Archive Ingest and Handling Test (AIHT)
Final Report of the Harvard University Library

Originally issued 2005-04-22
Revised 2005-10-27

Introduction

The Archive Ingest and Handling Test (AIHT) project was organized by the Library of Congress (LoC) in part to test a key assumption of the evolving National Digital Information Infrastructure and Preservation Planning (NDIIPP) infrastructure;[1] namely, that significant bodies of digital content can be easily transferred without loss between institutions utilizing radically different preservation architectures and technologies.

The significant areas of Harvard's interest in the AIHT project included:

- An opportunity to make enhancements to the HUL production preservation repository, the Digital Repository Service (DRS) <http://hul.harvard.edu/ois/systems/drs/>. The DRS has been in operation for over 4 years with a policy that limits deposit to objects created by known workflows, in a small set of approved formats, and accompanied by reliable preservation metadata. As the repository policy evolves towards that of an open institutional repository, the AIHT project presented the opportunity to investigate issues surrounding deposit of arbitrary content of unknown provenance.

  These enhancements included:

  - A tool to generate repository Submission Information Packages (SIP) packages automatically. This tool is based on JHOVE, the JSTOR/Harvard Object Validation Environment (pronounced "jove") <http://hul.harvard.edu/jhove/>, which was enhanced to support the AIFF, WAVE, and HTML formats as part of the project.
  - Identification of deposit-time constraints on object format and technical metadata values. These constraints enforce the technical specifications built into existing workflows. In an environment in which digital objects of unknown provenance are accepted, these constraints need to be removed.
  - Preliminary investigation of an enhanced metadata model to record PREMIS-like provenance information about the objects under managed storage.
  - Adding an export function to the repository API. Prior to AIHT, repository development was focused on ingest, storage, administrative reporting, and preservation issues.

- An opportunity to investigate JPEG-2000-based preservation transformations. The DRS has begun to provide support for JPEG 2000-encoded images. A number of current depositors wish to perform retrospective conversion of existing image data stored in TIFF and JPEG formats in order to take advantage of improved delivery services provided by JPEG 2000, e.g., dynamic zoom and pan. AIHT provided the opportunity to investigate preservation transformations of various image formats in common usage in the digital library community, e.g., GIF, JPEG, and TIFF, to the JPEG 2000 (JP2) format.

The AIHT project was administered for the LoC by Information Systems Support, Inc. (ISS) <www.iss-md.com>. The project particulars were laid out in the ISS Request for Proposal (RFP) 26504-01 and its related Statement of Work (SOW) of December 19, 2003. Four institutions participated in the test:

[1] Library of Congress, *Preserving Our Digital Heritage: Plan for the national Digital Information Infrastructure and Preservation Program*, October 2002 <http://www.digitalpreservation.gov/repor/ndiipp_plan.pdf>.

- Harvard University
- Johns Hopkins University
- Old Dominion University
- Stanford University

A project proposal from the Harvard University Library was submitted to ISS on January 23, 2004; Delivery Order SA-04-0011-01 was executed on April 8, 2004, specifying a work period from February 1, 2004, to January 31, 2005. All participating institutions were later granted a no-cost extension of the work period to March 31, 2005.

Within HUL the Principal Investigator (PI) for the AIHT project was Dale Flecker, Associate Director for Planning and Systems; the project manager was Stephen Abrams, Digital Library Program Manager. The following HUL staff members contributed significant effort to AIHT project activities:

- Stephen Chapman, Preservation Librarian for Digital Initiatives
- Suzanne Kriegsman, Digital Projects Librarian
- Julian Marinus, Programmer/Analyst
- Gary McGath, Digital Library Software Engineer
- Germain Seac, Production Systems Programmer/Systems Administrator
- Robin Wendler, Metadata Analyst

Additional project support was provided by the technical staffs of the HUL Office for Information Systems (OIS) Network and Desktop Support (NDS) and Systems Administration and Operations Support (SAOS) teams. The staff at the Harvard College Library (HCL) Digital Image Group (DIG) also provided valuable assistance during the Phase III post-migration quality review. Aware, Inc. (Bedford, Massachusetts), the vendor for the JPEG 2000 codec used by HUL also provided important technical support during the Phase III migration process. Discussions with the other project participants, both formal and informal, also proved to be interesting and fruitful.

The test corpus used for the AIHT project is the September 11 Digital Archive <http://www.911digitalarchive.org/> organized by the American Social History Project at the City University of New York Graduate Center and the Center for History and New Media at George Mason University (GMU), which hosts the collection. The archive comprises a thematically unified set of files in heterogeneous formats that were collected "to preserve and present the history of the September 11, 2001, attacks in New York, Virginia, and Pennsylvania and the public responses to them." [2]

Although the original SOW indicated that the corpus consisted of approximately 175,000 files (14 GB), the size of the collection as delivered was 57,450 files (12.2 GB). This decrease in the scope of the corpus (67% fewer files, 13% smaller total size) did not materially affect project planning or implementation.

The AIHT project was structured in four phases:

- Phase I – Import the test corpus from LoC
- Phase II – Export to/import from a fellow participating institution

[2] *About the Archive* (accessed March 8, 2005) <http://www.911digitalarchive.org/about/>.

- Phase III – Selected format migration
- Phase IV – Production of this final report

A narrative review of the work performed by HUL during the three substantive project phases is provided in subsequent sections, followed by general conclusions. Additional technical details of project activities are provided in the report appendices.

1 Phase I

The intent of Phase I of the AIHT project was to ingest the test corpus, provided with minimal administrative metadata, into a local preservation repository system. The accompanying metadata provided a file manifest and MD5 checksum values for the individual files in the collection; in particular, however, *no reliable* technical metadata was provided.

The nominal process defined by HUL for Phase I was:

- Staging the data off of the disk received from LoC
- Virus checking
- Verification of the transfer manifest
- Conversion to SIP package required by the HUL Digital Repository Service (DRS)
- Data load into the DRS

1.1 Staging

The test corpus was delivered to HUL on an NTFS-formatted TT-345 hard disk, S/N #ISS002, which was mounted on a Windows 2000 machine (733MHz Pentium-3 with 20 GB local hard disk and 147 GB NAS) via USB 2.0. The data layout of the disk was as follows:

```
archive/
        911da.tar.gz
        transfer911da.md5
dbases/
         access/
                    lc911digitalarchive.mdb
         mysql/
                    dump/
                                        lc911digitalarchive.sql
                    lc911digitalarchive/
                                        *.frm
                                        *.myd
                                        *.myi
                    xml/
                                        lc911digitalarchive.xml
reports/
        origManifest/
                    orig911archive.xml
                    orig911archiveDetail.log
                    orig911archiveNoNL.xml
                    orig911archiveSummary.log
```

The `dbases/` and `reports/` directories contained different versions of collection inventory information. Discrepancies between these inventories and the files actually on the disk will be discussed in § 1.3.

The actual collection data was provided in the compressed tar file `911da.tar.gz` (8.8 GB). The initial attempt to uncompress and disaggregate this file was performed on the Windows machine since all of the individual collection files were to be checked for viruses before transfer to a Unix environment. Due to space limitations on the local hard drive, the target for the decompression and dis-aggregation operations was the NAS device. An initial attempt at decompression using WinZip 8.0 <http://www.winzip.com/> failed due to a maximum file size limitation of 4GB. An investigation of alternative compression/

decompression tools indicated that PentaZip <http://www.pentazip.com/pw/Compression.htm> supported files larger than 4GB. PentaZip was installed and the decompression was again attempted and again failed, this time due to a NAS device limitation on individual file sizes of 2 GB.

At this point the decision was made to transfer the original compressed file to a Solaris 8 machine for decompression and dis-aggregation in a Unix environment where sufficient disk space, without individual file size limitations, was available. The elapsed time for the 8.8 GB ftp transfer using WS_FTP 4.5 <http://www.ipswitch.com/> was approximately 3 hours. After transfer, the file was decompressed using gzip 1.3 <http://www.gzip.org/>, which includes the 4g patch necessary for support of file sizes greater than 4 GB, yielding the 12.2 GB tar file `911da.tar`. A communication from ISS of June 21, 2004, alerted all project participants of the necessity of using GNU tar 1.13 <http://www.gnu.org/software/tar/> to disaggregate the tar file. Version 1.13.19 was installed and used to successfully disaggregate the file resulting in the following structure:

```
dbases/
         mysql/
                      dump/
                                          lc911digitalarchive.sql
                      lc911digitalarchive/
                                          *.frm
                                          *.myd
                                          *.myi
libexec/
         rmt
reports/
         origManifest/
                      orig911archive.xml
                      orig911archiveDetail.log
                      orig911archiveNoNL.xml
                      orig911Summary.log
share/
         locale/
websites/
          chnm/
                      september11/
                                  REPOSITORY/
                                             CONTRIBUTORS/
                                             EMAILS/
                                             IMAGES/
                                             LC_ART/
                                             LC_EMAIL/
                                             LC_STORIES/
                                             NMAH/
                                             RC_STORIES/
                                             SATAN/
                                             SEIU/
                                             STORIES/
                                             TOMPAINE/
                                             TYR_IMAGES/
```

The `dbases/` and `reports/` directories duplicated files already provided in uncompressed/ disaggregated form on the distribution disk.

The collection data files were in the `websites/` directory tree. In order to perform virus checking on these files it was necessary to transfer them back to the Windows environment on the NAS disk. The elapsed time for the ftp transfer in this direction was significant greater than the original Windows-to-

Solaris transfer.

Direction	*Files*	*Total size*	*Elapsed time*
Windows-to-Solaris	1	8.8 GB	3 hrs
Solaris-to-Windows	57,450	11.6 GB	35 hrs

The per-file overhead in either the generic ftp protocol or the implementation of that protocol by the WS_FTP client is obviously significant.[3] Additionally, there were numerous transfer errors detected by the ftp client that necessitated the automatic restart of transfers for individual files.

This was merely the instance in which concerns were raised about the performance of various software tools and systems under scale.

1.2 Virus checking

All of the collection files were checked for viruses using McAfee VirusScan Enterprise 7.1.0 (scan engine 4.3.20, virus definitions 4370). Surprisingly (albeit a pleasant surprise), all of the files were found to be virus free. Since many, if not most, of these files originated on the open web it may be hypothesized that virus detection and eradication was performed by GMU prior to providing the test corpus to LoC.

