9-10-2018

Colenda @ the University of Pennsylvania: Using a decoupled, pluggable architecture for object processing

Kate Lynch
University of Pennsylvania, katherly@upenn.edu

Recommended Citation

Lynch, K. (2018). Colenda @ the University of Pennsylvania: Using a decoupled, pluggable architecture for object processing. *Samvera Connect,* Retrieved from https://repository.upenn.edu/library_papers/111

Colenda @ the University of Pennsylvania: Using a decoupled, pluggable architecture for object processing

Abstract

This poster details the architecture of the repository and the deliverables of the first major release of Colenda, the open-source repository software developed at Penn Libraries. Staff in Digital Library Development & Systems created Colenda, a long-term preservation ecosystem including Samvera, an open-source software framework for repository development, at its core. Colenda is a Samvera instance that provides materials-agnostic fuThis poster details the architecture of the repository and the deliverables of the first major release of Colenda, the open-source repository software developed at Penn Libraries. Staff in Digital Library Development & Systems created Colenda, a long-term preservation ecosystem including Samvera, an open-source software framework for repository development, at its core. Colenda is a Samvera instance that provides materials-agnostic functionality for distributed workflows around administration of digital assets and metadata through a pluggable architecture for metadata schemata and entry. This poster offers a look at object processing workflows from the consumer end as well as a deep-dive into each component's purpose in the software stack.

Disciplines

Cataloging and Metadata | Library and Information Science | Programming Languages and Compilers | Systems Architecture | Theory and Algorithms

Colenda @ the University of Pennsylvania

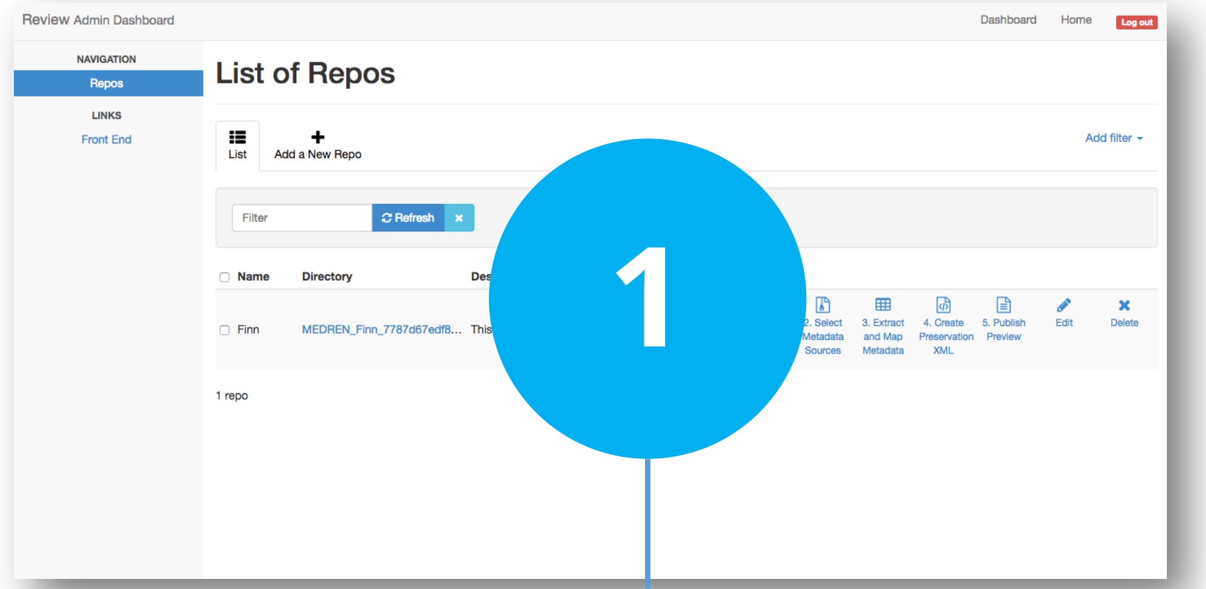
Make “bad” decisions with confidence!: Using a decoupled, pluggable architecture for object processing

