Preparing for the Future as We Build Collections

Jody L. DeRidder; University of Alabama; Tuscaloosa, Alabama.

Abstract

Producing online digital collections is only the tip of the iceberg, and many administrators are unaware of the need for preparation of content for long term access. When developing work flows and procedures for online delivery, we often overlook critical choices that could severely impact our ability to prepare our content for long-term availability. This paper highlights primary considerations beyond the standards selected for descriptive metadata or digital capture, and describes work flow processes for adapting typical CONTENTdm or other collections for long-term preservation.

Concern for the future

Most administrators are satisfied with the online appearance of digital collections, unaware of the fact that this is only the beginning. Preparation for the support of long-term access goes far beyond the immediate gratification of current delivery, but often requires forethought to prevent mass chaos when dealing with legacy content. A hundred years from now, digital archivists seeking to gain access to the content we create today, will need to be able to understand how to recompose our complex objects, locate the metadata associated with each one, identify component parts, and reconstruct as much of the provenance and rights as possible. Many of our cultural heritage institutions are hampered by the lack of awareness of simple steps they can take today to prepare for the future. Even those without programming staff can benefit from looking ahead and making choices now which may later enable reconstruction of their precious digital content.

When developing work flows and procedures for online delivery, it is easy to overlook critical choices that could severely impact our ability to prepare our content for long-term availability. Many publications address the standardization of digital capture and metadata (an excellent resource for locating this information is provided by NISO [1]). This paper highlights primary considerations beyond these, and describes work flow processes for adapting typical CONTENTdm [2] collections for long-term preservation. Delivery software will come and go, so planning the organization and storage of archival content is necessary for the future. While certain aspects of the discussion are tailored to CONTENTdm, the primary concerns addressed are applicable to most, if not all, delivery software and digitization work flows. Looking ahead in order to make intelligent decisions that will save time and money is critical to success and sustainability in digital library development.

Common Problems

The first few years of a digital library's development are often marked by "boutique collections," in which each metadata scheme and the attendant display labels vary to suit the specific materials, often with little regard for consistency between the collections. Original archival bit stream file names are often completely overlooked in the descriptive metadata; digital rights and terms of use specifications are often haphazard and piecemeal, patterned after those seen elsewhere. There may be no provenance metadata kept, no standard specification of the original or digital format, and no record kept of how the files themselves were created, all of which may be invaluable technical metadata for future generations. Early video captures, particularly, may undergo a number of reformatting transformations with little or no trail to mark their path or even to describe the original artifacts.

Recognizing these problems as soon as possible enables reconstruction of missing data, and prevention of additional loss. If your institution has the capacity and wherewithal to develop a full-fledged digital preservation policy, an excellent set of guidelines has been provided by JISC [3]. An overview of the kinds of information that will be needed by future archivists is described by Moore [4], and includes complete documentation of policies and procedures, and persistent names for identifying the records, archivists, and the repositories. Here we will consider organization of content and metadata, as a base layer of sanity to enable reuse in subsequent software, as well as laying the foundation for access by archivists in the future.

Choices from the beginning

The adoption of a data dictionary from the beginning to define the scope, patterning, controlled vocabulary, and standardization of the use of each metadata field in use, will prove invaluable in the long run and prevent headaches in later years. If anomalies in metadata are agreed upon, they must be tracked, and this is a nightmare to be avoided whenever possible, as they severely impact cross-walking and future reuse particularly in federated searching capacities. Whatever rights or provenance metadata can be collected should be, while the possibility of gathering it is still available. All of these should be stored in Unicode or ASCII text (standardized XML, if possible) in the collection or repository level archival directory to which it applies. When trying to determine what rights to document and how best to do so in a standardized manner, an excellent resource to reference, if not apply, is the Open Digital Rights Language [5], which covers usage, reuse, transfer and asset management. More recent standards development in the digital library realm for documentation of both rights management and provenance, is delineated in PREMIS [5].

