Schedule 3: Phase 2 work

David Livingstone Archive

Version History

Version	Description	Author	Date
1.0	Initial Draft	Nigel Banks	June 1st, 2016
1.1	Feedback	Nigel Banks	June 3rd, 2016

Ingest of Archival Packets

Write and execute two scripts, one to sync the *Archive Packets* to the production server, and the other to ingest those packets into Fedora, and remove them afterwards. This ingest script will not support updating, it will be single use. After this is complete users will be able to download the *Archival Packets* from the *Browse by Catalogue* page.

Estimate: < 5 hrs

Restricted Item Link

Change the browse by catalog “Access” column to show when objects are restricted:

This icon will appear **only** when the manuscript has viewable content (pages) and that content is limited to only administrators, it will not show when the manuscript has no content. Administrators will **not** see this icon as they have access to all content. This icon does not relate to downloadable Archive Packets.

The viewer will be displayed when this icon is clicked but it will not render the images it will instead display a message like “Livingstone Online has images of this item, but not permission to show them publicly. Individuals with specific research questions can contact us, and we can consult the item on your behalf”.

Estimate: < 5 hrs

Drupal Batch Import

To support Sitewide Search and the Browse by Timeline work, we need to import content from Fedora into Drupal. This will be achieved using the Solr index. This will require us to define a new *Drupal Content Type* that contains the appropriate fields.

A *nightly* cron job will execute a Solr query to fetch all the interesting information about the repository (Title, Creators, TEI content, etc) and import it into Drupal as *Drupal nodes*. A differential update will be made based on the Fedora object’s last modified timestamp to speed up this process.

After the update takes place another cron job will run to update the Drupal search index (this is provided by Drupal but must be triggered afterwards).

In addition to the automated cron job, a minimal administration interface will be built for testing purposes and to force updating when one does not wish to wait until the next day for the index to be updated. It will allow the user to:

- Remove a Manuscript node identified by its PID (i.e. liv:00001)
- Remove all Manuscript nodes
- Add / update a single Manuscript node identified by its PID (i.e. liv:00001)
- Add / update all Manuscript nodes.
- Display if the index is out of date

Estimate: 16 hrs

Sitewide Search

To have the *Sitewide Search*, link back to the appropriate page for Manuscript items we’ll be using *Drupal Node Redirects*.

This will rewrite the URLs for matching manuscript items such that it will take the user to *Browse by Catalogue* page, and display the Manuscript in the viewer when appropriate. Non-viewable Manuscripts will still link to the *Browse by Catalogue* page, but they will not launch the Manuscript viewer.

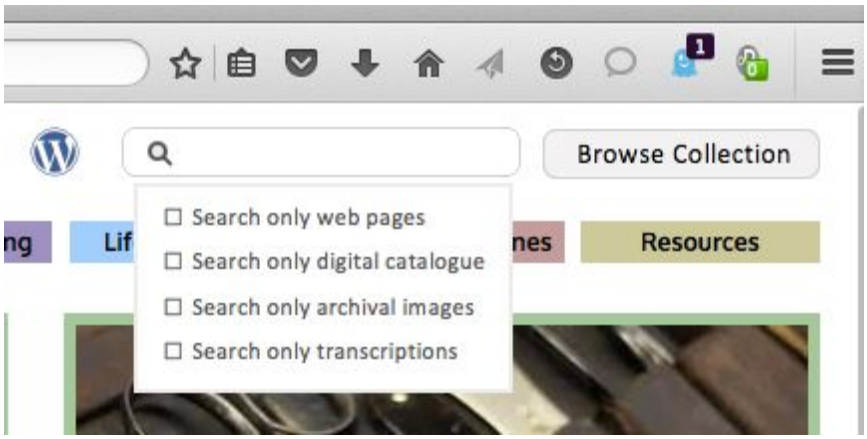
Search results will have links to pages like http://livingstoneonline.org/node/643 will then redirect to pages like:
http://livingstoneonline.org/islandora/search/”liv:000001”?view_pid=”liv:000001”

This **does not** include any changes to the display of the sitewide search page.

Estimate: N/A

Sitewide Search Widget

This is to build the sitewide search widget displayed here:



Since this is only searching *Drupal nodes* and does not require any Islandora based expertise I would suggest that you have Kathy build this for you as it would be more cost efficient.