Development & Deployment Team

- Katherine Lynch (Samvera)
  Senior Application Developer
- Michael Gibney (Git & git-annex)
  Senior Application Developer
- Martin Oestergaard (Ceph)
  Unix Systems Administrator

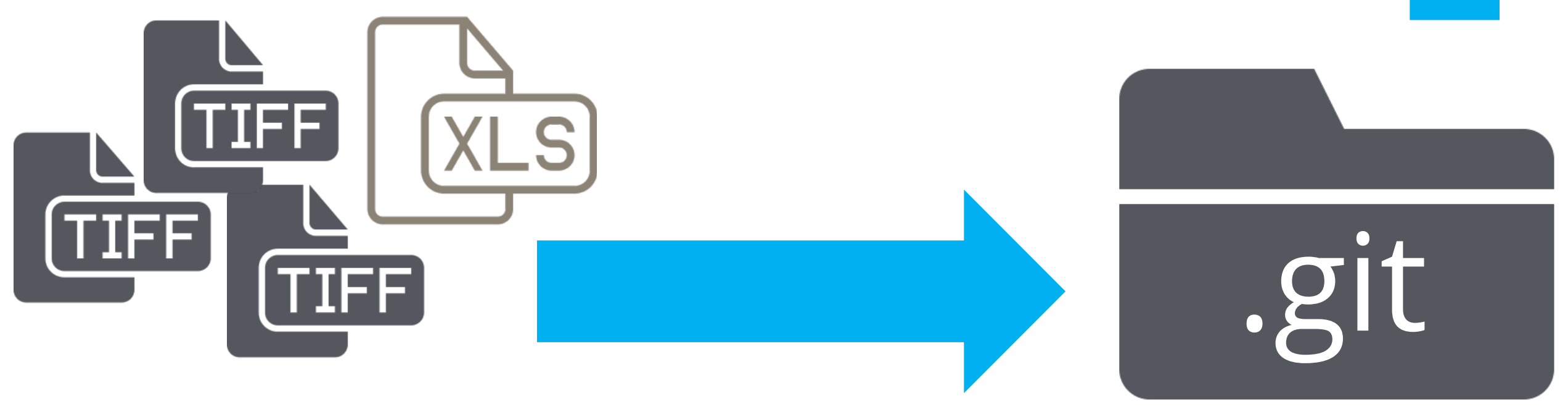
Penn Libraries are creating a long-term preservation ecosystem including Samvera at its core. Colenda is a Samvera instance that provides materials-agnostic functionality for distributed workflows around administration of digital assets and metadata through a pluggable architecture for metadata schemata and entry. Objects consist of two types of things: assets and metadata. Assets consist of all files associated with an object’s representation. Metadata consists of information describing the object or its structure. In Colenda, metadata sources come in the form of spreadsheets that directly contain metadata and/or information that allows metadata lookup services to extract bibliographic information for descriptive metadata from external sources such as the library catalog. Because the application accepts files as metadata sources, this allows the metadata entry interface to be versioned alongside the metadata it is creating, which allows an ongoing thorough understanding of the metadata based on the context in which it was entered.