One of the first considerations in delivery software, prior to use, should be the extent and form of exports available. If the metadata cannot be re-exported in a standardized format that meets the needs of preservation, then copies of the original metadata must be transformed into those formats and retained. Beware of the loss of metadata added to the content after import into the software; it's a waste of time to provide added-value metadata if it cannot be retained beyond the present delivery system. Examining

the batch exports from the beginning to verify completeness, accuracy, quality of output and method of treatment for non-ASCII characters can deeply inform the beginning digital librarian. Testing these exports for validation against widely-accepted schemas, or even against the locally-defined data dictionary, may reveal a number of problems. If these are left untended, they could later require programming skill to correct, or many hours of metadata librarian effort to normalize, when combined with other collections.

Critical considerations and consistency

A critical aspect that is often overlooked when creating a new collection is the necessity of retaining machine-processable information in a metadata field which will serve to reunite the descriptive metadata with the archival bit streams. Delivery software often renames content and metadata upon upload to sequential numbers according to its own needs. If the exported metadata does not include the original bit stream file names in a consistent field, the exports will be almost useless. Before uploading content to any delivery system, select a specified field in which to store this value, and hold it standard across all collections regardless of format or delivery method or software. Often this will be an identifier field of some sort, but care must be taken to select a field which will not collide with other needs as the digital library develops, such as persistent URLs. As file naming may have to be altered for upload into delivery systems, the original file name must be retained within the metadata, and, at minimum, an ASCII text flat file stored providing match-up lists between old and new file names, for later identification.

File names and organizing content

File naming of archival bit streams and metadata should be consistent across all collections, to ensure the uniqueness of names, machine processability, and scalability for the future. While selecting words which represent the collection name as the first part of a file name seems helpful, it will not scale, as soon collisions occur when similarly named collections appear. Beware of using the file name to represent metadata which may be lost over time if not stored instead within the descriptive record. Consider what information actually needs to be part of the file name if this file is combined with many thousands of other files in a digital storage repository. Adoption of the XML id attribute format [7] for file identifiers is recommended, as a reference to the file within an XML metadata record would require adherence to this format.

Structure of documents containing multiple bit streams must be documented in a form which is machine-processable. Metadata Encoding Transmission Standard (METS [8]) offers an excellent method of packaging the various forms of metadata about a complex object, as well as to organize the many associated files and bit streams. However, creation of METS is not a simple task, and is best automated; for institutions lacking the wherewithal to create or implement an automated method for METS record development, there are other options.

Standardized hierarchical file naming systems are a low-cost and low-tech method for organizing and structuring documents, which also supports the later creation of METS records. Minimally, a patterning of set lengths of numbers divided by underscores can specify a relation between files which compose a complex object.

For example, the 3rd photo on the 5th page of the 7th letter in the 23rd collection of repository 12 in a university could be represented by the file name u0012_000023_000007_005_003 with an appropriate extension. Here the first two sections of the file name indicate some level of provenance, which in a venue that supports the digitization of content from a variety of sources, becomes invaluable in organizing the storage of documents for later retrieval. The remainder of the structure of the file name represents the structure of the file; the finding aid for this collection may be named u0012_000023 with an appropriate extension, clearly related to the former file as a root or grandparent file. The 7th letter found within this manuscript collection would be identified digitally as u0012_000023_000007 with a proper extension, and the 5th page of that letter would clearly be u0012_000023_000007_005. Should alterations be made in the archival version of a file, the original should be retained and a record made as to the reason, type and extent of the alteration; the newer file may either be placed in a sub directory noting the version such as "Version2"; alternatively, "_v2" may be added to the file name, for example: u0012_000023_0000007_0005_v2.tif.

File names with this type of clearly delineated hierarchy enable organized storage in directories labeled for each level. This is a form of organization completely unreliant upon software, and thus far more hardy over time. As an example, a file named u0012_000023_000007_005.tif would be stored in the /u0012/000023/000007/005/ directory. All content from the university holdings location number 12 would be located in the u0012 directory, and information about this holdings location should also be stored in this directory. Likewise, all of the content in their 23rd collection would be stored in /u0012/000023/, and information about that collection should be stored at this level. Thus, at each level of the hierarchy, metadata or other documentation may exist. A file named u0012_000023_000007_mods.xml would be stored in the /u0012/000023/000007/ directory, as that is the level at which this metadata applies. Similarly, MIX technical metadata about the 5th image in this letter may be stored as u0012_000023_000007_005_mix.xml in the /u0012/000023/000007/005/ directory with all other information related to that image. It is sometimes possible to rely on the file extension to indicate the type of file; however, where multiple versions of a particular extension exist, for example a MODS and a Dublin Core record for the letter as an item, these records may be differentiated by an additional "_mods" or "_dc" before the XML extension. Otherwise, separation into different directories will be necessary, in which case these metadata directory names must be standardized as well across all holdings, to enable automated handling of records, either now or in the future.