1.3 Verify transfer manifest

The `dbases/` directory contained collection inventory information created by GMU in MS Access <http://office.microsoft.com/en-us/FX010857911033.aspx> and MySQL <http://dev.mysql.com/> formats, and contained references to 57,492 collection files, which comports with the figure given in documentation supplied by GMU.[4] The MySQL database was instantiated locally (version 3.22.32) and a flat file was exported in a format that included file pathname, MD5 checksum, MIME type, and file size (see Appendix B). Although this file also had 57,492 records, the range of primary key values ranged from 1 to 57,498, which caused some initial confusion.

Two Perl scripts were created to first verify that all records in the database had a corresponding file on the file system, and then that all files on the file system had a corresponding record in the database. Verification occurred at the level of file existence and MD5 checksum value.

Records in database	*Files found on file system*	*Files missing from file system*
57,492	57,443	49

Files on file system	*Records found in database*	*Records missing from database*
57,450	57,443	7

Of the 49 files documented in the database that were not found on the file system, the discrepancies fell

[3] J. Postel and J. Reynolds, *File Transfer Protocol (FTP)*, IETF STD 9, RFC 959, October 1985 <http://www.ietf.org/rfc/rfc959.txt>.

[4] Marty Andolino and Jim Safley, *September 11 Digit Archive: Transfer of Archive from Center for History and New Media to The Library of Congress*, Version 1.1, February 13, 2004 <http://www.iss-loc.com/aiht/Archive%20 Documents/sept11daCHNMtoLC1.doc>.

into three categories:

- Filename truncation. The database contained records for:

  ```
  CONTRIBUTORS/chris_combs/ ... /button3.asp? ... &url=http:/noscript
  CONTRIBUTORS/chris_combs/ ... /fool-com/big.chart? ... &rand=94
  ```

  while the file system contained the files:

  ```
  CONTRIBUTORS/chris_combs/ ... /button3.asp? ... &url=http:/noscript&javaOK=NO&
  CONTRIBUTORS/chris_combs/ ... /fool-com/big.chart? ... &rand=9451
  ```

- Unix shell escaping. The database contained records for:

  ```
  CONTRIBUTORS/chris_combs/www.timesofindia.com/images\\downarrow.gif
  CONTRIBUTORS/chris_combs/www.timesofindia.com/images\\arrow.gif
  ```

  while the file system contained the files:

  ```
  CONTRIBUTORS/chris_combs/www.timesofindia.com/images\downarrow.gif
  CONTRIBUTORS/chris_combs/www.timesofindia.com/images\arrow.gif
  ```

- Missing files.

Of the 7 files found on the file system that were not documented in the database, the discrepancies fell into three categories:

- Filename case uniqueness. The database contained records for:

  ```
  CONTRIBUTORS/chris_combs/ ... /msnbc.com/site_elements/bantop_ATTACK.gif?a1
  CONTRIBUTORS/chris_combs/ ... /msnbc.com/Site_Elements/dotBlack.gif
  ```

  while the file system contained the files:

  ```
  CONTRIBUTORS/chris_combs/ ... /msnbc.com/site_elements/bantop_attack.gif?a1
  CONTRIBUTORS/chris_combs/ ... /msnbc.com/Site_Elements/dotblack.gif
  CONTRIBUTORS/chris_combs/ ... /msnbc.com/site_elements/dotblack.gif
  ```

- Filename truncation, as above.
- Unix shell escaping, as above.

The `reports`/ directory contained an inventory in XML form created by LoC. Two additional Perl scripts were created to first verify that all records in the XML inventory had a corresponding file on the file system, and then that all files on the file system had a corresponding record in the XML file. Verification again occurred at the level of file existence and MD5 checksum value. The results of this operation confirmed that all files documented in the inventory existed on the file system, and vice versa.

1.4 HUL Digital Repository Service

The HUL Digital Repository Service (DRS) is a preservation and use repository that has been in production operation for over four years. The DRS draws a fundamental distinction between primary content data and metadata about that content. Note, however, that the DRS stores only administrative and technical metadata; descriptive metadata lives external to the repository in catalogs and other discovery services. DRS metadata is stored in an Oracle 9*i* database; content is stored on an RAID-based SAN with automatic replication to an offsite robotic tape library. Access to the Oracle table structure is mediated through a Java API.

Technical metadata requirements have been established for text (including structured text such as XML), raster still image, and audio media types.[5, 6] DRS image metadata is consistent with the draft NISO Z39.87 data dictionary; audio metadata is consistent with the evolving Audio Engineering Society (AES) X098B schema.[7, 8] The DRS also stores typed relationships between objects; for example, object *A* is a derivative of object *B*.

For the purposes of the AIHT project a completely independent instance of the DRS was created to avoid any potential for contamination of the production service and to simplify the post-project deletion of all test data as required by the ISS SOW.

1.5 SIP packaging

DRS data loads are defined in terms of batches. A batch consists of any number of arbitrarily-named content files and a single XML-formatted control file named `batch.xml`. This control file contains administrative and technical metadata about the content files as well as loader directives.[9]

At the onset of the AIHT project, the XML control file was typically generated by various depositing agents around the University using a variety of custom methods that often required a priori human knowledge of the technical characteristics of the content files. As part of the AIHT project, a new tool was created that automated the procedure of generating the deposit control file. This tool, called DSIP, is based on JHOVE.

1.5.1 *JHOVE*

JHOVE, the JSTOR/Harvard Object Validation Environment (pronounced "jove") is an extensible Java-based framework for format specific object identification, validation, and characterization. (See <http://hul.harvard.edu/jhove/>.) At the onset of the AIHT project, JHOVE was made publicly available in version 1.0 (beta 1). This version provided pluggable modules for the ASCII, GIF, JPEG, PDF, TIFF, UTF-8, and XML formats. As part of the AIHT project additional modules were developed for the AIFF, HTML, JPEG-2000, and WAVE formats. Additionally, the existing PDF module was extended to provide support for PDF versions 1.5 and 1.6 and the TIFF module was extended to support the DNG (Adobe digital negative) profile. During the course of the AIHT project, two additional beta versions of JHOVE have been released to the public. The initial production release is anticipated in April 2005. A list of the significant enhancements to JHOVE made during the course of the project is available in Appendix C.

The output format of command-line version of JHOVE is controlled by selection of a pluggable output handler. DSIP is based on an extension of the standard JHOVE Audit output handler. The Audit handler,

[5] Harvard University Library, *DRS Documentation: Administrative Metadata for Digital Still Images*, v1.3, March 26, 2004 <http://preserve.harvard.edu/resources/imagemetadata.pdf>.

[6] Harvard University Library, *Administrative Metadata for Digital Audio*, v1.2, February 11, 2004 <http://preserve.harvard.edu/resources/audiometadata.pdf>

[7] NISO Z39.87-2002/AIIM 20-2002, *Datq Dictionary – Technical Metadata for Digital Still Images*, Draft Standard for Trial use, June 1, 2002–Deccember 31, 2003 <http://www.niso.org/standards/resources/Z39_87_trial_use.pdf>.

[8] AES SC-03-06 Working Group on Digital Library and Archive Systems, *Core audio metadata XML definition* (in preparation).

[9] Harvard University Library, *Digital Repository Service (DRS): User Manual for Data Loading*, Version 3.02, January 11, 2005 <http://hul.harvard.edu/ois/systems/drs/drs_load_manual.pdf>.

introduced in the beta 3 release, is designed to invoke the JHOVE validation operation against all files visited during a breadth-first traversal of a file system hierarchy, producing summary output. (See Appendix D.)

In its current state the Audit handler reports the pathname of each file in absolute form. This may be problematic when using the handler output as the basis for a Dissemination Information Package (DIP). Attempting to re-instantiate the DIP would require matching the entire original pathname. It would be preferable for the Audit handler to reference individual files by their relative pathname and then provide the absolute path of the current working directory.

In August 2004 JHOVE was used in a retrospective validation test of the 1.1 million objects then existing within the DRS. This test uncovered a small, but significant number of systemic validation errors and inconsistencies between external and internal metadata. For details please see the report presented by HUL at the 2004 DLF Fall Forum in Baltimore.[10] The results of this test strongly suggest that JHOVE-like functionality should be integrated into repository workflows at the point of ingest.

1.5.2 *DSIP*

All JHOVE modules determine object well-formedness based on the strict application of the relevant format specification. The JHOVE validation performed by DSIP revealed a number of discrepancies in the format of the objects that formed the test corpus. Format information was available in the GMU inventory in terms of MIME type and, by implication, file extension. (See Appendix E.)

	Files as documented by GMU		***Files as validated***
DSIP/JHOVE MIME type	***by MIMEtype***	***by extension***	***by DSIP/JHOVE***
`application/octet-stream`	1,141	10,613	3,618
`application/pdf`	1,663	1,664	1,659
`audio/x-aiff`	162	151	162
`audio/x-wave`	2,015	2,016	2,015
`image/gif`	1,337	1,320	1,339
`image/jpeg`	12,752	12,763	12,576
`image/tiff`	1,538	1,533	1,537
`text/html`	16,677	16,579	3,649
`text/plain`	20,207	10,822	30,887
`text/xml`	0	1	8
	57,492	57,492	57,450

For each MIME type supported by JHOVE the second column indicates the number of files of that type as specified by the GMU inventory, the third column indicates the number of files of that type implied by the inventory file extension, and the last column indicates the number of files validated by JHOVE as being of that type. (Recall that the inventory contains records for 57,492 files although only 57,450 were present on the file system and thus subject to JHOVE validation.)

At the onset of the AIHT project the DRS provided support for a limited set of formats: AIFF, ASCII, GIF, JPEG, TIFF, UTF-8, and XML. During the course of the project support was added for HTML, PDF, and WAVE. Objects not in these formats can be deposited to the DRS, but only as "opaque" objects about which no format information is maintained. DSIP was therefore designed to declare any file in an unsupported format as MIME type `application/octet-stream`. (JHOVE behaves similarly: any file that is affirmatively validated as a supported format is declared to be of type `application/octet-stream`.) It is desirable that a future version of the DRS have the ability to store

[10] Stephen L. Abrams and Gary McGath, "Format Dependencies in Repository Operation," *DLF Fall Forum*, Baltimore, October 25-27, 2004 <http://www.diglib.org/forums/fall2004/abramsmcgath1004.htm>.

the purported MIME type of deposited files.