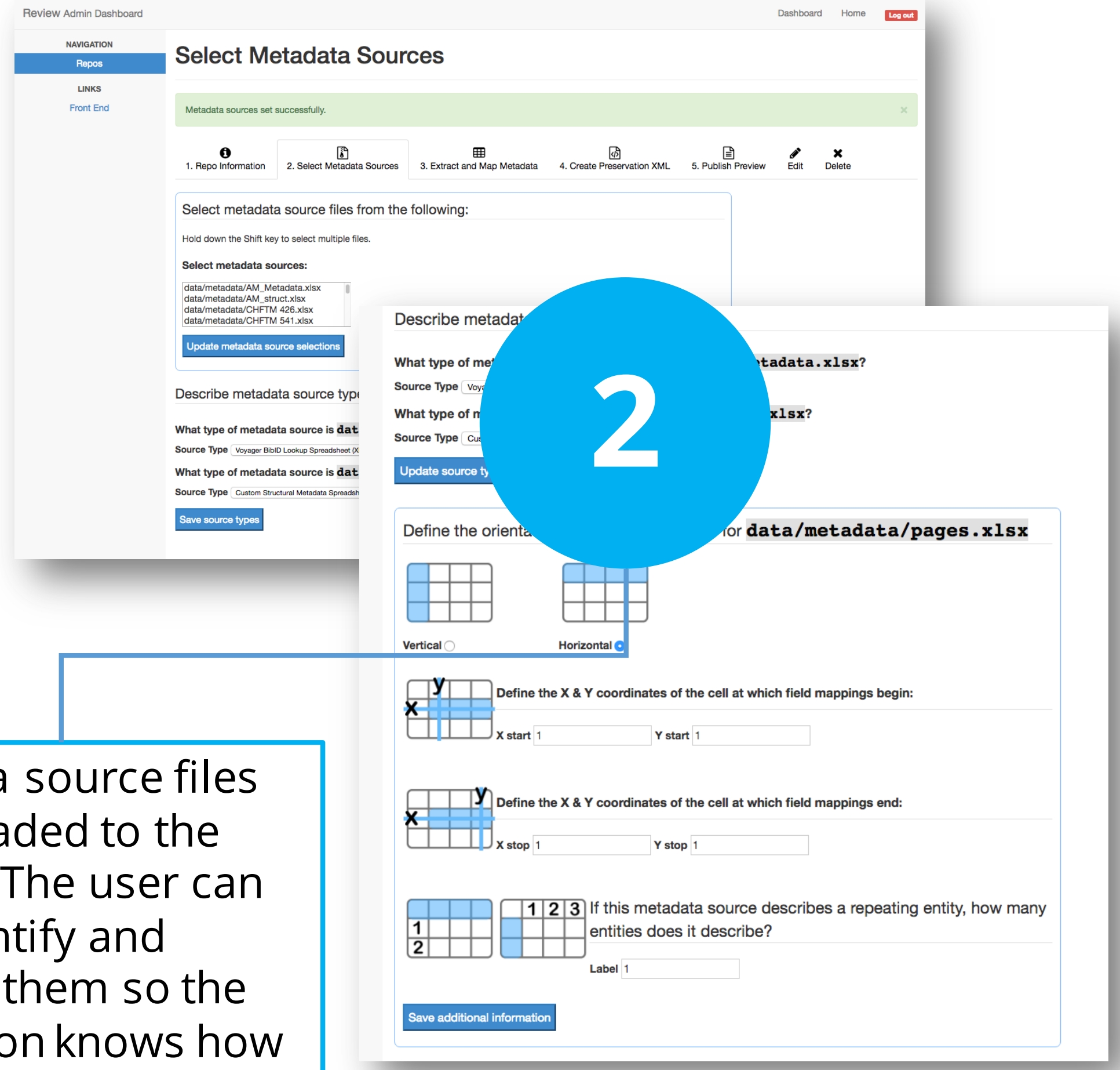
The beginning scope of the project addresses page-turning objects, specifically manuscripts from Penn’s collections, and represents significant development of a completely generalized approach to object creation and ongoing management. Its first phase of software development has just been completed and the team is currently working on crafting a robust stack of software and hardware fitting together with this application.


Review Admin Dashboard

List of Repos

Asset and metadata files are packaged in Git repositories. This allows users to upload securely using Git commands and supports long-term preservation as files are versioned, thereby allowing the system to keep track of all changes to files and metadata over the course of an object’s lifetime. Users create the Git remotes they will interact with for each object and perform administrative tasks to get the objects ready for ingest.




