Transfer and Inventory Services in Support of Preservation at the Library of Congress

Leslie Johnston
Library of Congress
National Digital Information Infrastructure
and Preservation Program

ABSTRACT

The digital content lifecycle is generally understood as a set of activities: select, get and/or produce, prepare and/or assemble, describe, manage, and, as appropriate, make available. At the bit level, digital content is viewed as files on a file system. Many crucial activities of the digital content lifecycle are therefore undertaken primarily at the bit level, including transferring, moving, and inventorying files, and verifying that files have not changed over time. The identifiable entities at the bit-level – files and directories -- are widely and easily understood by Library of Congress digital collection data managers and curators. As part of its initial development in support pf preservation services, the Library is working on a suite of solutions to enable the activities of the digital lifecycle for files and directories. Current and planned tool and service development focus on the BagIt specification for the packaging of content; the LC Inventory System to record lifecycle events; and workflow tools that leverage both. The outcomes for the Library include the documentation of best practices, open source software releases, and support for a file-level preservation audit.

1 INTRODUCTION

For the past three years, the Library of Congress National Digital Information Infrastructure and Preservation Program (NDIIPP) and Repository Development Center have been implementing solutions for a category of activities that we refer to as “Transfer” [3, 6]. At a high level, we define transfer as including the following human- and machine-performed tasks:

- Adding digital content to the collections, whether from an external partner or created at LC;
- Moving digital content between storage systems (external and internal);
- Review of digital files for fixity, quality and/or authoritativeness; and
- Inventorying and recording transfer life cycle events for digital files.

The work on transfer has focused primarily on work with external partners, including those that are part of the National Digital Information Infrastructure and Preservation Program NDIIPP[1] [1]; the National Digital Newspaper Program (NDNP)[2] [4, 5]; the World Digital Library (WDL)[3]; and the Library’s Web Archiving initiatives.[4]

The development of transfer services is not surprisingly closely linked with bit preservation, as the tasks performed during the transfer of files must follow a documented workflow and be recorded in order to mitigate preservation risks. The goal of bit preservation is to ensure that files and their vital contextual file system hierarchies are retained intact throughout the digital life cycle.

The digital content lifecycle is generally understood as a set of activities: select, get and/or produce, prepare and/or assemble, describe, manage, and, as appropriate, make available. At the bit level, digital content is viewed as files on a file system. Many crucial activities of the digital content lifecycle are therefore undertaken primarily at the file system and bit level:

- Transferring digital files to the control of the appropriate division or project at the Library, whether from external partners or produced internally;
- Moving digital files between storage systems, including archival storage systems;
- Inventorying digital files; and
- Verifying that the digital files have not changed over time.

The identifiable entities at the bit-level – files and directories – are widely and easily understood by Library digital collection data managers and curators. What we call the **Content Transfer Services** provide a suite of tools and services to enable the activities of the digital lifecycle for files and directories. Many of the existing tools and services have been or are being extended to provide additional support for bit preservation activities.

[1] For information on NDIIPP, please see: http://www.digitalpreservation.gov/

[2] For information on NDNP, see: http://www.loc.gov/ndnp/ and http://chroniclingamerica.loc.gov/.

[3] For information on WDL, see: http://www.wdl.org/.

[4] For information on the Library’s web archiving activities, see: http://www.loc.gov/webarchiving/.

2 CURRENT SERVICE COMPONENTS

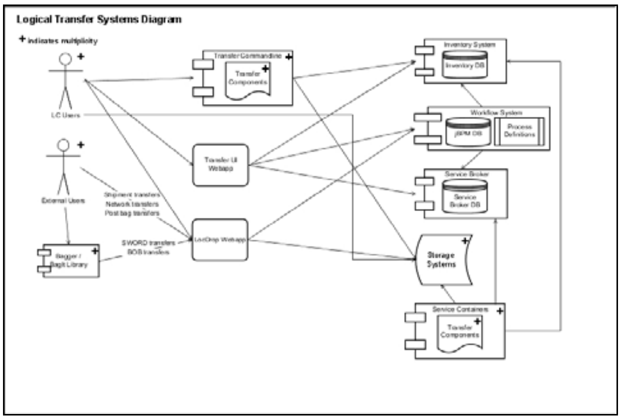
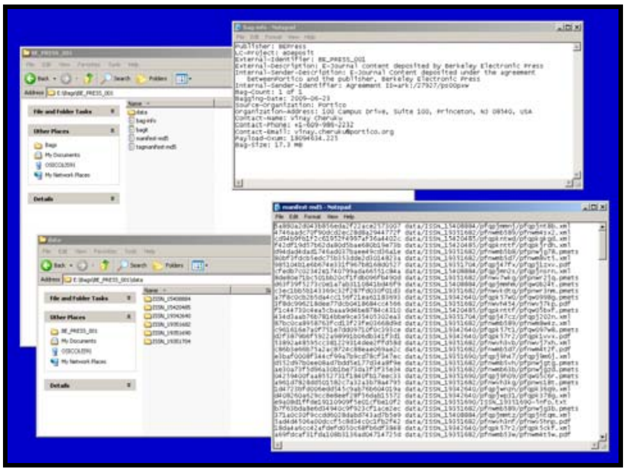


