed ASCII file while the output from JHove-without-S-option reports the file to be a "well-formed and valid" UTF-8 file.[27] Given that Unicode folds in ASCII this might be interpreted as saying the same thing in two different ways, but it is still an unanticipated inconsistency.

The Metadata Extraction Tool provided the expected results. It was able to recognize the PDF file, JPEG image, XML files, and the text files. It was unable to identify the JPEG 2000 files. The Local tool, largely dependent on the file() command line program, accurately identified all the file types except for the text files. The Local tool correctly reported "text/plain" for those files, but also incorrectly reported surprising mimetypes such as "text/troff," "text/x-c," "text/x-fortran," "text/lisp," and "text/pascal."[28]

LC Prints and Photographs (see Table 3 above)

Files from the Library of Congress's Prints and Photographs Online Catalog (PPOC) were all images. No metadata records were captured. Like CA, a cursory review of the results from the tools reveals a general consistency in the results. Of the 390 files evaluated, JHove, with and without the S option, positively identified 368 files. The remaining 22 files were zero-length files, meaning they were empty files, and JHove identified them all as "well-formed and valid" bytestreams (though there was no byte in them).[29] These were reported to have a mimetype of

27 For JHove: http://www.3windmills.com/thesis/data/ca/jhove/lccn_sn83030193_1908-11-06_ed-1_seq-9/ocr.txt.xml
For JHove-with-s-option: http://www.3windmills.com/thesis/data/ca/jhove-with-S-option/lccn_sn83030193_1908-11-06_ed-1_seq-9/ocr.txt.xml

28 http://www.3windmills.com/thesis/ca/local/extension/?value=txt

29 The 22 zero-length files are an anomaly from the original capture. This was a programming oversight from the capture process, which presumed the existence of certain files, such as a second image representing the verso of a baseball card. These zero-length files were nonetheless analyzed, as all files were, by the tools and create

"application/octet-stream." However, there were 23 files with a mimetype of "application/octet-stream."[30] JHove (without the S option) reported the twenty-third file to be a TIFF image with a size of 573938 bytes.[31] It has a "reportingModule" of BYTESTREAM but JHove was able to provide a "sigMatch" of TIFF-hul. This appears to be an error. JHove-with-S-option correctly identified the file to be a TIFF image, reporting a sigMatch with TIFF-hul and a mimetype of image/tiff.[32]

While JHove, with or without the S option, accurately identified the non-zero-length images, MET appears to begin identification of the zero-length files but stops short of positively identifying the file format. MET-1 goes so far as to report the correct mimetype (based on the file extension alone) but stops at the point where normally MET-1 would report a FileFormat.[33] It is possible to determine whether MET-2 positively identified a file only by the presence of additional elements in the output other than "METADATA." This is a little problematic when considering a non-zero-length (and positively identified) TIFF image and a zero-length TIFF image. The non-zero-length file includes additional elements in the output, such as HEADER and IMAGEFILEDIRECTORY and ELEMENT, but the only explicit hint of the file's format is its reported TYPE, which records the file's mimetype (image/tiff).[34] The same TYPE is reported even for a zero-length TIFF image.[35] The same behavior is also seen with GIF files.[36]

Analysis of the DROID results demonstrates the same consistency in output compared to the other tools with regard to the 368 positively identified PPOC files, but the DROID tool had

small anomalies in the data.

30 http://www.3windmills.com/thesis/ppoc/jhove/mimetype/?value=application/octet-stream

31 http://www.3windmills.com/thesis/data/ppoc/jhove/wp2003001055_PP/4a40940u.tif.xml

32 http://www.3windmills.com/thesis/data/ppoc/jhove-with-S-option/wp2003001055_PP/4a40940u.tif.xml

33 This is most clear when comparing MET-1 output. Starting with a positively identified JPEG image (http://www.3windmills.com/thesis/data/ppoc/met-1/007680722/0067fr.jpg.xml), MET-1 continues to output information after recording the "Mimetype." In the preceding instance, MET-1 continues to output information about the FileFormat (JPEG) and specific characterization information about the given image. Viewing the MET-1 output after analyzing a zero-length JPEG image (http://www.3windmills.com/thesis/data/ppoc/met-1/007680723/0068br.jpg.xml), output stops after the "Mimetype" element. When looking at the MET-1 output of a file type the tool is not programmed to recognize, but given a file that is not zero-length, MET-1 output stops after reporting a mimetype of "file/unknown" (http://www.3windmills.com/thesis/data/ia/met-1/2010-01-17__059-03__Core__Element_Two/2010-01-17__059-03__CORE__ELEMENT_TWO__video.flv.xml).

34 http://www.3windmills.com/thesis/data/ppoc/met-2/007678537/0005bu.tif.xml

35 http://www.3windmills.com/thesis/data/ppoc/met-2/007680727/0072bu.tif.xml

36 The output for a non-zero-length GIF: http://www.3windmills.com/thesis/data/ppoc/met-2/007678537/0005ft.gif.xml. The output for a zero-length GIF: http://www.3windmills.com/thesis/data/ppoc/met-2/007680727/0072bt.gif.xml.

slightly more difficulty with the remaining 22 zero-length files.[37] DROID tentatively identified 17 of the remaining 22 files and failed to identify the other 5. The tool reported a "zero-length file" warning for each. All of the 5 unidentified files were GIF images.[38] The 17 tentatively identified files were a mixture of TIFF images and JPEG images. Based on the output of the tentative hits, DROID provides all possible file matches because its "tentative" classification is based on the file's extension and it can be no more specific. Each TIFF output includes a tentative identification for a DNG image, TIFF/IT image, TIFF/EP image, GeoTIFF image, and TIFF-FX image.[39] Conversely, the results for the GIF files contained no FileFormatHit element at all, meaning that DROID did not even make a guess.[40] Why DROID would provide what appears to be all possible matches for the zero-length TIFF files but not one for the zero-length GIF files is unclear. DROID provides four FileFormatHit sections for each positively identified TIFF image, one each for versions 3, 4, 5, and 6 of the TIFF specification, because it was unable to determine to which TIFF version the files conform.

Finally, the Local tool positively (and correctly) identified all 390 files. The 22 zero-length files were given mimetypes of "application/x-empty."[41]

LC Performing Arts Encyclopedia

JHove (with and without the S option) performed as expected and as demonstrated with the other LC collections when run against the 1,324 files from the Library of Congress's Performing Arts Encyclopedia (see Table 4 above). JHove (without the S option) was unable to identify 123 files, of which at least twelve were zero-length files.[42] The remaining 111 files, based on their file extensions (mostly MP3, but a few MPG and one MP4), are file formats JHove is not programmed to recognize. JHove did, however, positively identify 57 ASCII text files.[43] These were RealMedia Metafiles (extension .ram). While this is technically correct (RealMedia MetaFiles are simple text files containing links to streaming media), the identification should be considered insufficient for digital preservation purposes. A RealMedia MetaFile may take the form of an ASCII text file, but its intent and information are purposeful enough to characterize it

37 http://www.3windmills.com/thesis/ppoc/droid/counts
38 http://www.3windmills.com/thesis/ppoc/droid/identifications/?value=Not_Identified
39 http://www.3windmills.com/thesis/data/ppoc/droid/007680724/0069bu.tif.xml
40 http://www.3windmills.com/thesis/data/ppoc/droid/007680725/0070bt.gif.xml
41 http://www.3windmills.com/thesis/ppoc/comparisons?sort=local
42 http://www.3windmills.com/thesis/ihas/jhove/identifications/?value=Not_Identified
43 http://www.3windmills.com/thesis/ihas/jhove/mimetype/?value=text/plain

more precisely. RealMedia MetaFiles contain semantics that distinguish it from other types. The PAE sample also contains 13 files that, based on their file extension, purport to be XHTML files. JHove's HTML-hul modules supports the identification of XHTML, but all of these files reported a sigMatch and reportingModule of XML-hul. Visual inspection shows these files to be HTML files, but they are not valid XHTML files (among many other reasons, they do not have a DOCTYPE declaration). They contain valid HTML mark-up and they are well-formed in terms of XML, hence their identification as such.