The DSIP crosswalk between JHOVE characterization metadata and the equivalent elements of the `batch.xml` control file is described in Appendix F.

The elapsed time for the DSIP traversal over the entire test corpus, including per-file format validation and MD5 checksum calculation,[11] was 9 hours 15 minutes running on a dual 300 MHz processor Sun Enterprise 450 with 2GB RAM and NFS-mounted file system.

1.5.3 *Loading*

For administrative convenience, the test data was loaded in 12 individual batches. The elapsed time for ingest processing, including rudimentary format validation, updating of database tables, and file transfer to the attached SAN, was:

Batch	***Files***	***Size (GB)***	***Elapsed time***	***KB/sec***
1	5,000	2.36	16:05	2,564.39
2	5,000	2.21	16:15	2,376.77
3	5,000	1.11	9:49	1,976.09
4	5,000	0.01	5:27	32.07
5	5,000	0.01	4:52	35.91
6	5,000	0.87	9:05	1,673.87
7	5,000	2.63	18:07	2,537.03
8	5,000	0.56	7:08	1,371.97
9	5,000	0.02	6:27	54.19
10	5,000	0.01	5:48	30.13
11	5,000	0.01	5:47	62.79
12	2,450	0.01	2:57	59.24
	57,450	9.80	1:37:47	1,751.50

A satisfactory explanation for the disparity in deposit throughput—2.5 MB/sec vs. 32 KB/sec—has not yet been determined.

The DRS was designed for highly-curated digital objects, created by known workflows, to pre-existing specifications, and accompanied by reliable metadata. Consequently, a number of constraints restricting allowable data values were defined and enforced by the repository API and the underlying database tables. For the AIHT project it was necessary to relax the following constraints:

- Permit 0 length files (34 files in the test corpus have length of 0)
- Define additional formats for AIFF, HTML, JPEG 2000, PDF, WAVE, and XML
- In the audio metadata, permit bit depth of 8
- In the image metadata:
    - Permit bits per sample of 1,1,1 (an unusual configuration of 3 1-bit channels exhibited by 27 files)
    - Permit compression type of 32773 (PackBits runlength encoding, used by 5 files)

The pre-existing constraints in the DRS were based on the established technical specifications used in production workflows. An open repository, however, must be prepared to accept digital objects with arbitrary technical characteristics.

[11] R. Rivest, *The MD5 Message-Digest Algorithm*, RFC 1321 April 1992 <http://www.ietf.org/rfc/rfc1321.txt>.

One additional problem uncovered during the loading process was the file:

```
CONTRIBUTORS/chris_combs/www.cnn.com/CNN/anchors_reporters/cnni/lüscher.bettina.html
```

whose file pathname used the ISO 8859-1 (Latin 1) encoding for the character “ü” (u with diaeresis). The repository Java API, expecting all pathnames to use the UTF-8 encoding, rejected this file. For convenience the file was renamed.as:

```
CONTRIBUTORS/chris_combs/www.cnn.com/CNN/anchors_reporters/cnni/luescher.bettina.html
```

to avoid any encoding issues.

2 Phase II

The intent of Phase II was to test the interchange of collection data between participating institutions. Each institution exported its internal representation of the test corpus using a locally defined Dissemination Information Package (DIP), and imported the DIP of another participant, in other words, performed a SIP-to-DIP conversion.

The nominal process defined by HUL for Phase II was:

- Export stage
  - Define and document DIP format
  - Add export function to repository API
  - Create DIP and upload to AIHT project web site
- Import stage
  - Evaluate the DIP formats of the other participants
  - Select DIP
  - Download DIP
  - Convert to DRS SIP format
  - Deposit SIP into DRS

2.1 Export

The AIHT Phase II export is formatted as a gzip'ed tar archive (`harvard.tar.gz`). The tar archive contains all of the individual files of the collection and a single METS file (version 1.3, <`http://www.loc.gov/standards/mets/mets.xsd`>) encapsulating administrative and technical metadata about the collection files (see Appendix G):

```
harvard.tar:
  export.xml
  aiht/data/2004/12/17/0/122.jpg
  aiht/data/2004/12/17/0/123.jpg
  ...
  aiht/data/2004/12/21/44/57571.jpg
```

The METS file is approximately 53MB. The tar archive is approximately 12GB uncompressed, 9 GB compressed.

Within the tar archive the collection files are referenced by their pathname as stored in the Harvard University Library (HUL) Digital Repository Service (DRS). The general form of DRS content file pathnames is:

`instance``/data/`*`yyyy/mm/dd/n/pk.ext`*

where

`instance` is the repository instance name;
`yyyy` is the year of deposit;
`mm` is the month of deposit;
`dd` is the day of deposit;
`n` is a sequence number, 0,1,2,… , used to keep the number of content files found in any given directory under 500;

pk	is the primary key of the files administrative record in the DRS database;
ext	is the canonical file extension of the file's data format.

The project-specific repository instance name is `aiht`. The un-italicized portion of the pathname syntax, e.g., "`data`," is an invariant part of the syntax.

Note that due to circumstances arising from deposit-time validation errors, the assignment of primary keys did not occur in a strict sequence. Thus the primary keys of the first and last objects in the export are 122 and 57571, respectively. Nevertheless, the collection contains 57,450 objects.

The canonical file extensions are:

Ext	***MIME type***	***Format***
aif	audio/x-aiff	AIFF
dat	application/octet-stream	Opaque object
gif	image/gif	GIF
jp2	image/jp2	JPEG 2000
jpg	image/jpeg	JPEG
jpx	image/jpx	JPEG 2000
htm	text/html	HTML
pdf	application/pdf	PDF
tif	image/tiff	TIFF
txt	text/plain	Text
wav	audio/x-wave	WAVE
xml	text/xml	XML

Note that the canonical file extensions are assigned to content files by the repository based on the file format. These extensions may be different than those provided in the deposit control file.

2.2 Import

The potential sources for the Phase II import were:

- Johns Hopkins University (JHU). The JHU DIP was distributed as a zip'ed tar file contained individual files representing each directory in the GMU archive; a pair of files for each original file in the GMU archive: one containing the content stream and the other a METS file containing descriptive metadata; and a manifest providing a comprehensive list of DIP file pathnames and MD5 checksums. No technical metadata was supplied.
- Old Dominion University (ODU). The ODU DIP was distributed as a (large) single file formatted using the MPEG-21 Digital Item Description Language (DIDL) to encapsulate all collection content and metadata.[12]
- Stanford University (SU). The SU DIP was distributed as two zip'ed tar files containing the collection content and content technical and provenance metadata, respectively, and a zip'ed METS file containing the administrative metadata.

Do to the lack of local expertise and tools relevant to MPEG-21 DIDL the ODU export was not considered as a source for the Phase II import. The METS-based DIP approach used by JHU and SU provided easy migration paths to the DRS SIP format. The decision to use the SU export was based on the fact that it included technical metadata, while the JHU export did not. This provided the opportunity

[12] ISO/IEC 21000-2:2003, *Information technology – Multimedia framework (MPEG-21) –- Part 2: Digital Item Declaration*.

to investigate the consistency of technical metadata produced by two institutions against the same test corpus.

Conceptually, the SU DIP consisted of a master manifest, aiht-mets.xml, containing descriptive and technical metadata, and two parallel directory structures: one containing the content files, and one containing the technical and provenance metadata. Technical metadata was provided by capturing in file form the output of the standard JHOVE XML handler.

A Perl script was created that iterated over the master manifest file, reading the external technical metadata files, and re-formatted this data in the equivalent DRS SIP form, e.g., the XML-formatted `batch.xml` control file. This process would have been simplified had the SU DIP used the master METS file to encapsulate all known metadata about the collection files. Because the METS schema has separate structures for file manifest (`<fileSec>`) and structural metadata (`<structMap>`) the Perl script generated the DRS SIP in two pieces that were then concatenated together.

As during the initial Phase I, the data was loaded in 12 individual batches. Files were assigned to batches in the sequential order in which they were found by a traversal of the file system. Note that these batches contained different sets of files than the Phase I batches.

Batch	***Files***	***Size (GB)***	***Elapsed time***	***KB/sec***
1	5,000	1.56	14:41	1,856.73
2	5,000	0.01	6:09	28.42
3	5,000	0.01	6:34	26.61
4	5,000	2.59	15:42	2,883.03
5	5,000	1.48	13:08	1,969.41
6	5,000	0.45	7:00	1,123.47
7	5,000	2.62	18:23	2,490.72
8	5,000	1.03	10:36	1,698.17
9	5,000	0.02	5:21	65.33
10	5,000	0.01	5:54	29.62
11	5,000	0.01	5:38	31.02
12	2,450	0.01	2:48	62.42
	57,450	9.8	1:52:54	1,516.98

As with the Phase I deposit, there is a 2 order of magnitude disparity in deposit throughput correlated to batch size that remains unexplained.

As in the initial Phase I import, it was necessary to relax the following constraints:

- Permit compression type 7 (ISO JPEG, used by 560 files)
- Permit photometric interpretation of 32803 (CFA (Color Filter Array), used by 560 files)

The following table provides the breakdown of AIHT files by MIME type as deposited into the DRS during the Phase I and II deposits. The MIME types for the initial import from LoC were based on JHOVE validation using the beta 3 release. All malformed files (or files in formats not supported by JHOVE or files in formats not supported by the DRS) are considered as opaque objects of MIME type `application/octet-stream;` any technical metadata about those files is discarded prior to deposit. It is desirable that future versions of the DRS support the ability to store unverified or unverifiable metadata about opaque objects.

The MIME types for the Stanford import are based on metadata supplied in the Stanford export package. Files in formats not supported by the DRS and files for which required DRS metadata properties were not available were automatically converted to application/octet-stream prior to deposit. The technical metadata in the SU export was generated using the JHOVE beta 2 release.