Figure1. Current Library of Congress production Content Transfer Services

BagIt is a specification for the packaging of content for movement between and within institutions. Its package-level metadata and manifest of files and fixities can aid in preservation over time.[5] The base directory of a Bag contains a bag declaration (bagit.txt), a bag manifest (manifest-*algorithm*.txt), a data directory (/data), and an optional bag information file (bag-info.txt). The bagit.txt file is a required file, and simply declares that this is a Bag, and which version of the specification it complies with. The bag-info.txt includes information on the Bag, including descriptive and administrative metadata about the package (not the package contents), as well as the bagging date and human and machine-readable Bag size.



```
Publisher: BEPress
LC-Project: adeposit
External-Identifier: BE_PRESS_001
External-Description: E-Journal content deposited by Berkeley Electronic Press
Internal-Sender-Description: E-Journal content deposited under the agreement
   betweenPortico and the publisher, Berkeley Electronic Press
Internal-Sender-Identifier: Agreement ID=ark:/27927/ps00pxw
Bag-Count: 1 of 1
Bagging-Date: 2009-06-23
Source-Organization: Portico
Organization-Address: 100 Campus Drive, Suite 100, Princeton, NJ 08540, USA
Contact-Name: Vinay Cheruku
Contact-Phone: +1-609-986-2232
Contact-Email: vinay.cheruku@portico.org
Payload-Oxum: 18094634.225
Bag-Size: 17.3 MB
```

Figure 2. A Bag with its bag-info.txt, Data Directory, and its Manifest.

The manifest lists the names and checksums of the content files; there is an additional checksum manifest for the shipping files. Any commonly recognized checksum algorithm can be used to generate the manifests, and must be identified in the name of the manifest file. The files comprising a package may be transferred in a container format such as ZIP or tar to be unpacked upon receipt. There is also the concept of a "holey" bag, which has the standard bag structure but its data directory is empty. The holey bag contains a "fetch.txt" file that lists the URLs of the content files to be fetched (so-called "holes" to be filled in). Transfer processes follow the URLs, download the files and fill the data directory. The sender's source files do not need to reside in the same directory or on the same server. The content manifest does not obviate the need for descriptive metadata being supplied by the package producer. The manifest assists in the transfer and archiving of the package as a unit, rather than supplying any description of the content.

The data directory is required, and contains the contents of the package, as defined by its producer. The data directory must always be named "/data," and may have any internal structure; there is no limit on the number of files or directories it may contain, but its size should make practical transfers easier, based on physical media limitations or expected network transfer rates. There is no limit on the number of files or directories this directory may contain, but its size should make practical transfers easier, based on physical media limitations or expected network transfer rates. In the Library's experience, 500 GB is the recommended maximum size, although Bags as large as 1.8 Tb have been transferred.

BIL is a Java library developed to support Bag services. A barrier to uptake of the BagIt specification was the inability to automate the Bagging process or support the development of tools. BIL is scriptable and can be invoked at the command line or embedded in an application. It supports key functionality such as creating, manipulating, validating, and verifying Bags, and reading from and writing to a number of formats, including zip, tar, and gzip tar. BIL also supports the uploading of Bags using the SWORD deposit protocol[6] using the Library's extension, BOB (Bag of Bits).