Estimate: N/A

Batch Upload / Replace via FTP

This tool is meant to update the content in Fedora from the Agnes FTP, it will rely on the data in the FTP server having a specific folder structure along with *manifest files* which describe each folder.

Manifest Files
Each *manifest file* will be a CSV file (***manifest.csv***), that list the files that constitute that object along with the computed *MD5 checksum* for each file like so:

> *liv_012046_0001.tif.xmp, 26379e5b930ab2f708aba72d338866c9*
> *liv_012046_0001.tif.txt, 4009d0c1796da2ad0b95ba4f274e5b7f*
> *liv_012046_0001.tif, 9e9754bd87c22d82b0423f1fc4fe4dae*

A script will be provided for generating these manifest files, given a folder as input it will generate the **manifest.csv** file in the given folder (*assumes the files are on your local computer*).

Folder Structure

The bullet points below show the folder hierarchy and it's expected contents. Each of the top level folders define the types the system supports (new types will require additional work).

manuscript objects will function as they currently do. **no_crop** objects will be related back to the manuscript in which they are defined from, and added to a collection which contains them all **no_crop** objects. **illustrative** objects will not be related to any other object but will exist in a collection all to themselves. **no_crop** and **illustrative** objects will be restricted to administrators only.

- ➔ manuscripts (***These correspond to the 01, 02, 03 directories***)
  - ◆ public (***These are accessible to everyone***)
    - liv_012046 (***Single manuscript including pages and archive***)
      - manifest.csv (required)
      - liv_012046_0001.tif.xmp
      - liv_012046_0001.tif.txt
      - liv_012046_0001.tif
      - liv_012046.zip
      - etc
  - ◆ private (***Only administrators can access images***)
    - liv_012047 (***Single manuscript including pages and archive***)
      - manifest.csv (required)
      - liv_012047_0001.tif.xmp
      - liv_012047_0001.tif.txt
      - liv_012047_0001.tif
      - liv_012047.zip
      - etc
- ➔ no_crop (***These correspond to the 04 directory***)
  - ◆ liv_000082 (***Single no crop object, all admin access only***)
    - manifest.csv (required)
    - liv_000082_0001_noCrop.tif.xmp
    - liv_000082_0001_noCrop.tif.txt
    - liv_000082_0001_noCrop.tif.md5
    - liv_000082_0001_noCrop.tif
    - Etc
- ➔ illustrative (***These correspond to the 05 directory, all admin access only***)
  - ◆ manifest.csv (required)
  - ◆ liv_013885_0001.tif.xmp
  - ◆ liv_013885_0001.tif.txt
  - ◆ liv_013885_0001.tif.md5
  - ◆ liv_013885_0001.tif
  - ◆ liv_013884_0001.tif.md5
  - ◆ liv_013884_0001.tif
  - ◆ etc

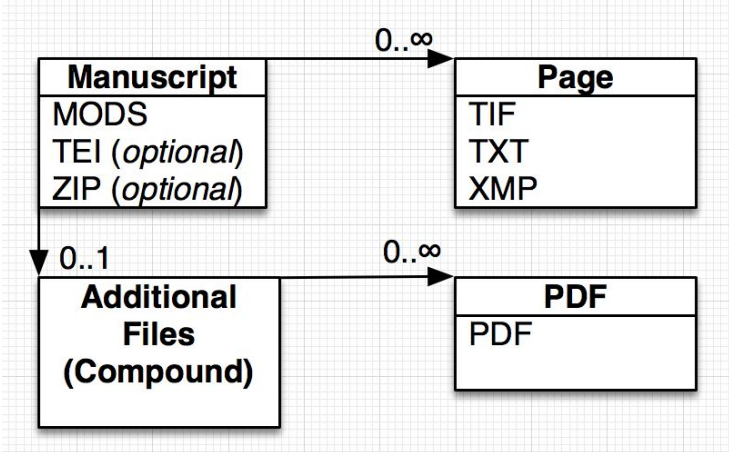
Object Types

For each of the top level types (*Manuscripts, No Crop, Illustrative*) listed above we create groups of objects that are comprised of the files in each of those top level types directories.

In the diagrams each box is an "*object*", and its "*type*" or "*content model*" is shown in **bold**. Inside each "*object*" it lists which files comprise it and if they are *optional*. In addition relationships are show between the objects, indicating how many of that type can be related to it.