Format	*MIME type*	*LC import*	*SU import*
AIFF	`audio/x-aiff`	162	162
ASCII or UTF-8	`text/plain`	30,887	30,910
GIF	`image/gif`	1,339	0
HTML	`text/html`	3,649	1,222
JPEG	`image/jpeg`	12,576	10,766
PDF	`application/pdf`	1,659	1,662
TIFF	`image/tiff`	1,537	1,537
WAVE	`audio/x-wave`	2,015	0
XML	`text/xml`	8	5
Unknown	`application/octet-stream`	3,618	11,186
		57,450	57,450

The majority of these discrepancies can be accounted for on the basis of the different JHOVE versions—beta 3 vs. beta 2—being used as the basis for the Harvard and Stanford categorizations. The beta 3 release used by the Harvard DSIP tool incorporating a number of enhancements and error corrections.

- ASCII/UTF-8 (`text/plain`). The Stanford DIP characterized 23 HTML files as plain (i.e., unstructured) text. The Harvard DSIP tool, making use of the new HTML module, characterized these files as HTML, rather than plain text.
- GIF (`image/GIF`). The JHOVE beta 2 GIF module contained a known error in which the image bits/sample property was not reported. As this is a required property for DRS image metadata, all Stanford GIF files were accepted only as opaque objects of type `application/octet-stream`.
- HTML (`text/html`). The HTML module introduced in the beta 3 release was used to characterize the files based on parsing the content. As this facility was not available in the beta 2 release, the Stanford characterization was based on other criteria, most notably, the file extension.
- JPEG (`image/jpeg`). The Stanford DIP characterized 1,810 fewer files as JPEG. The JHOVE beta 2 JPEG module contained a know error in which files containing an optional EXIF metadata segment were erroneously being reported as invalid. This error was corrected in the beta 3 release.
- PDF (`application/pdf`). The Stanford DIP characterized 3 additional files as PDF. The Harvard DSIP tool rejected these files as being not well-formed:

    `CONTRIBUTORS/joe_criscuoli/ArabWar/EPI_911.PDF`

    > The PDF header does not start at byte offset 0. (Although this is a violation of the strict syntax established in *PDF Reference*, the looser requirements for the Acrobat reader allow the header to start anywhere in the first 1024 bytes of the file. Future versions of JHOVE should follow the looser Acrobat requirement with appropriate notification.)

    `CONTRIBUTORS/national_guard_bureau/CRRDB/data/documents/2328.pdf`

    > Invalid destination object.

    `CONTRIBUTORS/national_guard_bureau/CRRDB/data/documents/2721.pdf`

    > Bad page label structure.

- WAVE (`audio/x-wave`). The JHOVE beta 2 WAVE module contained a known error in which the audio data encoding property is not reported. As this is a required property for DRS audio metadata, all Stanford WAVE files were accepted only as opaque objects of type `application/octet-stream`.

- XML (`text/xml`). The Stanford import characterized 3 fewer files as XML. The JHOVE beta 2 XML module did not report the character encoding for these files. As this is a required property for DRS text metadata, these XML files were accepted only as opaque objects of type `application/octet-stream`.

These discrepancies point out the desirability of the widest possible adoption of common characterization criteria and tools by the digital preservation community.

3 Phase III

The intent of Phase III was to investigate preservation format migrations. The selected Phase III migration was the transformation of various image files to JPEG 2000. HUL has recently begun to offer support for the delivery of JPEG 2000 images through standard infrastructure components. There is great local interest in the retrospective conversion of substantial numbers of existing TIFF images to JPEG 2000 in order to enhance the use of still images by permitting users to pan and zoom image content. The AIHT migration simulates the trigger event of a designated community insisting upon enhanced usability, rather than to avoid obsolescence of formats entering their sunset phases.

3.1 Provenance metadata

The current DRS data model does not provide a mechanism to capture information about change events in an object's lifecycle or other provenance metadata. During Phase III HUL began to investigate potential enhancements to the data model that would allow the capture of such metadata. The starting point for this investigation was the work of the PREMIS working group <http://www.oclc.org/research/projects/pmwg/>. HUL developed a number of use cases that modeled known repository activity and mapped these user cases to the evolving PREMIS data model.

The AIHT project schedule did not allow sufficient time for HUL to attempt to implement any changes to the DRS. (In addition to the necessary table model changes, implementation would require significant alteration of existing work flows.) However, the AIHT project team feels that the use case exercise provided valuable experience that will help to streamline the future efforts to enhance the DRS to capture provenance metadata.

Provenance metadata for the derived JPEG 2000 images was maintained in the DRS in the form of discursive text inserted into an existing administrative note field. This note briefly outlined the conversion process, e.g., source format, tools used, rationale for technical specifications, etc.

3.2 Transformation

The AIHT project team defined three goals in developing the transformation method to migrate source image files to the appropriate JP2 format:

1. To preserve fully the integrity of the GIF, JPEG, and TIFF source data when transformed into the JP2 format;
2. To maximize the utility of the new JP2 objects; and
3. To minimize migration costs.

Objective metrics can be used to evaluate rates of success and failure in achieving the first goal, but assessments of utility and cost are necessarily constrained by the performance of present-day tools and services.

Lacking explicit metadata to designate relationships between images in the test corpus (e.g., parent/child, or derivative, relationships), no attempt was made to infer relationships between collection files or to evaluate displayed images in order to identify cases in which the collection contained multiple versions of the same image. Related image files, particularly those designated by curatorially assigned roles such as "archival master", "production master", and "delivery," raise policy questions regarding the selection of appropriate sources for preservation migration. However, no conditional criteria were applied to the selection, or de-selection, of AIHT source images: every JHOVE-validated image file was included the input population for migration. (However, as disclosed in Appendix E, the AIHT collection did include image files in formats not supported by JHOVE or the DRS, e.g., `image/bmp` and `image/png`; none of these images were included in the migration source set.)

The potential source image files in the test corpus were categorized into sub-populations according to the attributes perceived to be meaningful to the pictorial integrity of the source images. Experts in the JPEG 2000 format at Aware, Inc., provided valuable assistance by reviewing the AIHT team's proposed workflow design according to the three stated goals that comprised the rationale for migration. These consultations confirmed that meaningful attributes of source images that could influence the selection of the appropriate JPEG 2000 profile (JP2 or JPX) and the optimal conversion settings in the codec include: photometric interpretation (i.e., color space), compression scheme, number and size of channels, and maximum pixel dimension.

MIME type	***Photointerp***	***Compression***	***BPS***	***Max. pixels***	***Files***		
image/gif	3 palette	5 LZW	8	0 – 300	963	1,339	1,339
				301 – 600	171		
				601 – 1200	204		
				1201 – 2400	1		
image/jpeg	6 YCbCr	6 DCT	8	0 – 300	2	67	12,576
				301 – 600	12		
				601 – 1200	37		
				1201 – 2400	13		
				2401 – 4800	3		
			8 8 8	0 – 300	3,390	12,501	
				301 – 600	3,061		
				601 – 1200	3,729		
				1201 – 2400	2,107		
				2401 – 4800	210		
				4800 – 9600	4		
			8 8 8 8	601 – 1200	8	8	
image/tiff	1 Bitonal	1 Uncompressed	8	2401 – 4800	6	6	1,537
	2 RGB	1 Uncompressed	8 8 8	0 – 300	561	1,510	
				301 – 600	2		
				601 – 1200	927		
				1201 – 2400	2		
				2401 – 4800	18		
		5 LZW	8 8 8	1201 – 2400	3	16	
				2401 – 4800	13		
		32773 PackBits	8 8 8 8 [a]	601 – 1200	5	5	
					15,452	15,452	15,452

[a] RGB TIFFs with unassociated alpha channel

Applying the appropriate criteria to categorize sub-populations of source data for migration is essential to avoid applying transformation algorithms at a level that only preserves the integrity of most, but not all objects. In addition, the sub-populations were created to serve the goal of maximizing the utility of the JPEG 2000 files. For example, knowledge of the maximum pixel dimension of the images is necessary in order to select the proper number of decomposition levels, i.e., zoom levels, for those images. (See Appendix H; "proper" should be understood in terms of the conventions used by HUL's Image Delivery Service (IDS) <http://hul.harvard.edu/ois/systems/ids/>.)

The JPEG 2000 transformations were performed using the command-line interface to the Aware SDK 3.6.0 codec. As this version of the codec did not support GIF as a source format, all GIF images were first transformed into equivalent uncompressed RGB TIFFs using ImageMagick. (See Appendix I.)

The JPEG 2000 format defines two profiles: JP2 and JPX. JP2 is the baseline profile; JPX provide support for non-standard colorimetric handling of images. Neither the GIF nor the JPEG format provides support for calibrated color management. While the TIFF format does provide such support, the current DRS technical metadata for images does not include fields for capturing this information. External processing of the deposited TIFF files using JHOVE revealed that none of the TIFF files made reference to external color profiles (either by name or URL) or made use of the TIFF RGB colorimetry tags: WhitePoint (tag 318) or PrimaryChromaticities (319). However, 14 of the files did contain embedded color profiles for an RGB color space. Since the sRGB is supported by the baseline JP2 profile, there was no necessity to make use of any JPX features supporting color-managed images. Thus, the target format for all transformations was the JP2 profile.

The Aware codec was configured to the following specifications:

- RLCP (resolution-layer-component-position) progression order
- Tile size 1024×1024
- Reversible 5-3 wavelet transform
- Decomposition levels based on source image maximum pixel resolution
- Two quality layers: 50% (35dB pSNR) and 100% (full quality)
- Reversible channel quantization
- Highest quality coding predictor offset
- Grayscale and sRGB colorspaces for single- and multi-channel images, respectively

which are optimized for fast decoding by the widest range of codecs. They are also based on the assumption that the JPEG 2000 files would function both as archival masters (100% quality, lossless transform) as well as use copies that would support pan and zoom operations (50% quality, decomposition levels).