Select Metadata Sources

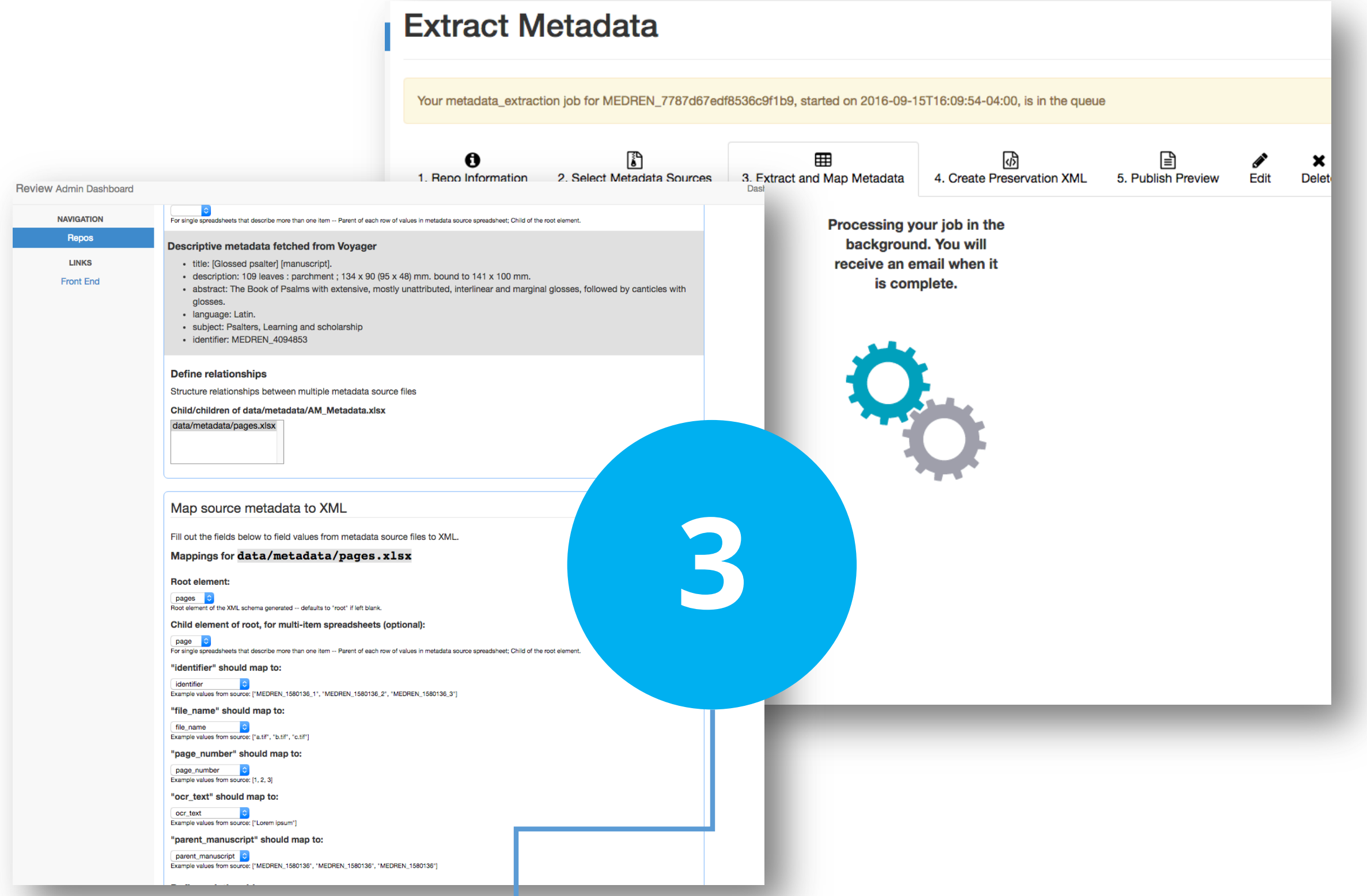
Describe metadata source

Define the X & Y coordinates of the cell at which field mappings begin:

Define the X & Y coordinates of the cell at which field mappings end:

If this metadata source describes a repeating entity, how many entities does it describe?

Metadata source files are uploaded to the remote. The user can then identify and describe them so the application knows how to handle them for metadata extraction.