While BIL proved vital in the development of scripted processes, the majority of its potential users at the Library are data managers and curators who are not accustomed to working at the commandline or writing programs. A graphical desktop application for the bagging of content is nearing completion of its development and testing. **Bagger** is a Java application developed on top of BIL with Spring Rich Client[7] as the MVC framework, and a HSQLDB[8] in-memory database. It is implemented as both a Java Webstart application for use across platforms and as a standalone version with its own bundled, Java JRE and various checksum generators. Bagger is a small application, taking up less than 100 MB, and is fully self-contained and requires no

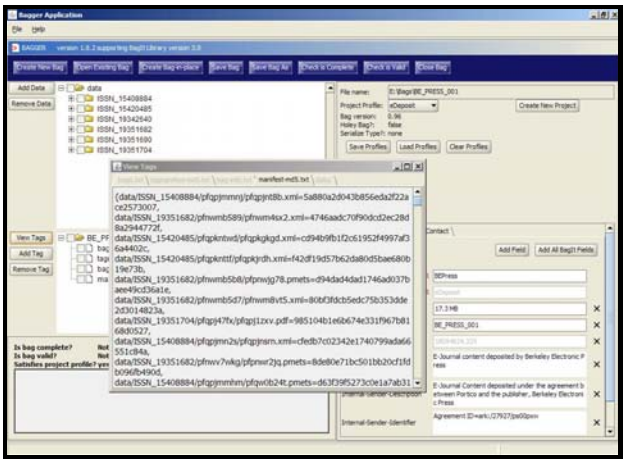
[5] The BagIt specification is available at: http://www.digitalpreservation.gov/library/resources/tools/docs/bagitspec.pdf

[6] For more information on SWORD, see http://www.swordapp.org/.

[7] http://www.springsource.org/spring-rcp

[8] http://hsqldb.org/

administrative privileges or an installer. The limits of its use are the available disk space and memory of the machine where it is used.



File name: E:\Bags\BE_PRESS_001
Project Profile: eDeposit
Bag version: 0.96
Holey Bag?: false
Serialize Type?: none

```
{data/ISSN_15408884/pfqpjmmnj/pfqpjnt8b.xml=5a880a2d043b856eda2f22ace2573007,
data/ISSN_19351682/pfnwmb589/pfnwm4sx2.xml=4746aadc70f90dcd2ec28d8a2944772f,
data/ISSN_15420485/pfqpkntwd/pfqpkgkgd.xml=cd94b9fb1f2c61952f4997af36a4402c,
data/ISSN_15420485/pfqpknttf/pfqpkjrdh.xml=f42df19d57b62da80d5bae680b19e73b,
data/ISSN_19351682/pfnwmb5b8/pfpnwjg78.pmets=d94dad4dad1746ad037baee49cd36a1e,
data/ISSN_19351682/pfnwmb5d7/pfnwm8vt5.xml=80bf3fdcb5edc75b353dde2d3014823a,
data/ISSN_19351704/pfqpj47fx/pfqpj1zxv.pdf=985104b1e6b674e331f967b8168d0527,
data/ISSN_15408884/pfqpjmn2s/pfqpjnsm.xml=cfedb7c02342e1740799ada66551c84a,
data/ISSN_19351682/pfnwv7wkg/pfpnwr2jq.pmets=8de80e71bc501bb20cf1fdb096fb490d,
data/ISSN_15408884/pfqpjmmhm/pfqw0b24t.pmets=d63f39f5273c0e1a7ab31
```

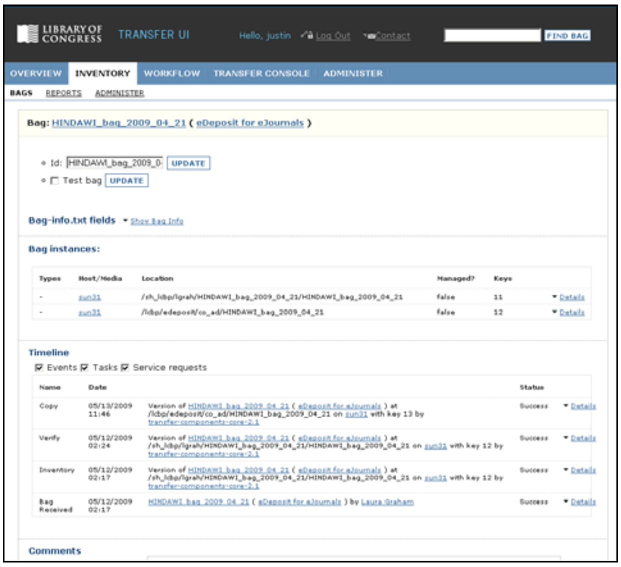
Is bag complete? Not
Is bag valid? Not
Satisfies project profile? yes