5,031 (32.5%) source files failed the transformation process:

MIME	***Photoint.***	***Compression***	***BPS***	***Max. pixels***	***Source Files***			***JPEG 2000 Files***		
image/gif	3 palette	5 LZW	8	0 – 300	963	1,339	1,339	391	706	706
				301 – 600	171			126		
				601 – 1200	204			188		
				1201 – 2400	1			1		
image/jpeg	6 YCbCr	6 DCT	8	0 – 300	2	67	12,576	2	66	8,183
				301 – 600	12			12		
				601 – 1200	37			36		
				1201 – 2400	13			13		
				2401 – 4800	3			3		
			8 8 8	0 – 300	3,390	12,501		1,922	8,117	
				301 – 600	3,061			1,798		
				601 – 1200	3,729			2,749		
				1201 – 2400	2,107			1,440		
				2401 – 4800	210			207		
				4800 – 9600	4			1		
			8 8 8 8	601 – 1200	8	8		0	0	
image/tiff	1 b/w	1 uncompressed	8	2401 – 4800	6	6	1,537	6	6	1,532
	2 RGB	1 uncompressed	8 8 8	0 – 300	561	1,510		561	1,510	
				301 – 600	2			2		
				601 – 1200	927			927		
				1201 – 2400	2			2		
				2401 – 4800	18			18		
		5 LZW	8 8 8	1201 – 2400	3	16		2	11	
				2401 – 4800	13			9		
		32773 PackBits	8 8 8 8	601 – 1200	5	5		5	5	

15,452	15,452	15,452	10,421	10,421	10,421

The conversion errors fall into the following categories:

- GIF source files (633 files)

  All of these GIF files make use of a transparent background color. (Presumable these are icons used on web pages.) The TIFF files that were derived from the GIFs also define a transparent background color. The Aware codec did not accept TIFFs using transparency as source images.

- JPEG source files (4,393 files)
  - Single channel (8 bit) (1 file)

    Program bug in the Aware codec.
  - 3 channel (8,8,8 bit) (4,384 files)

    Program bug in the Aware codec.
  - 4 channel (8,8,8,8 bit) (all 8 files)

    The codec did not accept 4 channel JPEGs as source images.

- TIFF source files (5 of 1,537)

  Program bug in the Aware codec.

Aware has subsequently released a newer version of the codec, 3.7.1, that corrects these programming errors. Unfortunately this release was not available within the remaining timeframe of the project.

The total file size of the resulting JPEG 2000 files increased 1.4 GB over that of the original source files, a 28% increase. However, the bulk of the increase came in replacing JPEG images with JPEG 2000 images. Since the majority of JPEG images in the DRS are pre-created use copies of TIFF master images, once the TIFFs were replaced by JPEG 2000 files, the JPEGs would no longer be necessary and could be deleted. The general practice is to replace families of files—TIFF master with multiple JPEG deliverables—with a single JPEG 2000 that can function as both an archival master and the source of use images dynamically created at the point of request. Thus, we would anticipate that the aggregate size of images would remain unchanged.

		Source		*JPEG 2000*	
MIME	***Files***	***Total***	***Average***	***Total***	***Average***
image/gif	706	35 MB	51 KB	111 MB	161 KB
image/jpeg	8,183	1.7 GB	214 KB	5.3 GB	678 KB
image/tiff	1,532	3.3 GB	2.2 MB	1.1 GB	700 KB
	10,421	5 .0 GB	501 KB	6.4 GB	647 KB

Since approximately 19% of the original images were in the RGB colorspace, an additional codec option to perform an RGB-to-YUV colorspace transformation could have been specified. Since this has the effect of removing correlations between the color channels, we would have expected to see somewhat better compression ratios.

3.3 Post-transformation QC

Post-transformation quality control had two aspects:

- Automated QC using JHOVE to characterize and compare the source and target images

- Manual QC (side-by-side viewing under controlled conditions using calibrated displays)

The JHOVE validation indicated that all JPEG 2000 images met the specification profile.

Due to limited availability of the appropriate work environment and equipment only a handful of source and target files were manually examined side-by-side under ISO 3664 calibrated viewing conditions. (The production facilities of the Harvard College Library (HCL) Digital Imaging Group (DIG) were used for this process. Additionally, HCL-DIG staff with high "visual literacy" were available for consultation during the examination process.)

MIME	***Photoint.***	***Compression***	***BPS***	***Max. pixels***	***Source file***
image/gif	3 palette	5 LZW	8	601 – 1200	`aiht/data/2004/12/21/0/35474.gif`
				1201 – 2400	`aiht/data/2004/12/21/0/35509.gif`
image/jpeg	6 YCbCr	6 DCT	8	1201 – 2400	`aiht/data/2004/12/21/4/37359.jpg`
				2401 – 4800	`aiht/data/2004/12/21/6/38351.jpg`
			8 8 8	2401 – 4800	`aiht/data/2004/12/17/12/6170.jpg`
				4800 – 9600	`aiht/data/2004/12/21/1/36060.jpg`
image/tiff	1 b/w	1 uncompressed	8	2401 – 4800	`aiht/data/2004/12/17/3/1880.tif`
	2 RGB	1 uncompressed	8 8 8	1201 – 2400	`aiht/data/2004/12/17/17/9107.tif`
				2401 – 4800	`aiht/data/2004/12/17/3/1884.tif`
		5 LZW	8 8 8	1201 – 2400	`aiht/data/2004/12/20/4/12217.tif`
				2401 – 4800	`aiht/data/2004/12/20/4/12230.tif`
		32773 PackBits	8 8 8 8	601 – 1200	`aiht/data/2004/12/17/19/9741.tif` `aiht/data/2004/12/17/19/9750.tif`

In addition to the side-by-side viewing of the source and target images, Adobe Photoshop CS was used to perform pixel-by-pixel comparison of the source and target images stored as independent layers. For all of the GIF and TIFF source files defined in the RGB colorspace the pixel comparison showed no variation between source and target images. For the JPEG source files, the codec needed to perform an additional colorspace transform from YCbCr to sRGB. In these cases Photoshop uncovered minor variations (standard deviations on the order of 0.02) in the pixel-by-pixel comparison, most probably the result of mathematical round-off error in during the colorspace transform. In no cases, however, were these differences visible to the human observers. Thus, the transformation process can be considered at best mathematically lossless and at worst, perceptually lossless.

3.4 Loading

The newly created JPEG 2000 files were deposited into the DRS. Explicit relationships were established within the DRS data model associating the original source image to its derived JPEG 2000 image. The elapsed time for the DSIP traversal over the JPEG 2000 files, including per-file format validation and MD5 checksum calculation, was 1 hour.

4 Conclusions

In general, the implementation of the project occurred along the lines originally defined in the project proposal. There were not any major difficulties that necessitated significant changes to the original HUL project plan.

All project participants successfully completed all project phases. Thus, the project did succeed in validating an approach to long-term digital preservation based on the free transfer of digital assets between institutions using radically different technology bases. This finding is of great significance for the development of the NDIIPP preservation environment. Under such an architecture, every local institution whose digital assets are at potential risk do not have to develop costly expertise in preservation. Instead, these assets could flow to regional or consortial centers of expertise for more efficient preservation handling. This architecture thus supports the widest applicability of preservation effort while at the same time minimizing the cost of that effort.

Based on the knowledge gained through participation in the AIHT project, HUL makes the following specific comments and recommendations:

- General archival transfer
  - While HUL expected to see some impact of scale on the transfer process, it was surprising that scaling issues arose so significantly and in the context of an archive that was not particularly large, either in total size or number.
  - Thus, all aspects of repository workflow, and the systems that implement that workflow, must be carefully designed and deployed in a manner that minimizes the impact of scale.
  - For efficiency, data transfers should be performed on smallest number of component objects as possible. In general the unit of transfer should be a single container object as the determinant of transfer throughput appears to be the total number of objects rather than the total size.
  - Although the AIHT test corpus was successfully transferred between project participants in a variety of locally-defined formats, there is a significant opportunity to minimize the complexity and cost of such future transfer through the development of community-accepted standards for packaging DIPs and tools to perform the packaging and unpackaging of those DIPs. The consensus-building process should be based on relevant work in this area such as XFDU and XOP.[13, 14] The JHOVE Audit handler may also be used as the basis for a common DIP. HUL believes that the consensus opinion of the LoC and the project participants in this regard would be given great weight by the digital preservation community and strongly recommends follow-up work in this area of standardization.

- Transformation to JPEG 2000
  - The transformation of GIF, JPEG, and TIFF source images to JPEG 2000 is amenable to automated processing.
  - Categorizing source images into sub-populations is a necessary pre-processing step to facilitate the transformation according to appropriate specifications.

[13] CCSDS, *XML Formatted Data Unit (XFDU) Structure and Construction Rules*, White Book, September 15, 2004 <http://www.ccsds.org/docu/dscgi/ds.py/GetRepr/File-1912/html/>.

[14] Martin Gudgin et al., eds., *XML-binary Optimized Packaging*, W3C Recommendation, January 25, 2005 <http://www.w3.org/TR/2005/REC-xop10-20050125/>

  - Dependent upon the colorspaces of the source and target images, the transform can be performed at best in a numerically lossless manner, and at worst in a perceptually lossless manner. In all cases the target images can be considered of greater utility than the source images by virtue of the increased range of behavioral contexts in which they can be used, e.g., dynamic zoom and pan.
  - For source files defined in an RGB colorspace, an RGB-to-YUV colorspace transform can be used to maximize JPEG 2000 compression ratios. This transform may result in numerically round-off error discoverable through pixel-by-pixel comparison of source and target images. However, the experience of the AIHT project indicates that these differences are not perceivable to a trained human observer.

- JHOVE
  - Format is a fundamental component of preservation metadata. Without knowing the format of a digital object, access to its full information content is not possible.
  - The variance in technical characterization, including format, uncovered in the Phase II underscores the importance of community wide standards for the criteria underlying such characterizations and the systems that implement those criteria.
  - The digital preservation community needs to come to consensus on the criteria for determining format well-formedness and validity. These criteria need to be publicly accessible through mechanisms such as the LoC's Digital Formats website <http://www.digitalpreservation.gov/formats/> or the proposed Global Digital Format Registry (GDFR) <http://hul.harvard.edu/gdfr/>.
  - Common tools that incorporate these format criteria, such as JHOVE or the National Library of New Zealand (NLNZ) Metadata Extraction Tool <http://www.natlib.govt.nz/en/whatsnew/4initiatives.html#extraction>, need to be developed and widely deployed.
  - The Audit handler should reference individual files by their relative pathnames and then define the absolute pathname of the current working directory.

