The MetaArchive Cooperative

Collaboratively Preserving Our Digital Memory

Katherine Skinner, PhD
Executive Director, Educopia Institute

January 8, 2014
Overview

- History and Current Practices
- Philosophy
- Research
- Practical Application
MetaArchive History

- Distributed digital preservation cooperative
- Founded 2004
- Supported by NDIIPP, NHPRC, NEH, IMLS
- Preservation aims: prevent loss and corruption from human malice/error or from a disaster
- First (known) preservation network to preserve special collections/unique materials

Skinner 2014
MetaArchive Practices

- Basic processes
  - “Producer” (OAIS) determines curation practices; brings SIPs to MetaArchive (low entry barriers)
  - Multiple copies of AIPs dispersed across geographical, political, and environmental lines
  - Checks and repairs automated across network
  - Supports automated versioning
  - Deaccession cycle versus data deletion

Skinner 2014
MetaArchive Practices

- More than 50 members in four countries
- Public, private, and academic libraries
- 20 geographically distributed nodes
- 200+ TB network preserving 150+ collections
- Ingest via repository harvest or BagIt
- Regular testing of restoration processes
- TRAC self-audit 2010; will pursue certification under ISO 16363 once there are certified auditors (ISO 16919 not yet approved)
50 members, 14 states/districts, 4 countries

Auburn University
Boston College
Carnegie Mellon University
Clemson University
Consorci de Biblioteques Universitaries de Catalunya
Florida State University
Folger Shakespeare Library
GALILEO Knowledge Repository
Greene County Public Library
HBCU Library Alliance
Indiana State University
Library of Congress
Oregon State University
Penn State University
Pontifícia Universidade Católica do Rio de Janeiro
Purdue University
Rice University
University of Hull
University of Louisville
University of North Texas
University of South Carolina
University of Tennessee
Virginia Tech
MetaArchive Pricing

- Three membership levels
  - Collaborative members: $2.5K/year
  - Preservation members: $3K/year
  - Sustaining members: $5.5K/year
- Server cost: <$5K/term
- Storage cost: $1K/TB/year
Member Responsibilities

- Undertake a 3-year membership term;
- Take responsibility for content preparation, evaluation, staging, and ingest testing;
- Monitor collections to ensure accurate long-term preservation;
- Host and maintain a MetaArchive cache (server)
MetaArchive Philosophy

- Common, neutral center
- Distribution of work
- Community of engagement
  - Building knowledge
  - Accomplishing preservation
- Concentrated effort toward unified goal

Skinner 2014
*Conspectus home*

### Archives

Archives are groups of collections that belong to a certain topic.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Title</th>
<th># of Collections</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETD</td>
<td>Electronic Theses and Dissertations</td>
<td>16</td>
</tr>
<tr>
<td>SDC</td>
<td>Southern Digital Culture Collections</td>
<td>133</td>
</tr>
<tr>
<td>EML</td>
<td>Early Modern Literature</td>
<td>3</td>
</tr>
<tr>
<td>GEN</td>
<td>Content that does not fit into any other archive.</td>
<td>58</td>
</tr>
<tr>
<td>NEWS</td>
<td>Newspapers</td>
<td>4</td>
</tr>
<tr>
<td>IH</td>
<td>Institutional History</td>
<td>2</td>
</tr>
<tr>
<td>COMM</td>
<td>Community Generated Content</td>
<td>8</td>
</tr>
</tbody>
</table>
Collection Details

- Title: Virginia Tech VA-news
- Archive: GEN
- Plugin: org.metasearch.AFLFromStart2 by Metaarchive
- Base URL: http://scholar.lib.vt.edu/Chronicles_Ingest
- Identifier: 238
- Description: The Virginia news collection at Virginia Tech. The collection includes the Roanoke Times from January 1992 to February 1997, the Virginian Pilot from 1994 to 1997, and WDBJ7 news transcripts from October 1995 to August 2008
- Repository Type: Not specified

Related Items
- Metadata
- Plugin XML
- Archival Units

This collection has no metadata.
Adding Collections

New Collection

- **Title**: The title of the collection, e.g. ‘John Hilton Papers’

- **Description**: (Optional) Some text describing the collection’s contents

- **Base url**: The base URL to be passed to the LOCKSS plugin for crawling, e.g. http://your-institution.edu/library/collection/

- **Plugin**: (Optional) The LOCKSS plugin that will be used to crawl this plugin

- **Archive**: The archive this collection should be categorized under

- **Repository type**: (Optional) What kind of repository this collection is hosted on

- **Versioning type**: (Optional) How this collection’s content should be revisited by LOCKSS

