MetaArchive Quality Assurance Strategy

1. Introduction

Quality assurance is the combination of control measures and quality control checks to ensure quality and consistency in a process. MetaArchive’s LOCKSS network has a variety of roles and responsibilities to ingest and preserve digital objects. The quality assurance program described below outlines how MetaArchive and its members work together to ensure that the desired content is receiving ongoing preservation. This document is organized into sections based on workflow stages described in How MetaArchive Works and user roles described in About MetaArchive Documentation.

It may be desirable to manage some or all of these activities through a project management tool like Airtable that allows automated messages and reminders to be scheduled, in addition to being transparent to members. The activities in this document have been mapped to a RACI matrix to clarify roles and responsibilities (displayed below).

<table>
<thead>
<tr>
<th>Workflow Stage</th>
<th>Task</th>
<th>Data Wrangler</th>
<th>Content Manager</th>
<th>Server Admin</th>
<th>Network Admin</th>
<th>Cache Buddy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Establish project management tracking</td>
<td>R</td>
<td>R</td>
<td>R = Responsible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Gather and collect digital objects</td>
<td>R = Responsible</td>
<td>C = Consulted</td>
<td>I = Informed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Sample objects to validate selection criteria</td>
<td>A = Accountable</td>
<td>R = Responsible</td>
<td>C = Consulted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Check for bad filenames.</td>
<td>R = Responsible</td>
<td>C = Consulted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Check for PII, PHI, FERPA data</td>
<td>R = Responsible</td>
<td>C = Consulted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Retain relationships between AUs using names</td>
<td>R = Responsible</td>
<td>C = Consulted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Check that no AU’s exceed 300 GB.</td>
<td>R = Responsible</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Sample packaged objects to validate</td>
<td>R = Responsible</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>Sample serialized objects to validate</td>
<td>R = Responsible</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ingestion</td>
<td>Validate new plugin in the test network</td>
<td>I = Informed</td>
<td>R = Responsible</td>
<td>I = Informed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ingestion</td>
<td>Submit MetaArchive Ingest Form</td>
<td>R = Responsible</td>
<td>I = Informed</td>
<td>C = Consulted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ingestion</td>
<td>Project management tracking</td>
<td>C = Consulted</td>
<td>I = Informed</td>
<td>R = Responsible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ingestion</td>
<td>Validate content post-ingestion</td>
<td>R = Responsible</td>
<td>C = Consulted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor &amp; Repair</td>
<td>Monitor network health and hygiene, monthly</td>
<td>A = Accountable</td>
<td>R = Responsible</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor &amp; Repair</td>
<td>Apply OS and software updates, review access logs</td>
<td>R = Responsible</td>
<td>I/C = Consulted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor &amp; Repair</td>
<td>Review recent ingest for AU status updates, quarterly</td>
<td>R = Responsible</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor &amp; Repair</td>
<td>Review polling quorums</td>
<td>I = Informed</td>
<td>R = Responsible</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor &amp; Repair</td>
<td>Audit 5% random sample of network AUs</td>
<td>A = Accountable</td>
<td>R = Responsible</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Prepare

The preparation of content for ingestion into MetaArchive involves many steps and decisions as directories and files are arranged into archival units and packaged. The following recommendations and quality control checks for data wranglers helps members assure that the content they think they are sending is indeed the content that is sent.

Primary Role(s): data wrangler

- When asserting intellectual control over a collection and its objects, don’t rush. Take the time to gather and collect the necessary information.
- While reviewing selected collections and objects, sample some objects to be sure they meet your selection criteria.
- Once your files are organized, data wranglers should run the find-bad-files python script to identify any invalid file names. Rename any files identified and re-run the script.
- If it hasn’t already occurred as part of processing, run tools to check for personally identifiable information, such as bulk extractor. Sensitive data is not permitted in the MetaArchive network.
- When arranging collection objects into Archival Units (AUs), use naming conventions that indicate their relationship. Be sure that no individual AU exceeds the maximum size limit of 300 GB.
- If you are packaging AUs into BagIt bags, sample some of the bags to verify they contain the expected content. Also check for required metadata in bag-info.txt.
- For serialized AUs (converted to uncompressed ZIP or TAR files), unpack a sample and verify that content matches what is expected.

3. Ingestion

Ingestion is a complicated process that can sometimes involve ingesting AUs into both a test and production network. Using the test network for new collections is an important quality control process to ensure that the collection plugin is configured properly.