- HUL Digital Repository Service
  - The DRS should integrate JHOVE into its ingest workflow. Digital objects that are not well-formed or that have inconsistencies between their internal characteristics and accompanying external metadata should not be accepted.
  - The DRS should accept arbitrary digital objects, not only those created through known workflows, in a small set of approved formats, and accompanied by reliable technical metadata. Note, however, that the use of particular formats or the absence of technical metadata may limit the range and viability of preservation services.
  - The DRS should not require any subset of technical metadata. The current requirement for some technical metadata for various media types, e.g., images, audio, forces some valid objects, about which some, though not all required properties are known, to be accepted only as opaque objects about which no metadata is stored.
  - The DRS should provide a mechanism for storing unverifiable, and possibly, incorrect metadata. Even purported technical metadata provides useful descriptive information about the prior context in which digital objects were used.
  - The DRS should provide expanded facilities for storing provenance metadata about the objects under its managed care.

Appendix A Software products

The following software products were utilized during the AIHT project:

• WinZip 8.0 (3105)	<http://www.winzip.com/>
• PentaZip	<http://www.pentazip.com/pw/Compression.htm>
• WS_FTP Professional 4.50 97.05.19	<http://www.ipswitch.com/>
• GNU gzip 1.3	<http://www.gzip.org/>
• GNU tar 1.3.19	<http://www.gnu.org/software/tar/>
• McAfee VirusScan Enterprise 7.1.0	<http://www.mcafeesecurity.com/>
• MySQL 3.22.32	<http://www.mysql.com/>
• Perl v5.8.0	<http://www.perl.org/>
• JHOVE 1.0 (beta 3)	<http://hul.harvard.edu/jhove/>
• HUL Digital Repository Service	<http://hul.harvard.edu/ois/systems/drs/>
o Oracle 9i	<http://www.oracle.com/technology/software/products/oracle9i/>
o Java 1.4.2	<http://java.sun.com/j2se/>
o Clariion	<http://www.emc.com/products/systems/clariion.jsp>
o Legtato	<http://www.legato.com/products/networker/>
• Aware JPEG2000 SDK 3.6.0	<http://www.aware.com/products/compression/jpeg2000.html>
• IrfanView 3.95	<http://www.irfanview.com/>
• Adobe Photoshop CS2	<http://www.adobe.com/products/photoshop/>

Appendix B

The format of the flat file exported from the restored MySQL database, dbases/mysql/dump/lc911digitalarchive.sql, was:

```
id "title" md5 "mime" size "type" "consent"
```

where

Field	Table	Column	
id	OBJECTS	OBJECT_ID	File ID
title	OBJECTS	OBJECT_TITLE	File name (may be changed from original)
md5	OBJECTS	OBJECT_MD5_CHECKSUM	MD5 checksum
mime	OBJECTS	OBJECT_MIME_TYPE	MIME type
size	OBJECTS	OBJECT_SIZE	File size
type	OBJECT_TYPE	OBJECT_TYPE_NAME	Level of consent
consent	CONSENTS	CONSENT_NAME	Media type

```
1 "/websites/…/wetc5.jpg"         a7…05 "image/jpeg" 200704 "UNKNOWN" "REVIEW"
2 "/websites/…/WTC1.jpg"          48…39 "image/jpeg" 217088 "UNKNOWN" "REVIEW"
3 "/websites/…/wtccardinal22.jpg" 0e…37 "image/jpeg" 167936 "UNKNOWN" "REVIEW"
...
```

(File pathnames and MD5 checksums partially elided for purposes of formatting.)

Although this file has 57492 records the range of file ID's was from 1 to 57498. An examination revealed that the file ID range was not continuous; the following ID's were not used: 12527, 12529, and 35972 through 35975.

Appendix C JHOVE enhancements

The following major enhancements were made to JHOVE for the two beta releases occurring during the course of the AIHT project. Complete lists are available on the JHOVE web site <http://hul.harvard.edu/jhove/>.

- JHOVE 1.0 (beta 3), 2005-02-04
  - Simplified API intended to facility the embedding of JHOVE functionality into other systems
  - Logging API
  - HTML module support versions 3.2, 4.0, 4.01, and XHTML 1.0 and 1.1
  - Multiple files, URIs, or directory names accepted in the command-line invocation syntax; directories are processed in a breadth-first traversal
  - PDF 1.5 and 1.6 now supported by the PDF module
  - DNG profile (Adobe digital negative) now supported by the TIFF module
  - Audio properties reported according to the AES-X098B schema version 1.03b
  - New human-readable dump utilities for GIF and JPEG formats
  - Bug correction to checksum calculation
  - Bug correction to JPEG module parsing of EXIF metadata
- JHOVE 1.0 (beta 2), 2004-07-19
  - Multiple files or URIs accepted in command-line invocation syntax
  - Folder drag-and-drop now supported in the Swing client
  - New modules for AIFF, including the AIFF-C profile
  - New module for JPEG 2000, including the JP2 and JPX profiles
  - New module for WAVE, including the BWF format
  - Audio properties reported according to AES-X098B schema version 1.02

Appendix D JHOVE Audit output handler

The JHOVE Audit handler performs a breadth-first traversal of a file system and produces output similar to the following example.

```
<?xml version="1.0" encoding="UTF-8"?>
<jhove xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns="http://hul.harvard.edu/ois/xml/ns/jhove"
       xsi:schemaLocation="http://hul.harvard.edu/ois/xml/ns/jhove
                 http://hul.harvard.edu/ois/xml/xsd/jhove/1.3/jhove.xsd"
       name="Jhove" release="1.0 (beta 3)" date="2005-02-04">
 <date>2005-03-22T13:23:39-05:00</date>
 <audit>
  <file mime="text/plain; charset=US-ASCII" status="valid" md5="27...9a">
                        C:\Program Files\jhove\examples\ascii\control.txt</file>
  <file mime="text/plain; charset=US-ASCII" status="valid" md5="d9...2a">
                 C:\Program Files\jhove\examples\ascii\cr-lf-and-crlf.txt</file>
   ...
  <file mime="text/plain; charset=US-ASCII" status="valid" md5="35...b3">
         C:\Program Files\jhove\examples\xml\external-unparsed-entity.ent</file
  <file mime="text/plain; charset=US-ASCII" status="valid" md5="e5...73">
                   C:\Program Files\jhove\examples\xml\valid-external.dtd</file>
 </audit>
</jhove>
<!-- Summary by MIME type:
application/octet-stream: 6 (6,0)
application/pdf: 5 (3,2)
image/gif: 3 (3,0)
image/jpeg: 2 (2,0)
image/tiff: 22 (20,2)
image/tiff-fx: 2 (2,0)
text/html: 2 (0,2)
text/plain; charset=US-ASCII: 24 (24,0)
text/plain; charset=UTF-8: 1 (1,0)
Total: 67 (61,6)
-->
<!-- Summary by directory:
C:\Program Files\jhove\examples: 0 (0,0) + 0,0
C:\Program Files\jhove\examples\ascii: 6 (6,0) + 0,0
C:\Program Files\jhove\examples\gif: 4 (4,0) + 0,0
C:\Program Files\jhove\examples\jpeg: 3 (3,0) + 0,0
C:\Program Files\jhove\examples\pdf: 5 (5,0) + 0,0
C:\Program Files\jhove\examples\pdf\ddap: 3 (1,2) + 0,0
C:\Program Files\jhove\examples\tiff: 4 (4,0) + 0,0
C:\Program Files\jhove\examples\tiff\badfiles: 3 (3,0) + 0,0
C:\Program Files\jhove\examples\tiff\ddap: 4 (4,0) + 0,0
C:\Program Files\jhove\examples\tiff\exif: 1 (1,0) + 0,0
C:\Program Files\jhove\examples\tiff\geotiff: 9 (7,2) + 0,0
C:\Program Files\jhove\examples\tiff\geotiff\noaa: 0 (0,0) + 0,0
C:\Program Files\jhove\examples\tiff\libtiff_v3: 20 (18,2) + 0,0
C:\Program Files\jhove\examples\utf-8: 2 (2,0) + 0,0
C:\Program Files\jhove\examples\xml: 3 (3,0) + 0,0
Total: 67 (61,6) + 0,0
-->
<!-- Elapsed time: 0:00:21 -->
```

The numbers reported for the MIME type summary are the total number of files of the given type, and parenthetically, the number of well-formed files and the number of well-formed and valid files. Format well-formedness is a syntactic property, while validity is a higher-order semantic property. The JHOVE web site documents the criteria for well-formedness and validity for each format module, see <http://hul.harvard.edu/jhove/>. The summary by directory includes additional counts of malformed files and files that are not found.

The Audit handler makes use of the new output handler API introduced in the beta 3 release. The API contains callback hooks for various points in the file system traversal as illustrated in the following pseudo-code:

```
AuditState state = OutputHandler.showHeader (root directory);
for (each directory) {
  OutputHandler.startDirectory (state);
  for (each file in directory) {
    if (OutputHandler.okToProcess (file, state) {
      RepInfo info = Module.parse (file);
      OutputHandler.show (info, state);
    }
  }
  OutputHandler.endDirectory (state);
}
OutputHandler.showFooter ();
```

Appendix E GMU inventory format aggregation

The documentation of file MIME type in the GMU inventory database made use of many variant types and file extensions that were aggregated together as follows to produce the table in §1.6. The first column lists the MIME types supported by JHOVE; the second, the MIME types reported in the GMU inventory that were mapped to the JHOVE-supported MIME type; and the third, the extensions of the files.