Unlike JHove without the S option, JHove-with-S-option identified all 1,324 files. But, based on reported mimetypes, JHove-with-S-option positively identified 91 "text/plain" files and 89 "application/octet-stream."[44] The "application/octet-stream" files represent zero-length and multi-byte-length file types not recognized by the JHove tool.[45] On the other hand, the files identified as "text/plain" by JHove-with-S-option deserve further scrutiny.[46] The majority of these files are the same RealMedia MetaFiles identified earlier, but more than 30 files in the list are MP3 files. Not one is a zero-length file and, although all the files were not tested, those that were are playable and produce audio. In short, they are not well-formed ASCII text files, and all evidence points to them being audio files. JHove-with-S-option treated the XHTML files the same as JHove without the S option.

MET-1 was unable to identify 99 of the 1,324 files.[47] Although 64 of the unidentified files are RealMedia Metafiles,[48] which MET-1 is not programmed to recognize, 26 of the files are MP3 files,[49] which MET-1 *is* programmed to recognize; the remaining 9 files are zero-length files. It may be that the unidentified MP3 files are not well-formed and valid, but that would need to be

44 http://www.3windmills.com/thesis/ihas/jhove-with-S-option/counts
45 http://www.3windmills.com/thesis/ihas/jhove-with-S-option/mimetype/?value=application/octet-stream
46 http://www.3windmills.com/thesis/ihas/jhove-with-S-option/mimetype/?value=text/plain
47 http://www.3windmills.com/thesis/ihas/met-1/counts
48 http://www.3windmills.com/thesis/ihas/met-1/identifications/?value=Not_Identified
49 This can be seen here http://www.3windmills.com/thesis/ihas/met-1/extension/?value=mp3. Related, but not necessarily connected to the aforementioned issue, MET-1 sometimes does not record a mimetype. This was unexpected. For files MET-1 cannot identify, it seemed to faithfully report a mimetype of "file/unknown." This issue was noticed when evaluating the results and finding that one view did not reproduce information found in another view but which should have been viewable in both views. Because of some (presumed) tool evaluation error, MET-1 did not record a mimetype for these 30-odd MP3 files. Quite simply the expected element was not in the XML output and therefore was missed in the scripted count function. Something like this could be a much larger issue in the future. It is not beyond reason that a repository manager might want, or need, to analyze his holdings by crawling the technical metadata. *Not* including a mimetype in the output is inconsistent in this instance.

reconciled with the fact that a desktop audio player is capable of accessing and playing the file.[50] Interestingly, some of the MPG video files from PAE could be considered tentative matches by MET-1. Although detailed characterization information is missing from the MET-1 output when run against MPG video files, the third-party library MET uses to determine a file's mimetype, in this case, returned one that departs from those expected from MET. XHTML files identified by JHove as XML files were identified by MET-1 as HTML.[51]

MET-2 positively identified 1,183 files (meaning the tool provided some characterization information); tentatively identified 106 files (meaning it reported some kind of mimetype if even it was "file/unknown"); and did not identify 35 files (these were zero-length files).[52] Most of the tentatively identified files were RealMedia Metafiles and MPG files – file types MET is not programmed to recognize); a few were zero-length JPG files.[53] What is not understood, and considered at this time to be an inconsistency of the tool, is why MET-2 would report a mimetype for a zero-length JPG file but not a mimetype for a zero-length MP3 file.[54]

DROID positively identified 1,292 files, tentatively identified only 6 files, and failed to identify 22 files.[55] The 6 tentative files were either JPEG or TIFF images, based on their file extensions, and they were all zero-length files. The 22 unidentified files were either MP3 or MPG files according to their file extensions (providing a file had an extension, there were 6 without), and all of these too were zero-length files. Notably, DROID positively identified the RealMedia Metafiles (even reporting a mimetype of "audio/vnd.rn-realaudio, audio/x-pn-realaudio").[56] PAE also contains one MP4 file. Interestingly, DROID positively identified the file, but did not report

50 http://www.digitalpreservation.gov/formats/fdd/fdd000012.shtml Although this paper is not focusing on characterization, a few words are appropriate here about the characterization information produced by the MET tool. Like JHove (without the S option), MET-1 tried to provide file characterization information though, unlike JHove, MET-1 outputs considerably less information than JHove. One quibble about MET-1 is its use of terminology when characterizing an audio file. For both MP3 and WAV audio file types, MET-1 records the file's "Resolution" and "BitRate," but by "Resolution" MET-1 means "Bit-depth" and by "BitRate" MET-1 means "Sample Rate." This is a small point, and easy to correct for, but the importance of clarity in file format identification and characterization feeds into the important role of documentation in digital preservation.

51 http://www.3windmills.com/thesis/data/ihas/met-1/loc.natlib.ihas.200035636/ihas_songofamerica_collection_200035636_200035636.xhtml.xml

52 http://www.3windmills.com/thesis/ihas/met-2/counts

53 http://www.3windmills.com/thesis/ihas/met-2/identifications/?value=Tentative

54 Because MET uses modules to characterize files, this may be the result of module design inconsistency,

55 http://www.3windmills.com/thesis/ihas/droid/counts

56 http://www.3windmills.com/thesis/ihas/droid/mimetype/?value=audio/vnd.rn-realaudio,%20audio/x-pn-realaudio

a mimetype for it.[57] DROID also reported 12 "possible file extension mismatch" warnings, all for the files with an XHTML file extension.[58]

Finally, the Local tool identified all PAE files, including the MP4 file, complete with mimetype. As in other tests, if the file was zero-length, the Local tool reported a mimetype of "application/x-empty."[59] Twenty-nine MP3 files were classified with a mimetype of "application/octet-stream."[60] These are all playable MP3 files. Clearly, the Library of Congress created some MP3 files that MET and the Local tools have difficulty identifying. This could be one of those instances where the identification/validation procedure is too strict, or it could be an instance of a forgiving software application.

Internet Archive

Turning to the Internet Archive files (see Table 5 above), a similar pattern to that established above continues: if a tool is designed to accommodate a particular file type, then that file is generally positively identified and characterized (depending on the tool). Therefore, it is not necessary to treat each tool separately but draw attention to areas of significant consistency and discrepancy between the tools.

In many instances, the results are quite consistent across the tools.[61] For example, based on identified mimetypes, DROID and the Local tool identified 43 ZIP archive files while MET-1 and MET-2 identified 42 ZIP archive files. Every tool identified 264 XML files, though MET-1, MET-2, and the Local tool use the mimtype "application/xml" while the others use "text/xml." Five files with a mimetype of "video/x-msvideo" were reported by DROID and the Local tool, while MET-1 and MET-2 reported five files with a mimetype of "application/x-troff-msvideo" - all the just-listed tools are identifying the same file. All four tools identified 4 PDF files. Naturally, JHove results have a high number of reported "application/octet-stream" mimetypes (JHove: 722; JHove-with-S-option: 507) and MET tools have a similarly high number of

57 http://www.3windmills.com/thesis/data/ihas/droid/loc.natlib.ihas.200155985/natlib.loc.gov_natlib_ihas_warehouse_coptic_200155985_seg01_0001.mp4.xml
58 http://www.3windmills.com/thesis/ihas/droid/warning/?value=Possible%20file%20extension%20mismatch
59 http://www.3windmills.com/thesis/ihas/local/mimetype/?value=application/x-empty
60 http://www.3windmills.com/thesis/ihas/local/mimetype/?value=application/octet-stream
61 http://www.3windmills.com/thesis/ia/comparisons

reported "file/unknown" mimetypes (425 for both).[62] Perhaps a little more bothersome is how often DROID, MET-1, and MET-2 failed to report a mimetype altogether: 387, 226, and 221 instances respectively. For example, some of the missing mimetypes would account for the 40 "audio/x-flac" files identified by the Local tool but not by DROID. This is a recurring pattern and one worth further comment.

The Local tool, which is basically the Unix *file* command, was able to *identify* notably more than the number of file types than even its closest competitor, DROID. Beyond providing zero-length files with an informative mimetype (versus not giving it one), the Local tool was able to identify these mimetypes where the other three tools did not report a hit or equivalent: application/ogg, application/vnd.rn-realmedia (this is a little surprising given DROID's accurate identification of the RealMedia Metafiles that are part of PAE), audio/mp4, audio/x-flac, video/mp4, and video/x-ms-asf. When the Local tool is placed against the other tools, the difference is greater. Of course, all of this is stated with the caveat that JHove and MET do much more than simply *identify* a file.