BEPress
eDeposit
17.3 MB
BE_PRESS_001
E-Journal content deposited by Berkeley Electronic Press
E-Journal Content deposited under the agreement between Portico and the publisher, Berkeley Electronic Press
Agreement ID=ark:/27927/ps00pxw

Figure 3. The Bagger Tool for Creating a Bag and its Fixities

The Library has developed utility scripts that support the BagIt specification. The **Parallel Retriever** implements a Python-based wrapper around wget and rsync, and transfers Bags and fills Bags when given a holey Bag manifest and a fetch.txt file. It supports rsync, HTTP, and FTP protocols. The **Bag Validator** Python script checks that a Bag meets the specification: that all files listed in manifest are in the data directory and that there are no duplicate entries or files that are not listed in the manifest. The **VerifyIt Shell** script is used to verify the checksums of Bag files against its manifest. These scripts and the BIL Java Library have been released as open source on SourceForge.[9] Bagger is the next tool under review for open source release.

The **Inventory System** keeps track of and enables the querying of important events in the preservation lifecycle of a Bag and its contents. Its data model is implemented using Java objects mapped to a mySQL database using Hibernate[10] for object-relational mapping. The goal in developing the Inventory Service is to satisfy needs identified through the process of doing transfers and attempting to record their outcomes as well as track the files once their enter the Library's infrastructure. These needs include keeping track of package transfers for a project, tracking individual packages and life cycle events associated with them, and a list of the files that make up each package and their locations. For legacy collections these tools can be pointed at existing directories to package, checksum, and record inventory events to bring the files under initial control. The data in the Inventory System can be used as a source to generate PREMIS metadata[11].



Bag: HINDAWI_bag_2009_04_21 (eDeposit for eJournals)

- Id: HINDAWI_bag_2009_0
- □ Test bag

Bag-info.txt fields

Bag instances:

Types	Host/Media	Location	Managed?	Keys	
-	sun31	/sh_ldbp/lgrah/HINDAWI_bag_2009_04_21/HINDAWI_bag_2009_04_21	false	11	▾ Details
-	sun31	/ldbp/edeposit/co_ad/HINDAWI_bag_2009_04_21	false	12	▾ Details

Timeline

☑ Events ☑ Tasks ☑ Service requests

Name	Date		Status	
Copy	05/13/2009 11:46	Version of HINDAWI_bag_2009_04_21 (eDeposit for eJournals) at /ldbp/edeposit/co_ad/HINDAWI_bag_2009_04_21 on sun31 with key 13 by transfer-components-core-2.1	Success	▾ Details
Verify	05/12/2009 02:24	Version of HINDAWI_bag_2009_04_21 (eDeposit for eJournals) at /sh_ldbp/lgrah/HINDAWI_bag_2009_04_21/HINDAWI_bag_2009_04_21 on sun31 with key 12 by transfer-components-core-2.1	Success	▾ Details
Inventory	05/12/2009 02:17	Version of HINDAWI_bag_2009_04_21 (eDeposit for eJournals) at /sh_ldbp/lgrah/HINDAWI_bag_2009_04_21/HINDAWI_bag_2009_04_21 on sun31 with key 12 by transfer-components-core-2.1	Success	▾ Details
Bag Received	05/12/2009 02:17	HINDAWI_bag_2009_04_21 (eDeposit for eJournals) by Laura Graham	Success	▾ Details

Comments

Figure 4. Reviewing the Life Cycle History of a Bag in the Inventory System

Packages are associated with a program and/or project, which are associated with a custodial unit, a content type (textual, still image, audio, etc.), a content process (partner transfer, digital conversion, web archiving, etc.), and an access category. Since it must also represent the history of a package, it records location paths and events that occur on a package level and on a file level. Examples of events include:

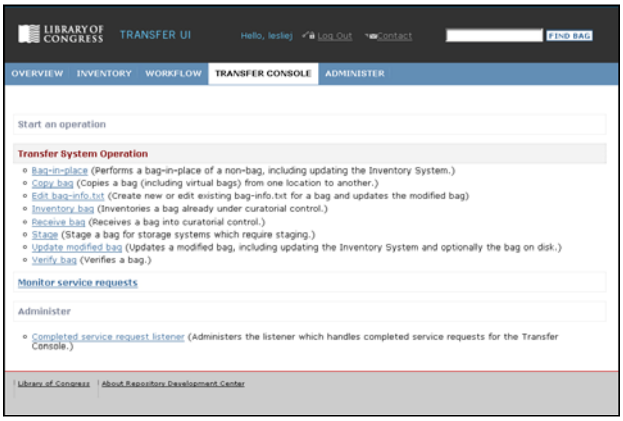
- Received Events, which include initial checksum verification and recording into the inventory;
- Quality Review Events, recorded when quality review is performed and noted as passed or failed;
- Accepted or Rejected Events, recorded when a project accepts or rejects curatorial responsibility for a package, usually due to verification failure or a failure to meet expected standards;
- Copy or Move Events, recorded when content is copied or moved from one location to another;
- Modification Events, recorded when a package or file has been modified, added or deleted;
- Delete Events, when entire packages are removed from the system;
- Ingest Events, when content has been ingested into a repository or access application;
- Recon Events, for the inventorying of legacy content already under Library control; and
- Verify Events, for ongoing auditing of fixities.

All events are recorded with the name of the performing agent and full date/timestamps. Multiple copies of content can be recorded as related instances, each with their own event history.

[9] http://sourceforge.net/projects/loc-xferutils/
[10] https://www.hibernate.org/
[11] http://www.loc.gov/standards/premis/

The Library has implemented low-level services such as file copying, inventorying, and verification, which are distributed across multiple servers as service containers. Mechanisms are available for invoking, managing, and monitoring the services through the command line or a web interface. Of particular note is the Copy Selector, which provides transparent access to a number of supported transfer protocols and tools; depending upon the source and copy locations, the most appropriate mechanism will be automatically selected (rsync, SCP, Signiant[12]) without the user having to be aware of the best option. Inventorying and verification services take advantage of the BIL Library and the Inventory Services.

The **Transfer Console/UI** is a web application that provides access to most aspects of the above services, plus project-specific workflows. It allows viewing and updating of the Inventory System, ad hoc transfer services (the Transfer Console), the monitoring and management of transfer services and workflows, as well as auditing and reporting functions. The name of this service is somewhat misleading; while it originally supported only transfer functions, it has been extended to supporting auditing and reporting on all inventoried content in the Library's server environment. The Transfer Console UI was implemented using Spring MVC.[13]



Start an operation

Transfer System Operation

- Bag-in-place (Performs a bag-in-place of a non-bag, including updating the Inventory System.)
- Copy bag (Copies a bag (including virtual bags) from one location to another.)
- Edit bag-info.txt (Create new or edit existing bag-info.txt for a bag and updates the modified bag)
- Inventory bag (Inventories a bag already under curatorial control.)
- Receive bag (Receives a bag into curatorial control.)
- Stage (Stage a bag for storage systems which require staging.)
- Update modified bag (Updates a modified bag, including updating the Inventory System and optionally the bag on disk.)
- Verify bag (Verifies a bag.)

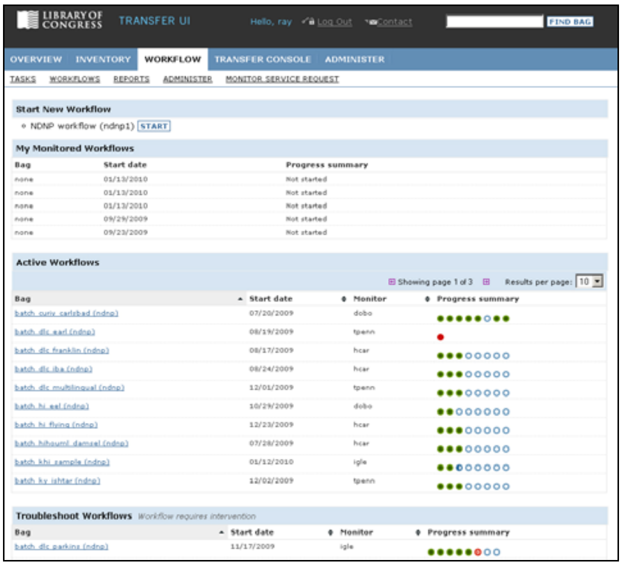
Monitor service requests

Administer

- Completed service request listener (Administers the listener which handles completed service requests for the Transfer Console.)