Remediating 7train METS

Pros and cons of exports and metadata transformation tools should be considered before modifying content within delivery software. If the remediated metadata or page-level descriptions cannot be captured, then the time and money spent to provide this may be wasted. Versions of CONTENTdm prior to 5, for example, failed to export page-level metadata excepting transcriptions or OCR content, and the page label (title).

California Digital Library developed the 7train software [9] to transform CONTENTdm exports into METS files. However, these files include links into CONTENTdm to access the derivatives, which is less than helpful for archiving metadata. The links themselves are composed of the server name, the cgi script name for the type of item, a moniker denoting the current collection container, and the database number for the internal XML record for that particular page. 100 years from now, this is not likely to be useful information for an archivist seeking to restore access to the associated archival files. As soon as the server, software, collection container, or even the location within the database is changed, these links are useless. Transforming these to links denoting the location of these particular images is much more helpful, and adding the directory location to the linking of the archival tiff is also clearly of use. If the derivative files are not desired for future use, then simply deleting these file sections improves the METS file for preservation purposes.

If retaining the derivatives created by CONTENTdm on upload, one must determine which bit stream corresponds to which link, in order to link appropriately (and possibly rename the image to match the archival version, which is highly recommended). The database number from the link to the derivative in question must be matched against the XML file in /index/description/desc.all. For example, the link for an image, content.lib.xx.edu/cdm4/item_viewer.php?CISOROOT=/collname &CISOPTR=132 contains the database number 132 for the metadata record within the desc.all for the collection denoted by "collname"). Within this metadata record, the <find> tag value will identify the correct image in the image directory. Normally, within the item-level metadata record is a <fullrs> tag containing the name of the archival tiff which was originally uploaded, though this may have been modified somewhat, (another danger to beware in delivery software). In addition, for compound objects, to verify that this particular image does belong to the archival object in question, one must check the supp directory for a sub directory matching the database number. For example, if within the supp directory there exists a subdirectory by the name "132", then this is a child file of a more complex object. This subdirectory will contain an index.xml with a <parent> value denoting the parent database record number, by which the object-level metadata record can be located in desc.all. The object-level metadata in this database record is what 7train inserts into the descriptive metadata section of the METS record. With the advent of Version 5 of CONTENTdm, it is hoped that page-level metadata will be available for 7train storage as well.

Organized file storage without METS

Should the librarian be using a different software, and/or be unable to create the METS file, the descriptive metadata XML should be stored for preservation at the appropriate directory for the object being described. If using the aforementioned file naming system, for example, descriptive metadata for the 23rd collection of university repository 12 would be stored in the u0012/000023/ directory. If the descriptive metadata is for the 7th letter in that same collection, it should be stored in the u0012/000023/000007/ directory, and so forth. Thus, without the skills to create METS files or to develop and maintain databases, we have a low-tech method of systematically storing the appropriate metadata and bit streams in a way that enables recreation of the complex digital object at a later date. Note that the metadata is fairly useless without the accompanying bit stream. The original archival content must also be stored in the appropriate directory, and reference to it in a standardized field in the metadata is of fundamental importance.

Adding technical metadata

An addition to the 7train METS for preservation would be the technical metadata for each archival bit stream, obtained by running JHOVE [10] against the files and transforming the output into the desired selection of fields, according to the schema of choice. For example, selecting the mandatory and required if applicable fields from the MIX output from JHOVE's extraction from a TIFF, will create a reasonable subset to include in the appropriate part of the METS file. Lacking the ability to create METS, simply running JHOVE against each archival bit stream and storing the output, with check sums, in the correct file name directory will enable future archivists to reconstruct access to the specified bit stream. This is a low-cost and reasonably functional way to prepare digital content for preservation.