Digging deeper, there are difficult-to-explain discrepancies. For example, DROID and JHove (without the S option) identified only 39 JPEG files ("image/jpeg") while JHove-with-S-option, MET-1, MET-2, and the Local tool identified 52 JPEG files. An even greater discrepancy in results exist when looking at "text/plain" files:

Table 7: Number of text/plain mimetypes reported by the tools for the IA material

Mime	DROID	JHove	JHove-With-S	Met-1	Met-2	Local
text/plain	5	79	281	5	5	79

The only files identified by DROID, MET-1, and MET-2 with a mimetype of "text/plain" had *.txt file extensions. JHove and the Local tool identified files with a *.txt file extension as "text/plain" but also files with *.md5 and *.m3u file extensions. This is technically correct: the former files are text files containing simple hashes for IA sample files and the latter are text-formatted audio playlist files. But, as with the RealMedia MetaFiles from the PAE sample, these files contain semantics that would permit a more meaningful identification. DROID considered

62 http://www.3windmills.com/thesis/ia/comparisons

the m3u files “audio/mpeg” files while MET-1 and MET-2 reported “file/unknown.” Strangely, JHove-with-S-option identified all files with an MP3 file extension as “text/plain,” which explains why JHove-with-S-option reported such a large number of “text/plain” files.

Sometimes what may at first appear to be a discrepancy is not, but instead represents an instance where the four tools use variant mimetype formats. For example, MET-1 and MET-2 use “audio/wav” and “audio/x-wav” for WAV files, JHove uses “audio/x-wave,” and DROID and the Local tool appear to use “audio/x-wav” exclusively. This was also seen with the mimetype for XML files. These are not insurmountable obstacles, but a lack of uniformity (even with the same tool!) adds an unnecessary layer of complication in the same vein as not including a mimetype in the output. MET-1 and MET-2 were unable to handle the IA's GIF files. These are animated GIFs and, in fact, caused a program error whenever MET encountered one.

Personal User Files

The more than 6,000 Personal User Files (PUF) (see Table 6 above) – a wide sample of real-life files - introduced an unpredictable element not present in the other samples. PUF contains 122 unique file extensions, suggesting 122 distinct file types. In comparison, the IA collection of files, the second most varied assemblage, contained 33 unique file extensions. But, like the IA collection, the end results are similar and tightly tied to the capabilities of the various identification, validation, and characterization tools. JHove (without the S option) was unable to identify 2,762 files (45%), reporting an application/octet-stream mimetype for each one. JHove-with-S-option reported 2,583 (42%) application/octet-stream mimetypes. MET-1 and MET-2 reported 2,365 (39%) file/unknown mimetypes. DROID was unable to provide a mimetype for 2,110 (35%) files. The Local tool reported a mimetype of application-octet stream (i.e. unidentified) for 1,239 files (20%), a success rate more than twice as good as JHove and 15 points better than DROID.

And, like the IA sample, there are mimetype designation inconsistencies and a few surprises. One surprise was from the MET-1 and MET-2 tools. Each identified 8 files with a mimetype of application/x-shar, which is the commonly-accepted mimetype for Unix-based operating system Shell Archive files (http://en.wikipedia.org/wiki/List_of_archive_formats) and the identification

of which MET does not seem to support. This is likely the result of MET's third-party library matching the file extension to a known mimetype. The Local tool correctly identified the file, reporting a mimetype of "text/x-shellscript." JHove and DROID were unable to identify the file.

The Local tool appears largely incapable of identifying video file types. MOV (Quicktime files) or MPG (MPEG files) either did not receive a mimetype designation at all or they were designated "application/octet-stream." DROID failed to report a mimetype for any file with a MOV extension; JHove reported the same as the Local tool. MET-1 and MET-2, again surprisingly because it is undocumented, reported "video/quicktime" for a mimetype for these files. The MET-1 and MET-2 also identified other video types, such as MPG, with which the other tools struggled (DROID has some support for MPEG files).

Archive files, those commonly created using Zip compression (http://en.wikipedia.org/wiki/ZIP_%28file_format%29) are problematic for all the tools. These are becoming common: Java JAR files are Zip compressed; OpenOffice files are Zip compressed; MS Office 2007 files are Zip compressed. DROID identified 388 files with a mimetype of "application/zip." The Local tool identified 204. MET-1 and MET-2 each found 22 Zip archives. Although a fair number of the DROID identified "application/zip" files were in fact Zip archives, the vast majority were OpenOffice documents.[63]

All 22 of the files identified by the MET tools were in fact Zip archives, but delving a little deeper into the results reveals that the MET tools found 59 files with a .zip file extension and categorized the majority of those as "application/open-office-1.x," an unexpected mimetype because of the .zip file extension.[64] Indeed, the tools had similar difficulty with OpenOffice file formats. The majority of these files have "odt," "odp," or "ods" as their file extensions. The MET tool, with modules for OpenOffice document types, positively identified these file types. The Local tool was able to identify those with an "odt" (Open Document Text file) extension,[65] but reported "application/octet-stream" for the other OpenOffice file types.[66] DROID identified

63 http://www.3windmills.com/thesis/userfiles/droid/mimetype/?value=application/zip
64 http://www.3windmills.com/thesis/userfiles/met-1/extension/?value=zip, http://www.3windmills.com/thesis/userfiles/met-2/extension/?value=zip
65 http://www.3windmills.com/thesis/userfiles/local/extension/?value=odt
66 http://www.3windmills.com/thesis/userfiles/local/extension/?value=ods, http://www.3windmills.com/thesis/userfiles/local/extension/?value=odp

all OpenOffice files as "application/zip."[67] As for Java JAR files, DROID appears to have some ability to identify them correctly, but still identified some JAR files as "application/zip" types.[68] The Local tool identified all JAR files as "application/zip."

A good many of the Personal User Files consist of computer code files: Java files, Actionscript, Javascript, PHP, Perl, Python, C# .Net, FLA and even *nix system shared library files (these have .so file extension). DROID positively identified files with a "js" file extension, presumed to be a file containing Javascript code, as "application/javascript."[69] JHove determined them all to be "text/plain," which, while not technically incorrect, isn't exactly complete.[70] MET reported "file/unknown."[71] The Local tool had slightly more difficulty: it found most to be of "text/plain" types but quite a few to be "text/x-c," "text/x-c++," and "text-pascal."[72] These kinds of inconsistencies continue with these types of files.

1) DROID found most PHP files to be "text/html."[73] DROID did not report a mimetype for *.pl (Perl), *.as (Actionscript), *.py (Python), *.cs (C#), *.fla (Macromedia/Adobe Flash files).
2) JHOVE reported most of the same files to be "text/plain," except *.fla and *.so, which the tool reported to be "application/octet-stream" (they are binary files)
3) MET-1 and MET-2 reported *.pl files to be "text/plain." All others were "file/unknown."
4) Local was a hodgepodge. Some of the *.pl files were correctly identified as "text/x-perl," while the remaining were classified as "text/plain." Py files were considered text/x-pascal. The Local tool identified most *.cs files as "text/x-c++" (the few remaining were "text/plain"). The FLA files all generated an error.[74] The *.so files were correctly identified as "application/x-sharedlib."

67 http://www.3windmills.com/thesis/userfiles/droid/extension/?value=odt, http://www.3windmills.com/thesis/userfiles/droid/extension/?value=ods, http://www.3windmills.com/thesis/userfiles/droid/extension/?value=odp

68 http://www.3windmills.com/thesis/userfiles/droid/mimetype/?value=application/java-archive Do note DROID identified a few files with ZIP extensions as JAR files. DROID also identified two JAR files as "application/zip" types, see http://www.3windmills.com/thesis/userfiles/droid/mimetype/?value=application/zip. There is also one WAR file, which stands for Web Application Archive and is a type of Java archive file, identified as "application/java-archive.". For links to the file format's XSD and Sun's file format documentation see the general Wikipedia entry for WAR files: http://en.wikipedia.org/wiki/WAR_%28Sun_file_format%29

69 http://www.3windmills.com/thesis/userfiles/droid/extension/?value=js

70 http://www.3windmills.com/thesis/userfiles/jhove/extension/?value=js

71 http://www.3windmills.com/thesis/userfiles/met-1/extension/?value=js, http://www.3windmills.com/thesis/userfiles/met-2/extension/?value=js

72 http://www.3windmills.com/thesis/userfiles/local/extension/?value=js

73 http://www.3windmills.com/thesis/userfiles/droid/extension/?value=php

74 http://www.3windmills.com/thesis/userfiles/local/extension/?value=fla

Limitations

MET and JHove are programmed to identify, validate, and produce characterization information for a select number of mimetypes and they manage to cover a very significant number of files these tools will need to analyze. This is in no way controlled for in this study. Simply put, it is assumed that TIFF images, JPEG images, MP3 audio files, WAV audio files, PDF files, XML files and common office documents are significantly more prevalent than most other types of files in digital library collections and in general use. Therefore, one is likely to encounter a file of the type JHove and MET are designed to handle. This issue complicates the discussion, because, when operated against a collection of files, 98 of which are JPEG images and 2 of which are Java Class files, the tools would report a 98% successful identification rate because JHove and MET are designed to recognize a JPEG file but not a Java Class file.[75] Another way to view this statistic, would be to evaluate only operative file formats, not the number of files analyzed, at which time one could conclude that JHove and MET reported only a 50% positive file identification rate.