Figure 5. Transfer Console Functions

The **Workflow Framework** supports the implementation of project-specific workflows that automate parts of the digital lifecycle by coordinating machine and manual processes. The underlying workflow engine is jBPM, an open-source workflow system.[14] The drivers of a workflow are process definitions, which represent the process steps. jBPM Process Definition Language (jPDL), the native process definition language of jBPM, is used to encode the workflow process steps as XML. A workflow can be designed using the visual editor Graphical Process Designer, a plug-in for the Eclipse platform.



Start New Workflow

- NDNP workflow (ndnp1) START

My Monitored Workflows

Bag	Start date	Progress summary
none	01/13/2010	Not started
none	01/13/2010	Not started
none	01/13/2010	Not started
none	09/29/2009	Not started
none	09/23/2009	Not started

Active Workflows

Showing page 1 of 3 Results per page: 10

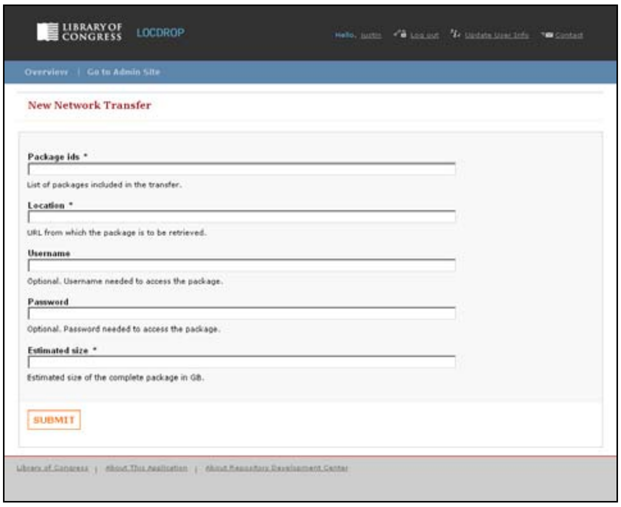
Bag	Start date	Monitor	Progress summary
batch curiv carlsbad (ndnp)	07/20/2009	dobo	●●●●●○●●
batch dlc earl (ndnp)	08/19/2009	tpenn	●
batch dlc franklin (ndnp)	08/17/2009	hcar	●●●○○○○○
batch dlc iba (ndnp)	08/24/2009	hcar	●●●○○○○○
batch dlc multilingual (ndnp)	12/01/2009	tpenn	●●●○○○○○
batch hi eel (ndnp)	10/29/2009	dobo	●●○○○○○○
batch hi flying (ndnp)	12/23/2009	hcar	●●●○○○○○
batch hihouml damsel (ndnp)	07/28/2009	hcar	●●●○○○○○
batch khi sample (ndnp)	01/12/2010	igle	●●◐○○○○○
batch ky ishtar (ndnp)	12/02/2009	tpenn	●●●○○○○○

Troubleshoot Workflows *Workflow requires intervention*

Bag	Start date	Monitor	Progress summary
batch dlc parkins (ndnp)	11/17/2009	igle	●●●●●●○○

Figure 6. Overview of Workflows for the Processing of Batches

In order to support the expanding numbers and types of transfers, a tool was needed to help automate transfers. The **LocDrop Service** is a web-hosted application for use by transfer partners in registering a new transfer; this application will support the registration and initiation of the transfer content via network transfer and via fixed media, such as hard drives or DVDs. LocDrop uses SWORD as its deposit protocol. At the time of this writing, LocDrop is in its initial use by multiple Library digital content acquisition projects, incorporating feedback into continued development.



New Network Transfer

Package ids *

List of packages included in the transfer.

Location *

URL from which the package is to be retrieved.

Username

Optional. Username needed to access the package.

Password

Optional. Password needed to access the package.

Estimated size *

Estimated size of the complete package in GB.