DSIP/JHOVE MIME type	*Inventory MIME type*	*Inventory file extension*
application/octet-stream	application/octet-stream application/msword application/postscript application/x-awk application/x-dosexec application/x-empty application/x-gzip application/x-not-regular-file application/x-zip audio/mpeg audio/unknown\t audio/unknown\t video/x-msvideo image/bmp image/png image/x-3ds video/mpeg video/quicktime\tmoov No MIME type reported	.adp .api .apl .asf .asp .avi .barto .bjdate .bmp .cdda .cfm .chm .com .crumb .cxt .dat .db and many others
application/pdf	application/pdf	.pdf
audio/x-aiff	audio/x-aiff\t	.aiff
audio/x-wave	audio/unknown\t audio/x-wav\t	.wav
image/gif	image/gif	.gif
image/jpeg	imge/jpeg	.jpg, .jpeg
image/tiff	image/tiff	.tif, .tiff
text/html	text/html text/html; charset=iso-8859-1 text/html; charset=unknown text/html; charset=us-ascii	.cgi .htm .html .jsp
text/plain	message/news\t8bit message/rfc822\t7bit text/plain, English; charset=iso-8859-1 text/plain, English; charset=unknown text/plain, English; charset=us-ascii text/plain, English; charset=utf-8 text/plain; charset=iso-8859-1 text/plain; charset=us-ascii text/rtf text/x-asm; charset=us-ascii text/x-c++; charset=iso-8859-1 text/x-c++; charset=unknown text/x-c++; charset=us-ascii text/x-c; charset=iso-8859-1 text/x-c; charset=us-ascii text/x-mail; charset=us-ascii text/x-news; charset=iso-8859-1 text/x-news; charset=us-ascii	.css .eml .js .log .lst .nes .out .php .pl .plain .story .text .txt

Appendix F DSIP metadata crosswalk

The following crosswalk defines the high-level mappings between JHOVE representation information, DRS control file (`batch.xml`) markup, and internal DRS metadata table fields.

JHOVE format	**Ext**	**MIME type**	**DRS format**	**DRS metadata type**	**DRS metadata table**	**DRS app, text metadata descriptor**
AIFF	aif	audio/x-aiff	*AIFF*	AUDIO	audioMetadata	
ASCII	tdf	text/plain	TDF	TDF	tdfMetadata	
	*		TEXT	TEXT	textMetadata	*UNKNOWN*
Bytestream	rrd	application/x-esri-pyramid-file	APP	APP	appMetadata	ESRI_PYRAMID_ FILE
	aux	application/x-esri-statistics-file				ESRI_STATISTICS_ FILE
	r	application/x-sonic-waveform-reduction				WAVEFORM_ REDUCTION
	gpk	application/x-wavelab-waveform-reduction				
	*	*application/octet-stream*				*UNKNOWN*
GIF	gif	image/gif	GIF	IMAGE	imageMetadata	
HTML	*htm*	*text/html*	*HTML*	TEXT	textMetadata	*HTML*
ICC	icm	application/x-icc	ICC	APP	appMetadata	COLOR_PROFILE
JPEG	jpg	image/jpeg	JPEG	IMAGE	imageMetadata	
JPEG 2000	jp2	image/jp2	*JPEG2000*			
	jpf	image/jpx				
PDF	*pdf*	*application/pdf*	PDF	APP	appMetadata	*PDF*
PhotoCD	pcd	image/x-photo-cd	*PCD*	IMAGE	imageMetadata	
RealAudio	rm	audio/x-pn-realaudio	*REALAUDIO*	AUDIO	audioMetadata	
SGML	sgm	text/sgml	TEXT	TEXT	textMetadata	*SGML*
TIFF	tif	image/tiff	TARGET	TARGET	imageMetadata	
			TIFF	IMAGE		
UTF-8	*	text/plain	TEXT	TEXT	textMetadata	*UNKNOWN*
WAVE	wav	audio/x-wave	*WAVE*	AUDIO	audioMetadata	
XML	xml	text/xml	*XML*	TEXT	textMetadata	
Italics are used to indicate values that were enhancements made during the AIHT project.						

The following tables define the type specific crosswalks between internal technical properties, JHOVE output, and the DSIP-created `batch.xml` control file. The `batch.xml` elements are described in the order in which they must appear to satisfy the constraints of the DTD.

F.1 App

The App metadata type is for opaque digital objects with minimal technical properties.

*			**DSIP batch.xml**
"COLOR_PROFILE" "ESRI_PYRAMID_FILE" "ESRI_STATISTICS_FILE" "*PDF*" "WAVEFORM_REDUCTION"	if MIME type is	application/x-icc application/x-esri-pyramid-file application/x-esri-statistics-file application/pdf application/x-waveform-reduction	descriptor^{M}
"*UNKNOWN*"	otherwise		
	creator		creator

^{M} Mandatory

F.2 Audio

AIFF (Chunk)		***RealAudio***	***WAVE (Chunk)***		***JHOVE AES-X098B output***	***DSIP batch.xml***
(*see F.2.2 below*)			(*see F.2.3 below*)			duration ^{M}
sampleSize	(COMM)		wBitsPerSample	(Format)	bitDepth	bitdepth ^{M}
sampleRate	(SSND)		samplesPerSec	(Format)	sampleRate	samplerate ^{M}
numChannels	(COMM)		wChannels	(Format)	numChannels	channels ^{M}
"PCM"			"PCM"		audioDataEncoding	dataformattype ^{M}
"0"			"1"		byteOrder	dataorientation ^{M}
						channelmap ^{M} *
						codec
						offset
						blocksize
						firstvalidbyte
						lastvalidbyte
					wordSize	wordsize
						timestampend
						timestampstart

^{M} Mandatory

* See § F.2.1 for the construction of the `<channelmap>` element

F.2.1 *Channelmap*

The form of the `<channelmap>` element structure is dependent upon the number of audio channels. Each `<channelmap>` element contains one `<soundfield>` element and *n* `<channelassignment>` elements, where *n* is the number of channels. Each `<channelassignment>` element contains one `<channelnumber>` and one `<maplocation>` element.

channels	***channelmap***

1	`<channelmap>`
	`  <soundfield>mono</soundfield>`
	`  <channelassignment>`
	`    <channelnumber>1</channelnumber>`
	`    <maplocation>left</maplocation>`
	`  </channelassignment>`
	`</channelmap>`

2	`<channelmap>`
	`  <soundfield>stereo</soundfield>`
	`  <channelassignment>`
	`    <channelnumber>1</channelnumber>`
	`    <maplocation>left</maplocation>`
	`  </channelassignment>`
	`  <channelassignment>`
	`    <channelnumber>2</channelnumber>`
	`    <maplocation>right</maplocation>`
	`  </channelassignment>`
	`</channelmap>`

3	`<channelmap>`
	`  <soundfield>surround</soundfield>`
	`  <channelassignment>`
	`    <channelnumber>1</channelnumber>`
	`    <maplocation>left</maplocation>`
	`  </channelassignment>`
	`  <channelassignment>`

	<channelnumber>2</channelnumber>
	    <maplocation>right</maplocation>
	  </channelassignment>
	  <channelassignment>
	    <channelnumber>3</channelnumber>
	    <maplocation>center</maplocation>
	  </channelassignment>
	</channelmap>

4	<channelmap>
	  <soundfield>surround</soundfield>
	  <channelassignment>
	    <channelnumber>1</channelnumber>
	    <maplocation>left</maplocation>
	  </channelassignment>
	  <channelassignment>
	    <channelnumber>2</channelnumber>
	    <maplocation>right</maplocation>
	  </channelassignment>
	  <channelassignment>
	    <channelnumber>3</channelnumber>
	    <maplocation>left_rear</maplocation>
	  </channelassignment>
	  <channelassignment>
	    <channelnumber>4</channelnumber>
	    <maplocation>right_rear</maplocation>
	  </channelassignment>
	</channelmap>

F.2.2 *AIFF duration*

The duration of an AIFF file can be derived from the AES sampleRate value and the JHOVE RepInfo "SampleFrames" property (which is not reported in the AES object). "SampleFrames" is a LONG property. It can be recovered from a RepInfo object as follows:

```
RepInfo info;
Property prop = info.getByName ("SampleFrames");
long sampleFrames = ((Long) prop.getValue ()).longValue ();
```

The duration (in seconds) is derived by dividing the number of sample frames by the sample rate (frames/sec):

$$duration = sampleframes / samplerate$$

The `batch.xml <duration>` element value is formatted as:

`hh|mm|ss|ff`

where *`hh`* represents hours (00-99), *`mm`* represents minutes (00-59), *`ss`* represents seconds (00-59), and *`ff`* represents frames (00-29).

F.2.3 *WAVE duration*

The duration of a WAVE file can be derived from the AES sampleRate value and a calculated number of sample frames derived from the JHOVE RepInfo "BlockAlign" and "DataLength" properties (which are not reported in the AES object). "BlockAlign" is an INTEGER property; "DataLength" is a LONG property.

The number of sample frames is derived by dividing the data length by the block alignment. The duration (in seconds) is derived by dividing the number of sample frames by the sample rate (frames/sec):

sampleframes = datalength / blockalign
duration = sampleframes / samplerate

The `batch.xml <duration>` element value is formatted as:

`hh|mm|ss|ff`

where *`hh`* represents hours (00-99), *`mm`* represents minutes (00-59), *`ss`* represents seconds (00-59), and *`ff`* represents frames (00-29).

F.3 ICC

ICC metadata element requires no additional properties.

F.4 Image

					JHOVE	***DSIP***
GIF	***JPEG***	***JPEG 2000***	***PhotoCD***	***TIFF***	***NISO Z39.87 output***	***batch.xml***
Pixel		BPC		BitsPerSample	BitsPerSample	bitspersample ^{M}
"5"	"6"	"65001"		Compression	CompressionScheme	compression ^{M}
"2"	"6"	colr		PhotometricInterp	ColorSpace	photointerp ^{M}
	Xdensity	HR		XResolution	XSamp lingFrequency	xres
	Ydensity	VR		YResolution	YSamplingFrequency	yres
	units			ResolutionUnits	SamplingFrequencyUnit	resunit
						qualitylayers
						reslevels
ImageWidth		WIDTH		ImageWidth	ImageWidth	imagewidth
ImageHeight		HEIGHT		ImageLength	ImageLength	imageheight
"1"	"1"			Orientation	Orientation	orientation
						targetnotes
						history
						source
				Make Model	ScannerManufacturer ScannerModelName ScannerModelNumber ScannerModelSerialNo DigitalCameraManufacturer DigitalCameraModel ScanningSoftware ScanningSoftwareVersionNo	system
				Artist	ImageProducer	producer
						optres
				Software	ProcessingSoftwareName ProcessingSoftwareVersion	prosoftware
						enhancements
						methodology
^{M} Mandatory						

F.5 TDF (Target Definition File)

TDF metadata requires no additional properties.