75 If considering only the operative file formats, not the number of files analyzed, one might conclude that JHove and MET reported only a 50% positive file identification rate.

CHAPTER 6

DISCUSSION

As suggested at above, files created for use within a digital library and particularly those created under controlled conditions were more frequently positively identified with greater consistency and more frequently by all the tools under investigation. When these tools must analyze files created in uncertain circumstances, if even the circumstances of their creation are known, the positive identification rate drops precipitously. DROID, MET, and JHove all have a positive identification rate above 80% for the three Library of Congress collections; for two, the tools reported a positive identification rate above 90%. DROID, JHove, and MET only reported, at best, a positive identification rate of 64% for the Internet Archive and Personal User Files collections.

At no time should a file's extension form any basis for identification, and certainly not validation. It is easily changed by a user or computer program and, as such, should be treated with suspicion. A file extension is a convenience, not a requisite component, and to use it as a factor in identification places unearned trust in the file's creator. DROID handles this decently; the software should only return a “positive” identification if it matched internal characteristics. “Tentative” identifications are reported when DROID can match the file extension to a known file type, which is not the same as *identifying* a file's type. But, JHove and MET, which do not assign a confidence measure to identification, report information based on this weak data point. MET assigns a mimetype based on file extension alone. In probably more than 99% of cases the extension will align with the file's true type (based on an assessment of its internal markers), but it does not convey information about a meaningful identification.

The tools under discussion all “positively” identified the otherwise empty and invalid files created especially for this project and containing only a correct magic number sequence. The tool developers should consider matching more internal characteristics than just a file's magic number. This is not to suggest that identification can only be determined based on matching *all*

of a file's expected characteristics (though that would be robust identification indeed), but that files of specific types tend to have many more internal characteristics than just a distinct magic number sequence. Again, the tools have some support for this (DROID makes a distinction between a generic and specific identification), but more work could be done. In this way, matching a magic number alone becomes a "tentative" identification; "positive" identification is reserved when additional internal characteristics are matched. This would significantly raise the confidence one places in the identifications made by the tools.

The tools appeared to rely heavily on mimetype information to communicate file type, but mimetypes do not unambiguously identify files, at least not when a repository manager may need to know which *versions* of given file types. When identifications were accompanied by exactly one version number of a file type, then the two data points create a precise identification. JHove and MET functioned in this manner. Although this is a workable solution, moving toward a unique identification method would remove some of the ambiguity. PRONOM's PUID (Persistent Uniquer Identifier) can provide an excellent foundation for this. PRONOM issues a PUID for each variation of each file type. Therefore, although more than 6 varieties of TIFF files share the MIME type "image/tiff," each one of those variations has its own PUID allowing unambiguous identification. In addition to already providing a PUID,The National Archives UK has started to experiment with making the PRONOM data available as Linked Data to foster reuse (The National Archives[, UK], 2011).[76]

Even if all files were identified uniquely, it will be necessary to develop more robust file identification algorithms with accompanying file characterization information. The tools often do not make a significant distinction between structural information (file extensions, magic numbers) and semantic information. This results in false-positives – files identified "positively" as one type based on structural aspects but which are, in fact, of another type when taking into account structural *and* semantic aspects of the files. For example, complex file types – these are often file types that wrap other files, such as ZIP archives, which structurally adhere to a specific file type but whose contents reveal the file to be of a more distinct type – often proved challenging, even though DROID and, to a lesser degree, the Local tool have some support for

76 Although the following is presented with the caveat that it is somewhat deductive, the UDFR effort will explore similar paths as PRONOM, see http://groups.google.com/group/digital-curation/browse_thread/thread/c06331e09727f47c.

complex file types. DROID, for example, reports an Open Document Text file to be a ZIP archive file. The Java JAR files encountered in the tests are another prime example of this issue. Structurally, these identifications are correct, but, for a valid Open Document Text file, for example, the ZIP archive wraps a number of distinctly named directories and files. It is this specific internal organization that distinguishes an Open Document Text file from it being a quotidian ZIP archive file. DROID does not identify this file type correctly, though the software appears capable of examining inside complex file types to some degree, such as some video file formats, and identifying that there are multiple files present (and audio and video stream). DROID reports this information in its output. JHove and MET seem to have little or no support for identifying complex files types (MET can identified Open Text Documents). In fact, JHove's lack of ability to handle this issue is being addressed in JHove2, which will be capable of handling complex file format types (the example used by the JHove creators was of images with embedded metadata or other images) (Abrams 2007). Ultimately, specific file types (e.g., an OpenOffice document), with defined mimetypes (application/vnd.oasis.opendocument.text),[77] that are more broadly of another type (ZIP file) will be an area that needs considerable refinement. Looking ahead, repository managers need to ensure they have the required documentation, the full Representation Information (such as the OpenOffice Document file specification or the Java JAR file specification *and* the ZIP file specification) to ensure future access, per the OAIS Reference Model.

Similarly, it will take considerable programming to correctly identify some simpler file types – those that do not employ some type of wrapper file type. The accurate identification of PHP or Python code files, for example, though they are not "complex" objects, is a manifestation of the same issue as with complex file types. Structurally they are of one type, but when considering structural form *and* semantic aspects, the file is of a distinct type. For example, when the tools identify them as text files, they are technically correct. But there are semantic patterns to these files (a PHP file must begin, for example, with "<?" or "<?php" or "#!/usr/bin/php") that would permit a more accurate identification. This might help to inform a data curation specialist charged with preserving code used in a federally-funded experiment that she must also save a copy of the PHP (or Python) specification, or even the PHP source itself. This would apply to a

77 http://www.iana.org/assignments/media-types/application/vnd.oasis.opendocument.text

basic SQL dump file also.[78] These more complicated identification issues – specifically, the difficulties the tools have with them – not only represent significant challenges to the premise of file identification, but also demonstrate current deficiencies in confident and accurate identification.

That JHove2 will integrate DROID, and separate identification from validation are significant improvements, and that the development team is exploring leveraging the file() command's magic number database for identification purposes are essential activities. The present, stable release of JHove can only identify 12 file types (including the vague BYTESTREAM type); MET can handle at least 18 different file types. Each, however, when given a file they are designed to identify outputs many more details about the file (but not necessarily complete and full Representation Information). JHove and MET manage this detail by requiring a custom-programmed module for every file type. This modular software design is laudable: third parties could create identification, validation, and characterization modules that they could then share with community. Unfortunately, few developers created and contributed JHove modules back to the community. Although the module development concept will remain in JHove2 – hopefully with better and easier support for the creation and publication of community-created modules (Abrams 2007) – history suggests that community involvement in module development will be low. Much will rely on how easy and intuitive it is to develop a JHove module for JHove2, and the support given to these endeavors by developers, managers, and funders. And, it bears repeating, identification is only part of the problem; validation and characterization are equally, if not *more* vital for long-term preservation and the development of needed tools for these activities has been slow.

78 The quantification of these false positives, especially across all the tools, would be another study unto itself (this was particular problematic with the Local tool, which never failed to identify a file positively).