Figure 7. Initiating a New Network Transfer Using LocDrop

All of the applications are developed using an agile process, and undergo extensive QA testing at the completion of each iteration. As each application nears a state where it is feature complete, it is released to staff from one or more projects for user acceptance testing, the results of which are incorporated into the development process. After testing by partners, a number of new features were identified for inclusion in

[12] http://www.signiant.com/
[13] http://www.springsource.org/
[14] http://www.jboss.com/products/jbpm/

LocDrop. While the Inventory System and the Transfer Console started out as two applications, user feedback showed that it would more useful to integrate the two services into a single interface. And as a result of testing by internal Library users and partners, the interface for Bagger changed significantly from the 1.x to the current 2.x development versions.

3 PLANNING FOR FUTURE WORK

At the time of this writing, the Content Transfer Services have been put into production for NDNP [6]. The Inventory Tool has been put into production for content transferred to the National Digital Information Infrastructure and Preservation Program (NDIIPP), and production implementation is nearing completion for the Inventory, LocDrop, and Bagger applications.

The development of these services is ongoing at the Library, tentatively scheduled through 2011. A number of tasks have been identified for the remainder of the initial period and transition into full production. An inventory that is independent of any storage system allows the Library to track the location of digital content and checksums to support auditing. While procedures for inventorying newly acquired or produced Bags/digital content is in place for some projects, procedures must be put in place for inventorying all new Library content. As well, a complete inventory of all existing legacy content and full coverage of the production Library server environment is required. This effort is underway.

Currently the Transfer UI and Transfer Console support a workflow for the National Digital Newspaper Program as well as ad hoc and project-specific transfer and inventorying activities. We envision additional project-specific workflows can and will be developed using the Workflow Framework and integrated into the UI to automate reliable, repeatable Bag-level bit preservation activities. As program offices/projects identify their needs, workflows will be formalized and added into the framework.

As the tools and services move into production and use by a greater number of Library projects and staff, the interface will require review and revision for increased usability. An ongoing iterative review and revision of interfaces will be put into place.

These services fit into a larger context of development over the next three years at the Library to implement tools that enable staff across the Library to easily perform digital content management and curation tasks. While we are currently focusing on Bag-level bit preservation, not all content will always be Bagged, and data managers and digital curators think in terms of files, not Bags. The current Bag-level services include limited tracking of files within Bags, but do not currently support file-level auditing and reporting other than lists and counts of file format types. The planned progression of work is to complete the development of services supporting Bags and then move on to file-level services. These services will be implemented as extensions to the Inventory System. Once all Bag-level services are in production (inventorying, auditing, and reporting on Bags), work will commence on adding file-level services, such as file format auditing, file validation, and, potentially, preservation risk reporting. This work requires that policies and procedures on preservation storage, auditing, and preservation formats be in place.

When data managers and digital curators think of files, it is often in terms of their relationships to “objects” that they represent and collections that they are part of. We will continue to focus on bit preservation, but we are considering methods to additionally support an overlay of services that identify which files have relationships to each other (compound objects, master and derivatives, etc.), which file(s) represent which objects, and potentially link to descriptive metadata in other systems. Understanding that a file is a TIFF that represents a page from a specific atlas in the Geography and Maps Division, that another file is a JPEG2000 derivative file representing the same page, and that a third file is a JPEG used as a web thumbnail in addition to managing those bits is important for the preservation and sustainability of the collection as a whole.

4 CONCLUSIONS

Why are such transfer tools and processes so important? After much experimentation, the best transfer practices that have emerged relied upon established, reliable tools; well-defined transfer specifications; and good communication between content owner and content receiver. Each transfer provided insight into the developing content transfer best practices and each exchange brought more expertise. The digital preservation community continues to engage with transfer best practices, helping these practices to evolve. Ultimately, these practices and tools focus not just on transfer optimization, but on ways in which to improve the communication between submitter and receiver. The most important part of transfer is not the connection but the exchange of information. Communicating what is coming, when it will arrive, what form it will take, making the process predictable and flexible is vital.

Why are we looking at close integration between transfer and inventory functions? Inventorying and audit functions have been identified as a vital aspect of data curation. Inventory services can bring several benefits, including collection risk assessment and storage infrastructure audits. Realizing any benefits for effective data management relies on knowledge of data holdings. Knowledge of file-level holdings and recording of life cycle events related to those files from the moment that they enter the collection and in every future action reduces future risk by storing information that can be used in discovery, assessment, and recovery if and when a failure occurs.

