Preservation Policies

Next steps for moving towards preservation

1. Start documenting policy decisions. This page, its siblings, and its children are good starting points.
    a. As we make decisions, determine what documents we should sort the decisions into, and what person/group should manage each document. Many documents will be managed by the Repository Manager on a daily basis, but regularly reviewed by the Repository Preservation Board.
2. Look over documentation from other preservation repositories. Possibilities include:
    - Harvard
    - California Digital Library
    - Digital Curation Center
    - National Library of Australia
    - Arts and Humanities Data Service
3. Determine who should be invited to join a Repository Preservation Board. This group needs broad representation across the libraries and the major users of the repository, *but* needs to be small enough to make decisions effectively. This group should meet regularly to ensure that policies meet the needs of the repository's users, and ensure that policies are being followed.
4. As collections are added to the repository, validate whether they meet the current policies. Provide some central way to track this...
5. Move towards a better Fedora/HPSS connection, so we can take advantage of preservation tools that are developed by the Fedora community.
6. Set up an initial meeting of a Repository Preservation Board.

RLG "Trusted Digital Repository" checklist

A new version of this checklist, called the TRAC, Trusted Repository Audit and Certification, has been released.

The most important document related to preservation is RLG's Audit Checklist for the Certification of Trusted Digital Repositories. While this checklist is a good place to start working on our preservation system, we won't treat it as a mandate.

Melanie Schlosser has started a workspace to organized the items on the checklist and document how the current DLP activities relate to them.

Items on RLG's checklist of immediate interest. For the most part, these items relate to our use of HPSS:

- 3.6 Repository commits to define, collect, track, and provide, on demand, its information integrity measurements.
- B1.1 Repository identifies properties it will preserve for each class of digital object.
- B1.5 Repository obtains sufficient physical control over the digital objects to preserve them. (The description of this is confusing, encompassing many different aspects of repository functionality.)
- B2.1 Repository has an identifiable, written definition for each AIP or class of information preserved by the repository.
- B2.4. Repository has and uses a naming convention that can be shown to generate visible, unique identifiers for all AIPs.
- B3.7 Repository actively monitors AIP integrity. (This includes maintaining checksums in a location separate from media files, and maintaining logs of the integrity checks.)
- B4.2 Repository can demonstrate that referential integrity is created between all AIPs and associated descriptive information.
- B5.2 Repository logs all access management failures, and staff review inappropriate "access denial" incidents. (This should probably apply both to Fedora and to HPSS.)
- D1.1 Repository functions on well-supported operating systems and other core infrastructural software. (Not sure if HPSS will ever meet this...)
- D1.2 Repository ensures that all platforms have a backup function sufficient for the repository's services and for the data held, e.g., metadata associated with access controls, repository main content, etc.
- D1.4 Repository has mechanisms in place to insure any/multiple copies of digital objects are synchronized.
- D1.5 Repository has effective mechanisms to detect data corruption or loss. For example, if the policy were the repository could not lose more than 0.001% of the collection per year, then the repository would need to confirm media integrity at least every 72 days to achieve an average time-to-recover of 36 days or about one tenth of a year.
- D1.6 Repository reports to its administration all incidents of data corruption or loss, and steps taken to repair/replace corrupt or lost data.
- D1.7 Repository has defined processes for storage media migration.
- D1.8 Repository has a documented change management process that identifies changes to critical processes.
- D1.9 Repository has a process for testing the effect of critical changes to the system.
- D1.10 Repository has a process to stay current with the latest operating system security fixes
- D3.4 Repository has written disaster preparedness and recovery plan(s), including at least one off-site copy of all deposited data. Multiple off-sites copies are expected of most repositories, but others may be able to justify not providing these.
- D3.5 Repository tests disaster plans regularly.
- D3.6 Repository has defined processes for service continuity and disaster recovery.

Possible threats to the preservation repository

- Failure to access HPSS

- Failure of HPSS tapes
    - Simultaneous failure in both locations is possible (nuclear war)
- Fire, tornado, or other catastrophic failure in WCC machine room
- Disgruntled employee or random attacker (who obtains the password) attempts to delete everything from HPSS
- Disgruntled employee attempts to delelte random sets of items from HPSS

Other preservation-related items

List of file formats and their expected longevity from the Library of Congress.

A paper on using Fedora as a preservation system

OAIS is the primary model for building preservation systems.

RLG's older document, Trusted digital repositories: Attributes and Responsibilities.

The October 2005 RLG DigiNews describes requirements for "certification" of a digital repository. Many of these requirements are related to preservation.

There is a paper from Los Alamos about integrating OAI and OpenURL with an OAIS model.

There is a paper on bottom-up preservation issues that may also be useful.

The architecture of the LOCKSS system may be a useful guide, but LOCKSS currently relys on each participating institution "owning" a copy of the item. This is most useful for electronic journals.

Portico is a preservation archive for e-journals, based on Documentum. They perform extensive validation before accepting data into their repository.

The Florida Center for Library Automation is developing an archive system. This may provide some ideas, and they may eventually release code that we can use.

Vol. 54 (2005) no. 1 of 'Library Trends' is a theme issue, entitled: "Digital Preservation: Finding Balance", with many interesting articles.

Xena is a program that can take many file formats and convert them to a (supposedly open-format, long-term stable) XML format. It has been used with DSpace.

In the IU computer science department, Thomas Reichherzer and Geoffrey Brown are working on tools to aid "preservation via emulation".

A paper on preservation issues from the LOCKSS group. And the related Saving Bits Forever presentation.

Another paper on preservation issues.

Building an integrity checker

If all goes well, an integrity checker will never report a problem. But a non-existent or non-running integrity checker would exhibit the same behavior. So we must set up tests that ensure the integrity checker is regularly exercised. One simple way would be to put an object in the repository and purposefully corrupt it, then measure how long it takes for the integrity checker to notice.

It may be useful to have two levels of integrity checking:

- a basic ping service that checks whether a file appears to be in place (e.g., HPSS reports that the file is there and is the correct size, but we don't take the time to download it)
- a full checksum service that downloads a copy of the file and verifies its checksum

Other thoughts

- We need a system to provide periodic reports on the existence of orphaned and incomplete objects, so we can keep the contents of the repository "clean".
- Eventually, we would like to set up a "digital preservation advisory board" to proof our decisions. This page and its children will eventually be maintained by the board.
- We want to preserve metadata as well as the master files, so all preservation activities should apply to the local disk space that Fedora manages in addition to the HPSS storage.
- For maximum security, we will need to make offline copies of everything. But how can we verify the integrity of the offline copies? What if the verification process destroys the copies? The only way to manage this is to ensure that the offline copies are read-only.

Open questions

1. Can we eventually store lots of small files (page images) directly in HPSS via the filesystem interface without aggregation, or will it simply take too long to retrieve these files?
2. Should we store copies of our metadata in HPSS, or are the regular server backups good enough for this?
3. How do we manage preservation packages for materials that will be accessed through Variations? We don't want to store duplicate copies of the derivative files, because they are quite large. Perhaps Fedora can store these as Redirect datastreams, and just keep the appropriate metadata in the actual repository directories.
4. What is the best way to provide "proof" that something hasn't been altered since it was originally digitized?