F.6 Text

ASCII	***HTML***	***UTF-8***	***XML***	***DSIP batch.xml***
"US-ASCII"	"Unicode"	"Unicode"	"Unicode"	characterrep ^{M}
"ISO_646.irv:1983"	META http-equiv charset	"UTF-8"	encoding or "UTF-8"	charactermap ^{M}
				descriptor ^{M}
^{M} Mandatory				

The mandatory descriptor element should be set to "UNKNOWN" unless a specific descriptor type is known explicitly.

Appendix G Phase II export METS profile

The exported METS file was organized as follows:

```
<mets xmlns="http://www.loc.gov/METS/"
      xmlns:aes="http://www.aes.org/audioObject"
      xmlns:app="http://hul.harvard.edu/ois/xml/ns/drs/app"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:mix="http://www.loc.gov/mix/"
      xmlns:tcf="http://www.aes.org/tcf"
      xmlns:txt="http://www.loc.gov/METS/text/"
      xmlns:xlink="http://www.w3.org/TR/xlink"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.loc.gov/METS/
                          http://www.loc.gov/standards/mets/mets.xsd"
      LABEL="AIHT export from HUL DRS" TYPE="DIP"
      PROFILE="HUL-AIHT-DIP">
      ...
```

where the following namespaces are defined for global use throughout the METS file:

Prefix	***Namespace***
`aes`	AES X098B audio technical metadata schema
`app`	Harvard schema for opaque object technical metadata
`dc`	Dublin core descriptive metadata schema
`mix`	MIX schema for NISO Z39.87 still image technical metadata
`tcf`	Harvard schema for AES31-3 Time Code Format (TCF)
`txt`	METS text extension schema
`xlink`	W3C XLink schema
`xsi`	W3C XML Schema instance object schema

```
...
<metsHdr CREATEDATE="yyyy-mm-ddThh:mm:ss-zzzz">
  <agent ROLE="DISSEMINATOR" TYPE="ORGANIZATION">
    <name>Harvard University Library</name>
  </agent>
</metsHdr>
...
```

Descriptive information about the collection as a whole is provided in Dublin Core metadata (version 2002-12-12, <`http://dublincore.org/schemas/xmls/simpledc20021212.xsd`>) in the descriptive metadata section:

```
...
<dmdSec ID="D0">
  <mdWrap MIMETYPE="text/xml" MDTYPE="DC">
    <xmlData>
      <dc:title>AIHT export from HUL DRS</dc:title>
      <dc:description>An export of the 57,450 files in the AIHT test
                      corpus as originally ingested from the LC
                      distribution.</dc:description>
      <dc:subject>9/11; 9-11; September 11, 2001; New York; World
                  Trade Center; WTC</dc:subject>
      <dc:type>Collection</dc:type>
    </xmlData>
```

```
      </mdWrap>
    </dmdSec>
    ...
```

A general rights statement applicable to the collection as a whole is provided in Dublin Core metadata in the administrative metadata section:

```
    ...
    <amdSec>
      <rightsMD ID="R0">
        <mdWrap MIMETYPE="text/xml" MDTYPE="DC">
          <xmlData>
            <dc:rights>No IPR clearance has been obtained for the AIHT
                       test corpus. None of this material can be
                       distributed outside of the scope of the AIHT
                       project.</dc:rights>
          </xmlData>
        </mdWrap>>
      </rightsMD>
      ...
```

Technical metadata for each file in the collection is provided in the administrative metadata section:

```
      ...
      <techMD ID="Tn">
        <mdWrap MIMETYPE="text/xml" MDTYPE="mdtype" OTHERMDTYPE="other">
          <xmlData>
            namespace-qualified technical metadata
          </xmlData>
        </mdWrap>
      </techMD>
      ...
    </amdSec>
```

where *n* is the ordinal position of this file in the export: 1–57,450.

The following technical metadata schemata are used:

MDTYPE	***OTHERMDTYPE***	***Schema***
NISOIMAGE		MIX schema for NISO Z39.87 still image metadata
OTHER	AES-X098B	Draft AES schema for audio technical metadata
OTHER	HUL-DRS-APP	Harvard schema for opaque objects
OTHER	METS-TEXT	Proposed METS extension schema for text metadata

The export uses MIX schema (draft version 0.2, <`http://www.loc.gov/standards/mix/mix.xsd`>) to report technical metadata for all stiff image formats.

The draft AES-X098B schema is under development by the AES working group SC-03-06, Working Group on Digital Library and Archive Systems.

The HUL opaque object schema defines a single element:

```
  <app:descriptor>descriptor</app:descriptor>
```

where ***descriptor*** is a general characterization of the file type, if known. The two descriptor values found in the export are "PDF" and "UNKNOWN".

The METS text extension schema (version 2.2, <http://dlib.nyu.edu/METS/textmd.xsd>) defines the following element structure:

```
<txt:textmd>
  <txt:character_info>
    <txt:charset>charset</txt:charset>
    <txt:byte_size>size</txt:byte_size>
  </txt:character_info>
</txt:textmd>
```

where *charset* and *size* are:

charset	***size***
ISO_646.irv:1983	1
UTF-8	variable

A full manifest of all collection files, including MIME type, size, and MD5 checksum, is provided in the <fileSec>:

```
  ...
  <fileSec>
    <fileGrp AMDID="R0" USE="ARCHIVAL">
      <file ID="Fn" MIMETYPE="mime" SEQ="n" SIZE="size" ADMID="Tn"
            CHECKSUM="md5" CHECKSUMTYPE="MD5" OWNERID="id">
        <FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="href"/>
      </file>
      ...
    </fileGrp>
  </fileSec>
  ...
```

where *n* is the ordinal position of this file in the export: 1–57,450, and *href* is a URL of the form:

```
file:///aiht/data/yyyy/mm/dd/n/pk.ext
```

The <file> element ADMID attribute is an IDREF to the relevant technical metadata in the administrative metadata section. The OWNERID attribute specifies the file name as originally received in the LoC distribution.

The <structMap> section contains a single collection-level <div> that enumerates all of the files in sequence with IDREF pointers to the relevant section in the file inventory:

```
  ...
  <structMap TYPE="PHYSICAL">
    <div ORDER="1" DMDID="D0" ADMID="R0">
      <fptr FILEID="Fn"/>
      ...
    </div>
  </structMap>
</mets>
```

where *n* is the ordinal position of this file in the export: 1–57,450.

The file contains a trailing comment providing summary statistics for the export operation:

```
    <!-- Export 1.0 (yyyy-mm-dd) summary statistics yyyy-mm-dd1Thh:mm:ss-zzzz -->
    <!-- Exported n files from DRS with PK's k1 to kn -->
    <!-- Elapsed time: hh:mm:ss -->
```

Appendix H Phase III JPEG 2000 decomposition levels

The values of the maximum pixel dimension categories were based on the formula:

$$\textit{maximum pixel dimension} = \lceil \exp(i) / 150 \rceil, \text{ for } i = 1,2,3,\ldots$$

where $\exp(x)$ is the exponential function, i.e., the inverse of the natural logarithm, $\exp(\ln(x)) = x$;
$\lceil x \rceil$ is the ceiling function (the smallest integer not less than x); and
150 was selected as the nominal “thumbnail” size.

Number of levels, n	n*ln(2)	exp(n*ln(2))	é150*exp(n ln(2))ù
1	0.693147	2	300
2	1.386294	4	600
3	2.079442	8	1200
4	2.772589	16	2400
5	3.465736	32	4800
6	4.158883	64	9600
7	4.852030	128	19200
8	5.545177	256	38400
9	6.238325	512	76800
10	6.931472	1024	153600

This is based on the formula supplied by Aware for calculating the number of decomposition levels, n:

$$n = \ln(d / t) / \ln(2)$$

where d is the maximum pixel dimension of the image and t is the maximum pixel dimension of the thumbnail image.

## Appendix I	Phase III JPEG 2000 transformation

The JPEG 2000 transformations were performed using the C SDK 3.6.0 codec from Aware, Inc. <www.aware.com/ products/compression/jpeg2000.html>. This version of the codec does not support GIF as a source format. Therefore, all GIF files were transformed into uncompressed RGB TIFF files using ImageMagick 6.1.9 <www.imagemagick.org/>, which incorporates Sam Leffler's LibTIFF 3.7.1 library <www.remotesensing.org/libtiff/>.

```
% gunzip  tiff-3.7.1.tar.gz
% tar xvf tiff-3.7.1.tar
% cd tiff-3.7.1
% ./configure –prefix./projects
% make
% make install

% gunzip  ImageMagick-6.1.9-4.tar.gz
% tar xvf ImageMagick-6.1.9-4.tar
% cd ImageMagick-6.1.9
% ./configure –prefix=./projects --disable-shared \
  --without-magick-plus-plus --disable-installed \
  --with-quantum-depth=8 --without-perl \
  --without-x CPPFLAGS=-I./include \
  LDFLAGS='-L./projects/lib -R/./projects/lib'
% make
% make install
```

The GIF-to-TIFF transformation was invoked as:

```
% convert -colorspace RGB –compress None file.gif file.tif
```

and was verified using IrfanView 3.95 <www.irfanview.com/> for side-by-side GIFF/TIFF comparisons.

NOTE	The following GIF file failed the conversion process:

```
convert: Corrupt image '/drstestdata/drs/aiht/data/2004/12/20/47/33943.gif'
```

This file was excluded from the transformation process.

The command-line invocation for the codec was:

```
% j2kdriver –set-input-image-type type file.ext –p RLCP \
  --tile-size 1024 1024 –w R53 levels –y 2 \
  --set-output-j2k-layer-psnr 0 35 –-set-output-j2k-layer-psnr 1 0 \
  -q ALL REVERSIBLE --predictor-offset 0 –t JP2 –o file.jp2
```

where

Max. pixels	***levels***
0 – 300	1
301 – 600	2
601 – 1200	3
1201 – 2400	4
2401 – 4800	5
4801 – 9600	6

The maximum pixel size is the greater of the horizontal or vertical pixel dimension of the source image. For more information on the algorithm used to define these decomposition levels see Appendix H.