CHAPTER 7

CONCLUSION

The roles of file identification, validation, and characterization in digital preservation are not without their detractors, though much of that discussion has happened outside of the traditional scholarly venues. David H. S. Rosenthal, who works on the LOCKSS project from the Standford University library, posted a substantial blog article in which he questioned how much focus should be placed on ensuring that files conform, stringently, to their specifications (Rosenthal, 2009). Rosenthal noted that the conformance of HTML pages, PDF documents, and the like, to their specifications was deplorable. But, he also noted that browsers and PDF Readers (specifically Adobe's) almost always succeeded in rendering the document. He cited JHove's strict evaluation of PDF files as an example of this problem (this is similar to the issue that Littman (2006) encountered, but with TIFF image files). In the present study, like Rosenthal's PDF files and Littman's TIFF files, perfectly accessible MP3 files were considered invalid by the tools under investigation.

Ultimately, Rosenthal concludes that there is little to be done. He believes archives will accept files whether they conform or not ("And how would an archive reject non-conforming files? By returning them to the submittor [sic] with a request to fix the problem?") and, unless the tool is perfect, there will likely be false positives (a file doesn't conform but the tool thinks it does) and false negatives (it doesn't conform, but it can still be opened and rendered). Indeed, given the sheer volume of digital material to be preserved, non-conforming but accessible files will continue to be an issue not only to repository managers but also the creators of file identification, validation, and characterization tools, making such endeavors almost feel like a best-effort undertaking. Thoughtful responses expressed acceptance of Rosenthal's basic idea, but not as an absolute, arguing that digital preservationists have a little more agency in guiding what types of files a repository accepts or manages, and hence a usefulness for these tools yet.

To echo those skeptical of these strategies, such as Rosenthal, it is difficult to determine the

value of these tools for the long-term preservation of digital material. This is not to suggest that all the information these tools provide is of little value; at issue is the time and energy required to create, test, and implement the software needed to manage the scope of digital content that will require preservation. On top of that, these tools require precision *and* flexibility. Large scale file identification – even if further internal checks were made – seems achievable; DROID is exceptional. But, it may be a Sisyphean effort to create and implement individual modules for each file type, such as required by JHove and MET. Research data and related materials (e.g., SQL, databases, very-specialized image formats, specialized computer programs that access specially generated data sets) are far different in format from traditional digital library material (e.g., digitized images, audio, video, text); if such tools were to succeed, curation professionals will need to contribute heavily to these types of tools, such as module development or more precise file identification algorithms, in order to manage the material they acquire in conformance with accepted digital preservation management standards.

At this time, however, the road ahead is long. Currently, identification output is somewhat superficial, evaluating mainly structural aspects of a file versus semantic aspects, when both are needed so that the identification is complement and rendered with greater confidence. This is a matter of trust, and one that goes beyond the reported results to the tools themselves, as evidenced by the changing identification metrics in the University of Southampton's Preserv2 EPrints toolkit study, during which two PDF files were later identified as HTML files after a DROID software update. On the opposite side of this problem is the ability to write a custom validation module for a groups of files, such as LC did for a collection of TIFF images, that effectively circumvents the “standard” validation module, causing the files to fail validation checks without the custom module and when using off-the-shelf validation software.

Unless all data to be preserved are migrated to preferred preservation formats or arrive in the preferred formats, data preservationists and curators will likely have to manage as best they can. In such a scenario, accurate and unambiguous identification plays a vital role. An invalid PDF (i.e. one not conforming to the PDF specification) does not necessarily mean it is not accessible, but it places a much greater burden on either 1) migrating the file so that it adheres perfectly to its specification or 2) capturing the precise software and hardware environment conducive to accessing the file. In such a scenario, accurate and unambiguous identification plays a vital role, with characterization being second.

Progress to date on the perceived need, value, and importance of file identification, validation, and characterization activities in digital preservation has been steady and methodical. It is safe to conclude that file specification knowledge will be imperative to accessing data objects in the future. It is generally accepted that specific knowledge about the particular characteristics of to-be-preserved files will also be essential, if not for access purposes then for validation and authenticity purposes. It is less clear, however, whether and to what degree files need to conform to their published standards and, more importantly, whether such a strict approach to these activities, especially file format validation, is sustainable and, moreover, achievable.

Nonetheless, developers should continue working on these tools – they serve general needs and they are excellent management tools, especially when interested in performing audits. But, data preservationists and curators must recognize the tools' limitations, especially those professionals who receive unfamiliar material for processing.

REFERENCES

Abrams, S. L. (2005). Establishing a Global Digital Format Registry. *Library Trends*, *54*(1), 125-143.

Abrams, S. Morrissey, S. & Cramer, T. (2009, December 7). "What? So What": The Next-Generation JHOVE2 Architecture for Format-Aware Characterization. *International Journal of Digital Curation*, *4*(3). Retrieved April 10, 2011 from http://ijdc.net/index.php/ijdc/article/view/139

Abrams, S. (2007). *JHOVE2: A Next-Generation Architecture for Format-Aware Digital Object Preservation Processing*. Retrieved April 11, 2011 from http://bitbucket.org/jhove2/main/wiki/documents/JHOVE2-Project-Proposal.doc.

Adobe Systems Incorporated. (2008). *Document management - Portable document format - Part 1: PDF 1.7*. Retrieved April 13, 2011 from http://www.adobe.com/devnet/acrobat/pdfs/PDF32000_2008.pdf.

Adobe Systems Incorporated. (1992). *TIFF Revision 6.0: Final*. Retrieved April 10, 2011 from http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf.

Arms, C., & Fleischhauer, Carl. (2005). Digital Formats: Factors for Sustainability, Functionality, and Quality. Presented at the IS&T Archiving Conference, Washington, D.C. Retrieved April 13, 2011 from http://memory.loc.gov/ammem/techdocs/digform/Formats_IST05_paper.pdf

Artefactual Systems, Inc. (2009). *DROID, JHOVE, NLNZ Metadata Extractor*. Retrieved April 11, 2011, from http://artefactual.com/wiki/index.php?title=DROID,_JHOVE,_NLNZ_Metadata_Extractor

Artefactual Systems, Inc. (2009). *Test File Results*. Retrieved April 11, 2011, from http://artefactual.com/wiki/index.php?title=Test_File_Results.

Ball, A. (2006). Briefing Paper: the OAIS Reference Model. UKOLN. Retrieved April 11, 2011 from http://www.ukoln.ac.uk/projects/grand-challenge/papers/oaisBriefing.pdf

Bearman, D. (December 6, 1994). *Toward a Reference Model for Business Acceptable Communications*. Retrieved April 14, 2011 from http://web.archive.org/web/19970707064048/http://www.lis.pitt.edu/~nhprc/prog6-5.html

Bearman, D & Sochats, K. (1995). *Metadata Requirements for Evidence*. Retrieved April 11, 2011 from http://www.archimuse.com/papers/nhprc/BACartic.html

Becker, C., Rauber, A., Heydegger, V., Schnasse, J., & Thaller, M. (2008). A generic XML language for characterising objects to support digital preservation. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 402-406). Fortaleza, Ceara, Brazil: ACM. Retrieved April 11, 2011 from http://portal.acm.org/citation.cfm?id=1363686.1363786&coll=portal&dl=ACM&CFID=5764831&CFTOKEN=12441359

Bekaert, J., & Van de Sompel, H. (2005). *Access Interfaces for Open Archival Information Systems based on the OAI-PMH and the OpenURL Framework for Context-Sensitive Services*. Retrieved April 11, 2011 from http://www.ukoln.ac.uk/events/pv-2005/pv-2005-final-papers/032.pdf

Brody, T., Carr, L., Hey, J. M., Brown, A., & Hitchcock, S. (2007). PRONOM-ROAR: Adding Format Profiles to a Repository Registry to Inform Preservation Services. *International Journal of Digital Curation*, *2*(2). Retrieved April 11, 2011 from http://www.ijdc.net/index.php/ijdc/article/viewArticle/53

Brown, A. (2007). Developing Practical Approaches to Active Preservation. *International Journal of Digital Curation*, *2*(1). Retrieved April 11, 2011 from http://www.ijdc.net/index.php/ijdc/article/viewArticle/37

Brown, A. (2006). Digital Preservation Technical Paper 1: Automatic Format Identification Using PRONOM and DROID. Retrieved April 10, 2011 from http://sourceforge.net/projects/droid/files/droid-technical/Technical%20Documentation/Pronom%20Documentation/Technical_Paper_1_-_Automatic_Format_Identification_v2.pdf/download

Buckheit, J.B., & Donoho, D.L. (1995). *WaveLab and Reproducible Research*. Retrieved April 10, 2011 from http://www-stat.stanford.edu/~wavelab/Wavelab_850/wavelab.pdf.