Primary Role(s): content manager, network administrator

- For new collections, always test the collection plugin in the test network first. Use the same steps below to verify AUs are being harvested and ingested into the test network as intended before establishing a collection plugin in the production network.
- Content managers should use the MetaArchive Ingest Form to initiate ingest.
- The network administrator should track the progress of ingest batches so that all steps can be completed. Information and steps to record:
  - Total data size
Number and names of archival units
Assigned caches to achieve 5 replications
Member and contact information
Date ingest request
Task completion
  ▪ AU status updated from Stage to Test or Test to Preserved
  ▪ LOCKSS title database updated, as necessary
  ▪ Request sent to content managers requesting production ingest
  ▪ Ingests as completed by content managers or network administrator
Date ingest completed and progress toward completion
AU Availability status updated in Conspectus to stop crawling (as applicable)
  ▪ Content Managers should examine AU’s within the network to verify contents are what is expected. This can be done in a couple ways.
  ▪ Non-cache hosting members can contact a network administrator to request access to another member’s cache to perform these steps. The cache should be one that ingested the AUs being reviewed. These two members are referred to as cache buddies. Ideally, the non-cache hosting members content should always be ingested by their cache buddy.
  ▪ The LOCKSS GUI contains two features that members can use to review AUs. Both are available when viewing an AU’s status page.
    (LOCKSS Daemon Status AU Ids (from dropdown))
      ▪ Serve AU
        ▪ Serve AU is a feature that “replays” the harvested content as it was staged. Generally, it looks like a rendered LOCKSS manifest that you can click through to explore the content.
        ▪ Content managers (possibly assisted by the data wrangler) should use Serve AU to verify that the ingested content is what is expected.
      ▪ AU File table
        ▪ The AU Status page includes a table of all the URLs for each file in the AU. If you click on a URL, you will see a split page, the top half will include LOCKSS metadata on the file and the bottom half will attempt to render the content.
  ▪ Conspectus contains a feature to validate BagIt bag manifests for AUs stored in LOCKSS. This requires you to retain a copy of the bag manifest to upload and compare.

4. Monitor and Repair

Active preservation is a key characteristic of digital preservation. LOCKSS’ polling and voting features enable regular integrity checks and self-repair in the event of damaged content. Described here are other activities that contribute towards quality assurance.

Primary Role(s): network administrator, content manager, server administrator

On a monthly basis:
  ▪ A network administrator should review network monitoring tools (Cache Manager, Nagios, PLN Monitor) to identify issues and create related tasks for the technical team or others. Potential issues include:
    ▪ Blocked critical ports/services (LCAP, SSH)
    ▪ Unavailable caches
    ▪ Expired TLS/SSL certificates
    ▪ DNS resolution issues
    ▪ Disk space under 1 TB
    ▪ RAID errors
  ▪ The server administrator should apply any updates to the cache operating system and review access logs for any signs of unauthorized access. Some of this may be automated.

On a quarterly basis:
  ▪ A network administrator should review ingests from the past three-months to see if any updates are necessary in Conspectus for AU Status or Site Availability.
  ▪ A content manager should review polling quorums to ensure a 70% threshold is achieved. Due to LOCKSS’ crawling nature and depending on the availability and manner in which content is published to the web, there may be small inconsequential differences in the harvested content across caches that will cause polls less than 100% agreement.

On an annual basis:
  ▪ A network administrator should initiate an annual audit of content preserved in MetaArchive’s production LOCKSS network. Some of this is a data collection activity that can be automated while other parts require manual action. The audit should be a 5% random sample of AUs being preserved. Any issues discovered should be addressed; if any patterns are found they should be documented and discussed.

Sample audit log:
  ▪ For each AU, the following data should be gathered
    ▪ AU Name
    ▪ Plugin
    ▪ AU Status
    ▪ Site Availability
    ▪ Disk Usage
    ▪ Ingest Date
    ▪ File count
    ▪ Replication count and location of each.
  ▪ For each AU replication, the following should be recorded:
    ▪ Poll status
    ▪ Results of content examination (Good, Bad, Questionable)
  ▪ Server administrators who have automated systems or tools for updating and monitoring caches should review configurations for any required updates due to infrastructure changes.

5. Retrieve

This section is undeveloped. Presumably there is a way to validate or verify that retrieved content matches expectations. Insufficient experience to add additional details.
6. Recover

This section is undeveloped. Presumably there is a way to validate or verify that retrieved content matches expectations. Insufficient experience to add additional details.

Notes

1. The LOCKSS GUI also had a feature called Serve Content. This feature is less useful for MetaArchive as we do not index bibliographic metadata for AUs. It is more useful in the Global LOCKSS Alliance.