Meta Archive Cooperative
Plug-In pathway
Plug in Wizard

Plugin Template Details

Name: All from Start

Description: Craws everything from a given start path, blah blah blah blah.

```xml
<string>plugin_config.props</string>
<list>
  <org.lockss.daemon.ConfigParamDesc>
    <key>start</key>
    <displayName>Start</displayName>
    <type>LockssConfigPathType</type>
    <definition>true</definition>
    <defaultValue>false</defaultValue>
    <defaultOnly>false</defaultOnly>
  </org.lockss.daemon.ConfigParamDesc>
  <org.lockss.daemon.ConfigParamDesc>
    <key>lockss_url</key>
    <displayName>Locks URL</displayName>
    <description>Usually of the form http://&lt;journal-name&gt;:com/</description>
    <type>LockssURL</type>
    <definition>true</definition>
    <defaultValue>false</defaultValue>
    <defaultOnly>false</defaultOnly>
  </org.lockss.daemon.ConfigParamDesc>
  <org.lockss.daemon.ConfigParamDesc>
    <key>path</key>
    <displayName>Path</displayName>
    <type>LockssConfigPathType</type>
    <definition>true</definition>
    <defaultValue>false</defaultValue>
    <defaultOnly>false</defaultOnly>
  </org.lockss.daemon.ConfigParamDesc>
</list>
</plugin>
<entry>
<string>plugin_version</string>
```
Auditing Control

Archival Unit Details

Collection: Virginia Tech VA-news

Plugin: org.metaarchive.AllFromStart2 by Metaarchive

Status: Test

Up/Down: Site Up

Parameter Values:
- base_url: http://scholar.lib.vt.edu/Chronicles_ingest
- start: VA-news_bag

Preservation Status

Coming soon.

Auditing Tools

Use the form below to request an audit on this archival unit.
When the process completes, a new Report will be created.

Audit Type

[Input field]

The type of audit.

Cache

[Input field]

The cache to use.

[Button] Begin Audit
Research: Practical Application

- NEH & IMLS
  - BagIt Ingest Implementations
    - Bag Splitting & Rebuilding Support
    - Bag Auditing Tools
    - Change Management Support for Bags
  - Curation Micro-Services Ingest Implementations
    - Format Recognition
    - Virus Checking
    - PREMIS Metadata Creation & Management

Skinner 2014
What is BagIt?

- BagIt is a packaging specification for validating digital content exchanges between 2 or more parties (some people also use it to store content)
  - Not a ZIP or a TAR package - easier to peek inside
  - Creates a manifest (listed inventory) of folders/files and corresponding checksums (md5 or sha1)
  - Contains a data directory of all included folders/files
  - Has an optional metadata file
- There are several open source BagIt tools that will package & validate sent packages
## BagIt Structure

### Index of /chronicles/GTWhistle

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td>27-Feb-2013 13:15</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>bag-info.txt</td>
<td>27-Feb-2013 13:15</td>
<td>593</td>
<td></td>
</tr>
<tr>
<td>bagit.txt</td>
<td>27-Feb-2013 13:11</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>data/</td>
<td>27-Feb-2013 13:11</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>manifest-md5.txt</td>
<td>27-Feb-2013 13:11</td>
<td>148K</td>
<td></td>
</tr>
</tbody>
</table>
Bag-It Metadata

Bag-Software-Agent: bagit.py <http://github.com/edsu/bagit>
Bagging-Date: 2013-02-27
Payload-Oxum: 523930395.2743
Contact-Email: jody.thompson@library.gatech.edu
Contact-Name: Jody Thompson
Contact-Phone: 404-894-9626
External-Identifier: SMARTech:/1853/3790
Source-Organization: Georgia Tech Library
Organization-Address: 704 Cherry St, Atlanta, Ga 30332-0900
External-Description: Faculty newspaper issues submitted by the Georgia Tech Library. Files were delivered as born digitally and scanned from print. License information and extracted text information is included.
Bag-Size: 526M

bag-info.txt
BagIt Version & Encoding

BagIt-Version: 0.96
Tag-File-Character-Encoding: UTF-8

bagit.txt
## BagIt Data Directory

### Index of /chronicles/GTWhistle/data

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td>27-Feb-2013 13:11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>102/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>103/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>104/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>105/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>106/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>107/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>108/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>109/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/</td>
<td>13-Feb-2013 15:36</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How Do You Build Bags?