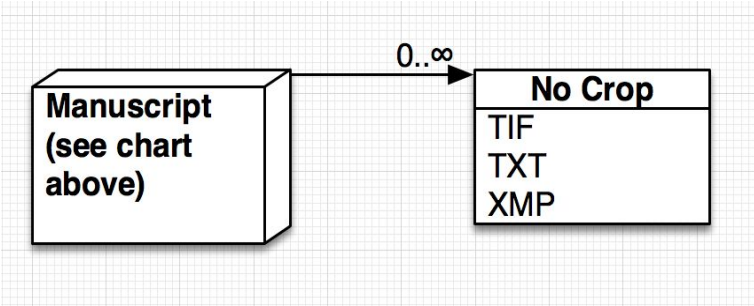
Manuscript

The "Additional Files" compound object is a place for things related to a manuscript that are not currently displayed or do not have a category of their own, for now it only includes PDF's.



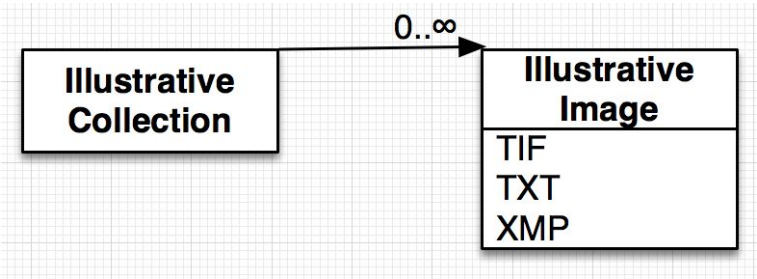
No Crop

No crop objects are essentially the same as paged, but have a different type, so that their display in the site can differ.



Illustrative

Illustrative images are all grouped under a single collection.



Update Process

1. Fetch Manifest Files From FTP
2. Compute files to be added / updated / removed
    a. Added files will be ones not in the repository but mentioned in the manifests
    b. Comparison for update will be done via MD5 in manifest
    c. Any files present in the repository but not present in any manifests will be removed
3. Remove a file
    a. Purge file from object within repository
    b. If object has no files
        i. Check for related objects and remove them as well
        ii. Remove the object
4. Add new file
    a. Copy file from FTP to the Server
    b. Create required Fedora object if not present (Depending on the location / file type additional files will be created / added)
        i. Restricted objects will generate an XACML Policy
        ii. MODS will be mapped to DC
        iii. JP2 will be generated from TIFF
    c. Delete the copied file, and any temp files.
5. Update a file in Fedora (*repeated for each file to be updated / added*)
    a. Copy file from FTP to the Server
    b. Update the file on the appropriate Fedora object
    c. Delete the copied file, and any temp files.

The update process will be triggered nightly via a cron job.

In addition to the cron job and minimal administrative interface will be build that will allow users trigger the **update process** manually.

This will only support syncing Manuscript, No Crop, Illustrative objects, as noted above in the **Object Type** section.

Estimate: 28 hrs

Browse by Timeline

This is to enable Drupal Simple Timeline module. Since Drupal Batch Import will bring in Drupal content, we can just use the module *as is* without extensive changes to support fetching information from Fedora. Kathy should be able to take the module and theme it as needed.

That being said there is no image data to import for display as thumbnails at this time, currently within Fedora we have TIFF and JP2 files which are far too large to be used as thumbnails. If we want to be able to include thumbnails in the output they will need to be generated and persisted in Fedora, as is outlined in Thumbnail Generation.

Time only includes adding the module to the build scripts and enabling it.

Estimate: < 1 hrs

Thumbnail Syncing

This task would be to add thumbnail support to the Batch Upload / Replace via FTP, and Drupal Batch Import. This would not be to generate thumbnails, but rather to copy them from the FTP server into Fedora and then add support for importing them into Drupal.

Estimate: 2 hrs

Site Backup

This would be to sync the Drupal files directory to the FTP server daily via a cron job. To minimize network congestion when transferring we'll use timestamps to sync only that which has changed. The Drupal files directory contains database dumps as generated by previous work Kathy has done so that will cover the database backup requirement as well.

Estimate: 2 hrs