Caplan, P. (2009). *Understanding PREMIS*. Library of Congress. Retrieved April 10, 2011 from http://www.loc.gov/standards/premis/understanding-premis.pdf.

CCSDS [Consultative Committee for Space Data Systems]. (2002). *Reference Model for an Open Archival Information System.* Blue Book, Issue 1. January 2002. 650.0-B-1. Retrieved April 11, 2011 from http://public.ccsds.org/publications/archive/650x0b1.pdf

CCSDS [Consultative Committee for Space Data Systems]. (2009). *Reference Model for an Open Archival Information System*. Pink Book, Issue 1.1. August 2009. 650.0-P-1.1. Retrieved April 11, 2011, from http://public.ccsds.org/sites/cwe/rids/Lists/CCSDS%206500P11/Attachments/650x0p11.pdf

Copeland, G. P., & Khoshafian, S. N. (1986). Identity and versions for complex objects. *Proceedings on the 1986 international workshop on Object-oriented database systems*. Pacific Grove, California, United States: IEEE Computer Society Press. Retrieved from http://portal.acm.org/citation.cfm?id=318826.318873&coll=portal&dl=ACM.

Digital Curation Centre. (2010). *What is Digital Curation?*. Retrieved April 10, 2011from http://www.dcc.ac.uk/digital-curation/what-digital-curation.

Fedora Development Team. (2005). *Fedora Open Source Repository Software: White Paper.* Retrieved April 13, 2011 from http://web.archive.org/web/20070403235401/http://www.fedora.info/documents/WhitePaper/FedoraWhitePaper.pdf

Ferreira, M., Baptista, A. A., & Ramalho, J. C. (2006). A Foundation for Automatic Digital Preservation. *Ariadne*, (48). Retrieved April 11, 2011 from http://www.ariadne.ac.uk/issue48/ferreira-et-al/

Ferreira, M., Baptista, A., & Ramalho, J. (2007). An intelligent decision support system for digital preservation. *International Journal on Digital Libraries*, *6*(4), 295-304. Retrieved April 11, 2011 from doi:10.1007/s00799-007-0013-x

Giaratta, D. (2009). Significant Properties, Authenticity, Provenance, Representation Information, and OAIS. *iPres Conference, 5 & 6 October 2009, San Francisco, CA*. Retrieved April 11, 2011 from http://www.cdlib.org/services/uc3/iPres/presentations/GiarettaSigProps.pdf.

Hedstrom, M. (1988). Optical Disks: Are Archivists Repeating the Mistakes of the Past? *Archives & Museum Informatics Newsletter, 2,* 53-54. Retrieved April 10, 2011 from http://www.archimuse.com/publishing/AMInewsletters/AMInewsletter1988_2-3.pdf

Higgins, S. (2008). The dcc curation lifecycle model. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries* (pp. 453-453). Pittsburgh PA, PA, USA: ACM. Retrieved from http://portal.acm.org/citation.cfm?id=1378889.1378998&coll=portal&dl=ACM&CFID=5764831&CFTOKEN=12441359

Hockx-Yu, H., & Knight, G. (2008). What to Preserve?: Significant Properties of Digital Objects. *International Journal of Digital Curation*, *3*(1). Retrieved April 12, 2011, from http://www.ijdc.net/index.php/ijdc/article/view/70

IBM Corporation and Microsoft Corporation. (1991). Multimedia Programming Interface and Data Specifications 1.0. Retrieved April 13, 2011 from http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/Docs/riffmci.pdf.

IDEALS. (2008). *Preservation-related Tools*. Retrieved April 11, 2011 from https://services.ideals.illinois.edu/wiki/bin/view/IDEALS/Internal/PreservationTools.

Ince, D. (2010, February 5). If you're going to do good science, release the computer code too *guardian.co.uk*. Retrieved March 27, 2011, from http://www.guardian.co.uk/technology/2010/feb/05/science-climate-emails-code-release

Knight, G., & Pennock, M. (2009). Data Without Meaning: Establishing the Significant Properties of Digital Research. *International Journal of Digital Curation*, *4*(1). Retrieved April 10, 2011 from http://www.ijdc.net/index.php/ijdc/article/view/110

Knight, S. (2005). Preservation Metadata: National Library of New Zealand Experience. *Library Trends*, *54*(1), 91-110.

LeVeque, R.J. (2009). Python Tools for Reproducible Research on Hyperbolic Problems. *Computing in Science and Engineering (CiSE),* 11(2009): 19-27. Retrieved March 26, 2011 from http://www.amath.washington.edu/~rjl/pubs/cise09/cise09.pdf.

Littman, J. (2006). A Technical Approach and Distributed Model for Validation of Digital Objects. *D-Lib Magazine*, *12*(5). Retrieved April 11, 2011 from doi:10.1045/may2006-littman

Lord, P., Macdonald, A., Lyon, L., & Giaretta, D. (2004). From Data Deluge to Data Curation. *eScience All Hands Meeting 2004*, 371-375. Retrieved April 10, 2011 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.7425&rep=rep1&type=pdf

McCullough, B. (2007). Got Replicability? The "Journal of Money, Credit, and Banking" Archive. *Econ Journal Watch* 4(3), 326-337. Retrieved March 26, 2011 from http://www.econjournalwatch.org/pdf/McCulloughEconomicsInPracticeSeptember2007.pdf.

METS Editorial Board. (2010). *<METS> Metadata Encoding Transmission Standard: Primer and Reference Manual (Revised)*. Digital Library Foundation. Retrieved April 10, 2011 from http://www.loc.gov/standards/mets/METSPrimerRevised.pdf.

National Digital Newspaper Program. (2010). Technical Specifications 2009 and 2010 - Profiles and Schemas: National Digital Newspaper Program (A partnership between the Library of Congress and the National Endowment for the Humanities). Retrieved April 13, 2011, from http://www.loc.gov/ndnp/techspecs.html

Nguyen, Q. (2008). Performance Study of Digital Object Format Identification and Validation Tools. *Digital Library Federation Forum Fall 2008 Forum, Nov 11-14, 2008*. Retrieved April 11, 2011 from http://www.diglib.org/forums/fall2008/presentations/Nguyen.pdf.

NISO. (2006). ANSI/NISO Z39.87 - Data Dictionary - Technical Metadata for Digital Still Images. NISO Standards. Retrieved April 10, 2011 from www.niso.org/standards/z39-87-2006/.

OCLC/RLG Working Group on Preservation Metadata (June 2002). *A Metadata Framework to Support the Preservation of Digital Objects*. Dublin, OH: OCLC. Retrieved October 1, 2010 from http://www.oclc.org/research/pmwg/pm_framework.pdf.

PREMIS Editorial Committee. (March 2008). *Data Dictionary section from PREMIS Data Dictionary for Preservation Metadata.* Retrieved October 2, 2010 from http://www.loc.gov/standards/premis/v2/premis-dd-2-0.pdf.

PREMIS Editorial Committee. (March 2008). *Introduction and Supporting Material from PREMIS Data Dictionary for Preservation Metadata.* Retrieved October 2, 2010 from http://www.loc.gov/standards/premis/v2/premis-report-2-0.pdf.

PREMIS Working Group. (May 2005). Data Dictionary for Preservation Metadata: Final Report of the PREMIS Working Group. Dublin, OH: OCLC. Retrieved April 20, 2011 from http://www.oclc.org/research/projects/pmwg/premis-final.pdf.

Preserv.org.uk. (n.d.). *Preservation Services*. Retrieved April 11, 2011 from http://preserv.eprints.org/guide/preservation/?slide=1.

Prom, C. (2010). *Using DROID for Appraisal*. Retrieved April 11, 2011, from http://e-records.chrisprom.com/?p=861.

Rosenthal, D. (2009). *Postel's Law*. Retrieved April 14, 2011 from http://blog.dshr.org/2009/01/postels-law.html.

Sawyer, D. & Reich, L. (April 1996). *Archiving Reference Model, Version 4*. Retrieved April 15, 2011 from ftp://nssdcftp.gsfc.nasa.gov/standards/nost/isoas/int02/refmod4.ps

Swiss Federal Archives. (n.d.). *Archiving of Databases: SIARD Suite*. Retrieved April 11, 2011 from http://www.bar.admin.ch/dienstleistungen/00823/00825/index.html?lang=en.