- Several BagIt utilities that will “bag” content – below are the most popular

  - Java Bag Library (Library of Congress/CDL)

  - Bagit.py (Ed Summers Python Tools)
    https://github.com/edsu/bagit

  - Bagger-GUI (Library of Congress/CDL)
    http://sourceforge.net/projects/loc-xferutils/files/loc-bagger/
MetaArchive BagIt Research

- NHPRC Project (2007-2011)
  - Exported 256 AUs (1.4 TB) into “bags” and successfully transferred to Chronopolis
  - Created workflows & scripts for packaging

- NEH Chronicles in Preservation (2011-2013)
  - Ingesting digital newspaper collections as “bags”
  - Bags have no size limitations (can = multiple TBs)
  - Creating scripts to generate smaller “bags” from larger “bags” for MetaArchive Archival Units (AUs)
Benefits of Using BagIt

- Pre-Ingest File Fixity & Inventory
- Improved Post-Ingest Auditing
- Simplified Ingests
  - Out-of-the-box plugin
  - Simple manifest page
  - No need to worry about Archival Units upfront
- Maintains Collection Integrity
The first thing that a member would need to do is export copies of their digital collections/files.
The member then needs to get these placed on a local staging server for performing Bag creation operations.
Once staged the member can run any of a number of user-friendly BagIt utilities (Java Bag Library, bagit.py, Bagger-GUI) to “bag” up collections/files.
This will give the member a “bag” sitting on their staging server. What now?
MetaArchive Bag Creation

If the bag can fit into our current AU sizes (1GB-30GB), then we can put a manifest page & plugin in place, define the AU...
...and off we go!

Skinner 2014
But what if the bag is bigger than our current AU sizes (1GB-30GB)?
Then we need to apply a custom-built script to split the large bag into several smaller bags.

```bash
$ bag splitbagbysize <BAG> --maxbagsize 30
$ python bagit-split.py split <BAG>
```
MetaArchive Bag Creation
We would put a manifest page and plugin in place, define the AUs...
MetaArchive Bag Ingest

...and off we go!

Skinner 2014
So...we know getting it in works!
MetaArchive Bag Export

What about getting it out and Putting it back together again?
MetaArchive Bag Rebuilding

$ python bagit-split.py unsplit <DIRECTORY CONTAINING BAGS>
1. Make sure collection AUs have 100% agreement
2. Use LOCKSS Export Content feature on one cache (uncompressed zip) – download to processing server
3. Run Bag Unsplit Utility
   1. $ python bagit-split.py unsplit <DIRECTORY CONTAINING BAGS>
MetaArchive Change Management

- MetaArchive members change collections over time
  - Add files
  - Change existing files
  - Remove files
- LOCKSS Versioning
  - Automated re-crawling for changes
  - Replaced versions are stored but not voted/pollled on
- MetaArchive Versioning Policy
  - Semi-automated and scheduled re-crawling for changes
  - Improves uniformity and agreement levels across copies
Member Staging Server & Bag

Skinner 2014
1. New Files!
2. Changed Files!
3. Deleted Files!

Member Staging Server
&
Bag w/changes
Three basic methods:

1. Re-Bag Approach
   - Swap out old bag for a new updated bag
   - Easier when managing a single storage environment

2. Bag Merge Approach
   - Sync & update existing bags according to diffs
   - Easier when there are fewer reps to manage

3. Change Bag Approach
   - Bag and ingest only the files that have changed
   - Easier when storage is diverse and there are more reps
MetaArchive BagIt Change Management

Re-Bag Approach?

Member Staging Server &
Bag w/changes

Skinner 2014
MetaArchive BagIt Change Management

Member Staging Server & Bag w/changes

Re-Bag Approach?

Skinner 2014
MetaArchive BagIt Change Management

Member Staging Server & Bag w/changes

Re-Bag Approach? Bag Merge Approach?
Member Staging Server & Bag w/changes

Re-Bag Approach?
Bag Merge Approach?
MetaArchive BagIt Change Management

Member Staging Server & Bag w/ changes

Re-Bag Approach?
Bag Merge Approach?
Change Bag Approach?
MetaArchive BagIt Change Management

Member Staging Server & Staged Bag

Skinner 2014
MetaArchive BagIt Change Management

Member Staging Server & Bag w/changes

Skinner 2014
MetaArchive BagIt Change Management

% python bag-diff.py /path/to/bag

% python bag-diff.py -m /path/to/old/manifests/dir /path/to/bag

Member Staging Server & Bag w/changes

Skinner 2014
Member Staging Server
w/Bag
plus a new
Updates & Changes Bag
Member Staging Server w/Bag
plus a new Updates & Changes Bag

Skinner 2014
Member Staging Server w/Bag

plus a new

Updates & Changes Bag

Skinner 2014
MetaArchive BagIt Change Management

Boston College Libraries - Chronicles Project Test Exchanges

Collection

- Boston College bcheights-bcsrh-bc885

- Skinker 2014
find-bad-files.py
  Recursively scans a directory for filenames that violate a set of naming standards meant to prevent problems when ingesting collections into LOCKSS over HTTP.

lockss-manifest-validate
  a tool to verify checksum hashes produced by LOCKSS against hashes provided by a BagIt manifest document.
Get the Scripts

https://github.com/MetaArchive

- find-bad-files.py
- bag-split.py
- lockss-manifest-validate
- bag-diff.py
Katherine Skinner
404 783 2534
katherine@educopia.org

www.metaarchive.org