Beyond the local repository

A further step which is within reach of modestly-funded cultural heritage repositories is the incorporation of LOCKSS [11] surveillance of the established archival repository. Collaboration with other institutions both locally and at a distance safeguards the continued existence of precious digital holdings against a multitude of dangers which would prevent long term access. A side benefit of such collaborations is the heightened exchange of information and raised awareness of developing standards and options in the broader digital environment. Costs for this option are primarily for storage space, communication with other participants, and perhaps an hour or two of management per month. Implementation requires installation of the software on a separate server, the ability to create HTML pages linking archival content in web accessible directories, and selection of appropriate patterns for crawling the storage area (using the LOCKSS Plug-in Tool [12]).

Work flow steps

With a data dictionary to guide metadata field creation, a consistent field for a standardized identifier, a standardized file naming system and an archival storage system, work flow organization is the next step. As file names and identifiers (which may be derived from the same) must be unique across all holdings, a centralized tracking spreadsheet or database must be maintained. In our work flow, students check for the next sequential collection number to be assigned and enter in information about that collection (source, type, location, description, etc.) into the spreadsheet when assigning the number by which it will henceforth be known. This information is also stored in a newline-delimited pattern plain text file at the collection level for transfer to the archival storage location. Item numbers are assigned incrementally during scanning, and "page numbers" are assigned for sequential delivery patterning with no regard to the page number as indicated on the material. If programming staff is not available, students could run JHOVE on each image at the time of scanning, saving the content to files named for the image number

with an XML extension, storing this file in a sub directory of the metadata.

Metadata spreadsheets for each collection are drawn from template spreadsheets which include subsets of the complete list of potential metadata fields we allow for any collection (our metadata is standardized across all collections). A copy of the complete spreadsheet is modified to hide the columns not needed for the collection currently being digitized. We have created template spreadsheets from the master for sheet music, manuscript items, images, and audio. Students enter information about each item during the scanning process, inserting correct identifiers (derived from the file names) into the spreadsheet. Digital files are organized in folders named for the collection number with a word or two following for easy reference. Within these collection folders are standardized sub folders for administrative information, item-level metadata, transcriptions and scans. Digitized content is placed into the correct folders, with compound object scans in sub folders under a folder named for the item number within scans. The completed spreadsheet is saved in the metadata folder. For upload into CONTENTdm, the metadata spreadsheets are exported into tab-delimited text files. If complete metadata cannot be exported from delivery software in the formats desired, consider adding XML field mappings to your spreadsheet and save the content as XML, preferably after validating against a standardized metadata schema. Minimally, save as tab-delimited or comma-delimited ASCII text and store a data dictionary with it in plain ASCII text to explain the labels or column headings used.

If utilizing CONTENTdm's access to controlled vocabularies to remediate metadata, upload the content with the tiffs, and then (after modifications) export from CONTENTdm in the CONTENTdm Standard XML. Unfortunately, this currently must be done via the web and is not yet supported from command line, which impedes automation of preservation support. These exports must then be run through 7train, which can be done either by script or drag-and-drop of each file. This creates an XML file for each item, whether compound or singular, named for the first identifier field. Thus it is extremely important to have a standardized unique identifier in the first field mapped to dc:identifier in CONTENTdm for each record. Modifications to the 7train XSLT will enable the capture of qualified Dublin Core as opposed to unqualified, though there is no method to capture the page-level metadata until version 5. Programmers may be able to extract the page-level metadata from desc.all by locating the appropriate database number for the image in question, but this is a level of work for which few digital libraries using CONTENTdm have the staffing.

If removal or remediation of derivative links, or alterations of the links to the archival image are desired, these must be done prior to storing the METS with the bit streams. Now is also the time to add in the technical metadata obtained from running JHOVE if programming support is available. If not, storage of the technical metadata as well as the METS (or descriptive metadata in Unicode or ASCII text XML) should be at the level of the item in the archival storage area. Relocate the archival bit streams with all metadata to the prepared archival area, placing them into the directories which were created earlier, based on the file naming structure. Add collection-level and administrative metadata into the collection-level directories in standardized formats, preferably in Unicode or ASCII text XML. Notify other LOCKSS participants of the additional content, and ensure local backups are sufficient. A rotation of 3 weekly full backups with one off-site is recommended; incremental backups on shorter rotation are an added value. In addition, if possible, include MD5 check sum verifications of all files prior to each backup, to prevent storage of corrupt content.