Swiss Federal Archives. (2008). *SIARD: Format Description*. Retrieved April 11, 2011 from http://www.bar.admin.ch/dienstleistungen/00823/00825/index.html?lang=en&download=M3wBPgDB/8ull6Du36WenojQ1NTTjaXZnqWfVp3Uhmfhnapmmc7Zi6rZnqCkkIN0hHx+bKbXrZ6lhuDZz8mMps2gpKfo.

Tansley, R., Bass, M., & Smith, M. (2004). DSpace as an Open Archival Information System: Current Status and Future Directions. In R. Heery and L. Lyon (Eds.), *Research and Advanced Technology for Digital Libraries* (pp. 446-460). Retrieved from http://www.springerlink.com/content/hdepd4443hl00k4k

The National Archives[, UK]. (n.d.). PRONOM | Welcome. Retrieved September 30, 2010, from http://www.nationalarchives.gov.uk/PRONOM/Default.aspx

The National Archives[, UK]. (2011). *Update – Linked Data and PRONOM*. Retrieved April 14, 2011 from http://labs.nationalarchives.gov.uk/wordpress/index.php/2011/01/linked-data-and-pronom/.

Unified Digital Formats Registry. (2009). *Proposal and Roadmap*. Retrieved April 11, 2011 from http://www.gdfr.info/udfr_docs/Unified_Digital_Formats_Registry.pdf

University of London Computing Centre. (2006, December). *Digital Asset Assessment Tool – File Format Testing Tools - Version 1.2*. Retrieved October 22, 2010 from http://www.ulcc.ac.uk/uploads/media/DAAT_file_format_tools_report.pdf.

Wilson, A. (2007). *Significant Properties Report*. Arts and Humanities Data Service. Retrieved April 13, 2011 from http://www.significantproperties.org.uk/wp22_significant_properties.pdf.

APPENDIX A

FROM THEORY TO PRACTICAL IMPLEMENTATION: ESTABLISHING TECHNICAL AND PRESERVATION METADATA FRAMEWORKS

In 2000 the OCLC/RLG Working Group on Preservation Metadata was established as a joint venture by the two named organizations (OCLC/RLG Working Group on Preservation Metadata, June 2002). The aim of the group was "to develop a preservation metadata framework applicable to a broad range of digital preservation activities" (OCLC/RLG Working Group on Preservation Metadata, June 2002, 1). Using the not-then-formally-published OAIS Reference Model, in June 2002, the group published its report: *A Metadata Framework to Support the Preservation of Digital Objects*. By way of introducing OAIS, the *Framework* actually begins by discussing the notion of Representation Information in OAIS, giving it some prominence (OCLC/RLG Working Group on Preservation Metadata, June 2002, 7). The authors use a text file to illustrate how the form – ASCII – would be considered Structure Information while the language of the text – English – would be considered Semantic Information. For preservation metadata purposes, the *Framework* divided Representation Information into two parts, one of which is Content Data Object Description. The Content Data Object Description should record such aspects of a data object as its structural type (image, sound, etc.), file description (if an image file, the dimensions, resolution, color palette, etc.) file size (bytes), etc. The *Framework* also devotes significant space to explicating the role and need for Fixity information, which serves to record and verify a data object's authenticity. Fixity information in this context is primarily the act of capturing a checksum, or some form of hash value unique to only that particular file, and recording that checksum for future authentication. This piece of information is vital to determining whether the stored data object is, at the byte-level, unchanged since its ingest into the system and provides one way to validate the integrity and authenticity of the data object.

Stemming from the work of the OCLC/RLG Working Group on Preservation Metadata, in 2003, work began on Preservation Metadata Implementation Strategies, which is more commonly known as PREMIS (http://www.loc.gov/standards/premis/). The PREMIS group – also

established by OCLC and RLG and composed of a body of representatives from academic institutions, government organizations, and national libraries – translated the *Framework* "into a set of implementable units" (PREMIS Editorial Committee, March 2008, *Introduction and Supporting*, 3). It published its core documentation, the *Data Dictionary*, in 2005, complete with specifications to serialize PREMIS data in XML. The *Data Dictionary* supplies room for information about a data object's fixity (identifying hash values), format (of the file type), and various other "significant properties," or what OAIS would consider part of a data object's Representation Information. PREMIS not only identified particular attributes about any given file's Representation Information, thereby adding emphasis to these properties and providing the community with a means to store that information, but PREMIS also supplied space to record additional Representation Information about a data object.

As such, members of the digital preservation community have developed metadata schemes to partially record data pertaining to an object's Representation Information. The NISO Metadata for Images in XML Schema (MIX) (http://www.loc.gov/standards/mix/), first published in 2004, and Technical Metadata for Text (textMD) (http://www.loc.gov/standards/textMD/), first implemented at New York University in late 2001/early 2002, are two such metadata formats. MIX is based on the 2006 *Data Dictionary – Technical Metadata for Digital Still Images* published by NISO. The NISO Data Dictionary "defines a set of metadata elements...to support the long-term management of and continuing access to digital image collections," among additional objectives (NISO, 2006). A brief examination of the schema will show that it provides a means to record identification and characterization information, both of which might be used in the future validation of the file, for still images. These include, but are not limited to, information about an image's Color Profile, ICC Profile, YcbCr Sub Sampling, format considerations, such as Codecs and Codec Versions, and much more. Unlike MIX, textMD is based not on an established data dictionary, but originally in support of digital collection development at New York University. Yet, like MIX, the information textMD is designed to capture is detailed technical metadata in support of the identification and characterization of a given text file. textMD provides space to record information about a text file's encoding, character sets, languages, byte order, and line break information, among many additional properties. Both MIX and textMD have been formally associated with Jhove, a file characterization tool discussed below, and PREMIS. MIX and textMD records can be embedded

in PREMIS.

The *PREMIS Data Dictionary* also expends considerable efforts to explain the importance of the "format" element, and related supporting elements, and their role in digital preservation, noting that "a preservation repository must record format information as specifically as possible" (PREMIS Working Group, 4-1). It further clarifies that simply recording the file extension or MIME type is not enough. It is necessary to record more specific information such as the format name and version (TIFF is the format name from the example in the Introduction above; version was unspecified but could be 1, 2, 3, 4, 5, or 6). The PREMIS group recognized that registries for this type of information would not only be necessary but also the most scalable way to approach organizing and accessing this type of information.

APPENDIX B

DATA PROCESSING

As noted in the main text, DROID seeks only to identify the file and so reports on whether the file's identification was "positive," "tentative," or "not identified." Included in all DROID XML output is information pertaining to the identification session, such as which DROID version performed the test, which DROID signature file was used, and the date and time of the identification. The DROID signature file is a truncated version of the PRONOM database, a file including enough information for file identification. The signature file was version 35 and dated 10 May 2010. This is a substantial file, one that includes at the very minimum a basic entry for every file type included in PRONOM, of which there are 728. A DROID typical output file is readily available.[79] For each individually evaluated file there is one IdentificationFile element which hosts an attribute describing DROID's identification confidence for this particular file. Within the IdentificationFile block, the file's path is recorded and a FileFormatHit block for each file format identified for the tested file (Figure 2).

```
−<FileCollection>
    <DROIDVersion>4.0</DROIDVersion>
    <SignatureFileVersion>35</SignatureFileVersion>
    <DateCreated>2010-05-11T03:30:14</DateCreated>
  −<IdentificationFile IdentQuality="Positive">
    −<FilePath>
        /home/snk/thesis-data/ia-2010-01/tth_100105/tth_100105_vbr_mp3.zip
      </FilePath>
    −<FileFormatHit>
        <Status>Positive (Specific Format)</Status>
        <Name>ZIP Format</Name>
        <PUID>x-fmt/263</PUID>
        <MimeType>application/zip</MimeType>
      </FileFormatHit>
    −<FileFormatHit>
        <Status>Positive (Specific Format)</Status>
        <Name>MPEG 1/2 Audio Layer 3</Name>
        <PUID>fmt/134</PUID>
        <MimeType>audio/mpeg</MimeType>
        <IdentificationWarning>Possible file extension mismatch</IdentificationWarning>
      </FileFormatHit>
    </IdentificationFile>
  </FileCollection>
```

Figure 2: DROID Output, Example

79 A JP2 file from the Chronicling America dataset: http://www.3windmills.com/thesis/data/ca/droid/lccn_sn83025287_1882-10-29_ed-1_seq-3/seq-3.jp2.xml

Most identified files have only one FileFormatHit, but DROID supports the identification of some "complex" file formats (meaning an individual file may in fact be composed of multiple file types), as seen in Figure 2. "Complex" types include ZIP archives, which are themselves valid file formats but which are essentially wrappers for at least one other file, and various video file types (such as MPEGs or FLVs) which may contain a video stream and separate audio file.