Transfer processes are not surprisingly linked with preservation, as the tasks performed during the transfer of files must follow a documented workflow and be recorded in order to mitigate preservation risks. Defining, implementing, and documenting appropriate transfer processes depends on the requirements of each collection building project, which can vary wildly. While our initial interest in this problem space came from the need to better manage transfers from external partners to the Library, the transfer and transport of files within the organization for the purpose of archiving, transformation, and delivery is an increasingly large part of daily operations. The digitization of an item can create one or hundreds of files, each of which might have many derivative versions, and which might reside in multiple locations simultaneously to serve different purposes. Developing tools to manage such transfer tasks reduce the number of tasks performed and tracked by humans, and automatically provides for the validation and verification of files with each transfer event.

Bit preservation is not synonymous with digital preservation, but is rather an essential subset of digital preservation activities. So why is the focus on bit level operations? Bit preservation is not a solved problem [7]. Bit preservation is a useful starting point because bit-level activities tend to have more in common than activities at other levels. The act of copying a file is the same regardless of whether the file is an image or text or geospatial data. All files should have their formats validated and the checksums regularly verified, whether they represent newspaper pages or a photographs or manuscripts. As well, it is often sufficient to guarantee only the preservation of digital content as bits; in some situations that is all that is possible.

The work at the Library described in this paper has not focused on storage systems (as per Rosenthal); that work is progressing in the Enterprise Systems Engineering group at the Library and elsewhere [8].[15] Inventorying and audit functions have been identified as a vital aspect of data curation and preservation. The Library's developing services provide observability of the state and location(s) of files, enabling querying, auditing and reporting. This allows the Library to manage its bits as well as additional levels of abstraction: that the bits represent certain types of data (file formats), and that they have relationships (to batches, projects, curatorial divisions). Knowledge of file-level holdings and recording of life cycle events related to those files from the moment that they enter the collection and in every future action reduces future risk by storing information that can be used in discovery, assessment, and recovery if and when a failure occurs. This reduction of risk is vital to the Library's near-term preservation activities.

5 REFERENCES

[1] Anderson, Martha. 2008. Evolving a Network of Networks: The Experience of Partnerships in the National Digital Information Infrastructure and Preservation Program. *The International Journal of Digital Curation* (July 2008: Volume 3, Issue 1). http://www.ijdc.net/ijdc/article/view/59/60.

[2] Beckley, Elizabeth Thompson. LOC Expands Tech Focus: Saving Sound and Scene. *FedTech Magazine* (November 6, 2008). http://fedtechmagazine.com/article.asp?item_id=490

[3] Johnston, Leslie. Identifying and Implementing Modular Repository Services: Transfer and Inventory. In *Proceedings of DigCCurr 2009: digital curation: practice, promise & prospects: April 1-3, 2009, University of North Carolina at Chapel Hill, NC USA* (pp 145-148). Edited by Helen R. Tibbo, et. al., University of North Carolina, Chapel Hill, N.C.: 2009.

[4] Littman, Justin. 2006. A Technical Approach and Distributed Model for Validation of Digital Objects. *D-Lib Magazine* (May 2006: Volume 12, Number 5). http://www.dlib.org/dlib/may06/littman/05littman.html.

[5] Littman, Justin. 2007. Actualized Preservation Threats: Practical Lessons from Chronicling America. *D-Lib Magazine* (July/August 2007: Volume 13, Number 7/8). http://www.dlib.org/dlib/july07/littman/07littman.html.

[6] Littman, Justin. 2009. A Set of Transfer-Related Services. *D-Lib Magazine* (January/February 2009: Volume 15, Number 1/2). http://dlib.org/dlib/january09/littman/01littman.html.

[7] Rosenthal, David S. H. . "Bit Preservation: A Solved Problem?" In *Proceedings of iPRES2008*, London, UK, September 2008. http://www.bl.uk/ipres2008/presentations_day2/43_Rosenthal.pdf

[8] Schiff, Jennifer L. Library of Congress Readies New Digital Archive. *EnterpriseStorageForum.Com* (October 10, 2007). http://www.enterprisestorageforum.com/continuity/article.php/3704461

[15] See the presentations from the "Designing Storage Architectures for Preservation Collections" meeting, held September 22-23, 2009, at the Library of Congress: http://www.digitalpreservation.gov/news/events/other_meetings/storage09/index.html.