Tracking the stages of this work flow per collection in another spreadsheet or database is advisable to ensure that each step is appropriately completed prior to the next, and to ensure completeness. This level of tracking also enables quality control steps to be built in to prevent later repair of what is much simpler to correct at the point of creation. In addition, tracking the type and version of archival bit stream, and the type and version of descriptive metadata have been used in each collection, will enable the location of files for format migration and metadata remediation when the time is propitious. Centralized location, and constant updating of tracking spreadsheets or databases will provide for smoother transitions, and far less chaos in dealing with legacy content, for years to come.

In summary

Thinking ahead to standardize and streamline file naming and organization makes it possible to prevent chaos when confronted with the rudimentary steps required to store content for future access. Delivery software systems come and go, and we cannot be dependent upon them to provide us with what we will need 5 years from now, much less 50 years from now. By looking ahead, it becomes clear that we need standard methods of organizing our materials for retrieval, and for associating the many metadata and bit stream files with their relationships to one another. Clarifying how to do this across multivariate collections, and building file storage systems where the directory names reflect the standardized file names builds consistency which can be leveraged at a later time by automated or even manual procedures, to reconstruct the library.

Programming staff, while extremely valuable, are not absolutely necessary to preparing content for long-term storage. The use of open source tools such as JHOVE, 7train, and LOCKSS can be extremely useful even for non-programmers. If the use of only one of these tools is possible, however, a minimal approach would be a standard file naming system, a standard organization of those files in directories that reflect them, the addition of whatever XML Unicode or ASCII metadata is available (in hopefully a current standard), and the use of LOCKSS to protect these holdings for the future. Even those of us with minimal resources at our disposal may find this a reachable goal. The future use of our content, in the next generation of delivery software and in the years to come, should be of concern from the very beginning. Planning, standardizing, streamlining, organizing, and documenting are extremely useful approaches to preparing our digital content for long-term access, even as we put them online for the very first time.

References

[1] NISO, "A Framework of Guidance for Building Good Digital Collections," 3rd ed., 2007 (framework.niso.org).

[2] OCLC, CONTENTdm Digital Collection Management Software, (www.contentdm.com).

[3] N. Beagrie, N. Semple, P. Williams, and R. Wright, "Digital Preservation Policies Study," 2008 (www.jisc.ac.uk/media/documents/programmes/preservation/jiscpolicy_p1finalreport.pdf).
[4] Reagan Moore, "Towards a Theory of Digital Preservation," Int'l Jour. of Digital Curation, 1:3, 2008 (www.ijdc.net/index.php/ijdc/article/view/63/42).
[5] W3C, Open Digital Rights Language (ODRL) Version 1.1 (www.w3.org/TR/odrl).
[6] OCLC, PREMIS Data Dictionary for Preservation Metadata, Version 2.0, March 2008 (www.oclc.org/research/projects/pmwg/premis-final.pdf).
[7] W3C, Extensible Markup Language (XML) 1.0 (Fifth Ed.), (www.w3.org/TR/html401/types.html#type-id).
[8] Library of Congress, Metadata Encoding and Transmission Standard (METS), (www.loc.gov/standards/mets).
[9] California Digital Library, 7train METS generation tool, Version 1 (seventrain.sourceforge.net, available from sourceforge.net/projects/seventrain). Copyright the University of California Regents.
[10] JSTOR/Harvard Object Validation Environment (JHOVE), (hul.harvard.edu/jhove, available from sourceforge.net/projects/jhove).
[11] Stanford University, LOCKSS (Lots of Copies Keep Stuff Safe) Program (www.lockss.org/lockss/Home).
[12] Stanford University, LOCKSS Plugin Tool (www.lockss.org/lockss/Plugin_Tool).

Author Biography

Jody L. DeRidder received her MS in Computer Science (2002) and her MS in Information Sciences (2008) from the University of Tennessee. She is currently the Head of Digital Services at the University of Alabama. Her recent work has focused on developing streamlined, standardized processes for digitization, delivery, and preparation for long term digital access. She is on the Steering Committee for the SAA MDOR, and is a member of ACRL, and ACM.