Detailed and substantial documentation exists explaining not only the structure of the signature file, but also the algorithm DROID employs when attempting to identify a file, specifically the conditions required to make a "positive" or "tentative" identification, or none at all (Brown 2006). DROID's support for file validation extends so far as it can make a "positive" identification of the file's format, which DROID reports only when it can establish a signature match based on a file's internal markers, such as magic numbers (Brown 2006). DROID actually has two levels of "positive" identification: "Specific" and "Generic". "Generic" means that DROID was able to match an internal signature, but it is an internal signature marker shared by a number of different file types, usually versions. These might be different versions of MS Word documents or TIFF images. "Specific" implies that the database had a specific, matching internal marker. It is also possible to have an "IndentificationWarning" of "Possible file extension mismatch," which indicates DROID was able to match an internal marker, but the file extension does not align with any known type suggested by the internal marker. However, it may make a "positive" determination after only matching one internal marker (it's actually based on the number of markers indicated for a particular file type within the DROID signature file), which, though not inconclusive, may not necessarily be definitive for validation purposes. A "tentative" identification is based on matching the file's extension with a known type. DROID reports "not identified" when either the file is a zero-length file or when the software cannot establish a match based on internal (magic numbers) or external (file extension) markers. Armed with this information, it is clear why DROID "tentatively" identifies a basic text file,[80] but "positively" identifies an Open Document Text document as a ZIP archive.[81] As for the positive identification of the Open Document Text file as a ZIP archive, while this is technically correct (an ODT file is a ZIP archive with an altered file extension), it is not "correct" in digital

80 http://www.3windmills.com/thesis/data/ia/droid/outformation2010-01-02.matrix_dts.flac16/outformation2010-01-02_info.txt.xml

81 http://www.3windmills.com/thesis/data/userfiles/droid/02bdec920168576c03ca622136eba8bf/02bdec920168576c03ca622136eba8bf.odt.xml

preservation, file identification terms. Beyond identification, and some insecure support for validation, DROID captures no characterization information beyond reporting a file's MIME type.

Table 8: General Tool Capabilities

General Tool Capabilities				
	Identification	**Validation**	**Characterization**	
JHove	X	X	X	
DROID	X	?		Validation fairly weak (and not an expressed aim), and dependent on purely technical characteristics of the data object.
MET	X	X	X	
Local	X	?		Same as for DROID, but weaker still

JHove and MET, on the other hand, seek not only to identify the file, but also capture sufficient metadata about the file as to faithfully validate and characterize it (Table 8). But, whereas DROID is backed by PRONOM's extensive database, JHove and MET are only capable of properly identifying, validating, and characterizing a select number of file types. This is a significant (and acknowledged) shortcoming of these tools. And, unsurprisingly, when JHove fails to properly identify and characterize a given MP3 or PPT file, it is because it has not been designed with that functionality in mind. But, when given a file type with which JHove has been designed to identify and characterize, the output is far richer than DROID's output (Figure 3).

From the example in Figure 3, it is possible to see that, in addition to registering a "reportingModule" of UTF-8, a format of "UTF-8," and a MIME type of "text/plain," all of which effectively combine to identify the file, JHove has also recorded additional "properties" about the file, such as the number of "Characters" (14903), the various Unicode Blocks detected, and the types of LineEndings. Near the bottom, JHove has included checksum hashes for the file, which can aid in authenticating it in the future (for a user wishing to know it is identical to the one stored in the system or as a check for bit rot).

```
−<jhove xsi:schemaLocation="http://hul.harvard.edu/ois/xml/ns/jhove http://hul.harvard.edu/ois/xml/xsd/
  <date>2010-05-11T08:24:02-04:00</date>
 −<repInfo uri="/home/snk/thesis-data/ca-overcoats/lccn_sn83030193_1906-01-05_ed-1_seq-5/ocr.txt">
    <reportingModule release="1.4" date="2007-08-30">UTF8-hul</reportingModule>
    <lastModified>2010-05-09T23:47:31-04:00</lastModified>
    <size>14906</size>
    <format>UTF-8</format>
    <status>Well-Formed and valid</status>
    <mimeType>text/plain; charset=UTF-8</mimeType>
   −<properties>
    −<property>
      <name>UTF8Metadata</name>
     −<values arity="List" type="Property">
      −<property>
        <name>Characters</name>
       −<values arity="Scalar" type="Long">
          <value>14903</value>
        </values>
      </property>
      −<property>
        <name>UnicodeBlocks</name>
       −<values arity="List" type="String">
          <value>Basic Latin</value>
          <value>Latin-1 Supplement</value>
        </values>
      </property>
      −<property>
        <name>LineEndings</name>
       −<values arity="List" type="String">
          <value>LF</value>
        </values>
      </property>
     </values>
    </property>
   </properties>
   −<checksums>
      <checksum type="CRC32">93e15eca</checksum>
      <checksum type="MD5">249e027a2f3dba37b055a379949978a9</checksum>
      <checksum type="SHA-1">706328178303bd8151acab7ba4f4a1aefcdf75bd</checksum>
    </checksums>
   −<note>
      Additional representation information includes the line endings: CR, LF, or CRLF
    </note>
  </repInfo>
</jhove>
```


Figure 3: JHove Output, Example

Once the data was collected, the data produced by each tool was normalized to facilitate comparison of the results. DROID's three-option identification model was adopted and applied to the data from JHove and MET, though both of these services do not explicitly report an identification confidence measure. Therefore, a confidence measure was developed for JHove and MET output (Table 9).

Table 9: Identification Matrix

Identification Matrix			
	Positive	**Tentative**	**Not Identified**
JHove	Presence of a sigMatch element in output	Reporting module is something other than BYTESTREAM	No sigMatch, module is BYTESREAM or not reported
JHove-with-S-option	Same as JHove		
DROID	Status reported is positive (based on internal markers, such as magic numbers)	Status reported is Tentative (identification based on file extension)	Status reported as not identified
MET-1	Presence of File element	N/A	No File or MIME type of "file/unknown"
MET-2	Root element something other than DEFAULT and a MIME type is reported	Root element is DEFAULT	No MIME type reported
Local	Everything Positive (Local tool always returns something, but may have been file/unknown)	N/A	N/A

JHove "positively" identifies all files. However, if it is unable to identify a file using one of its modules, JHove considers the file simply a well-formed bytestream, with a mimetype of "application/octet-stream." Although not technically incorrect, it is such a vague file identification as to be the equivalent of "not identified," an acknowledged shortcoming of the current JHove software (Abrams, Morrissey, & Cramer 2009). For normalization purposes, if JHove reported a module within the "sigMatch" element in its output format, the file was considered Positively identified. If JHove used a "reportingModule" other than "Bytestream" but did not report a "sigMatch," the identification was treated as "Tentative." If JHove simply reported that the file was a well-formed bytestream, it was considered "Not identified." Both JHove and Jhove-with-S-option were treated in this manner.

The Metadata Extraction Tool functions similarly to JHove, but the the MET-1 and MET-2 normalization procedures were slightly different. Like JHove, the MET tool does not explicitly positively identify the file. It does report, however, a mimetype of "file/unknown" for those it is unable to identify. Therefore, all files analyzed in the MET-1 test were classified as "Identified"

or “Not identified.” If the XML output contained a File element and did not report a mimetype of “file/unknown” the file was considered to be “Positively” identified. Otherwise, it was classified as “Not identified.” In the MET-2 test, semantics are embedded in the root XML element name. If that element was “DEFAULT,” then the result was considered a “tentative” identification. Met-2 still might have reported, however, a MIME type of “file/unknown” for a Tentative match. If the output did not contain a MIME type at all, it was considered “not identified.”