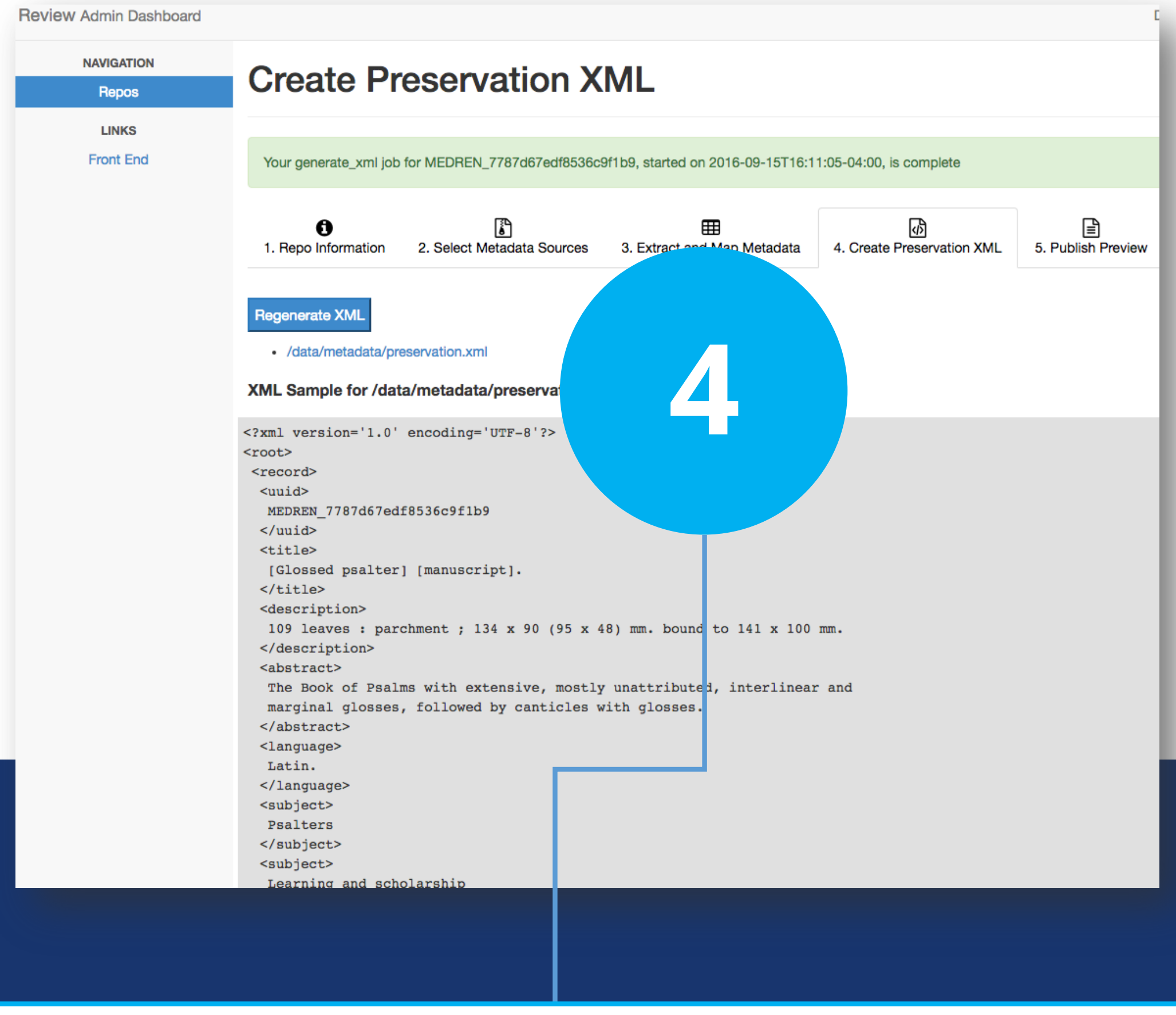

Extract Metadata

Processing your job in the background. You will receive an email when it is complete.

Map source metadata to XML

3
1. Repo Information
2. Select Metadata Sources
3. Extract and Map Metadata
4. Create Preservation XML
5. Publish Preview
Edit
Delete
Descriptive metadata fetched from Voyager
Define relationships

The application extracts and maps metadata for the object based on the contents and description of its metadata sources. At this stage, users can review the extracted metadata for verification, specify different field mappings if those used in the metadata source do not match up with the approved metadata schema for preservation, and define relationships between sources.


Review Admin Dashboard

Create Preservation XML

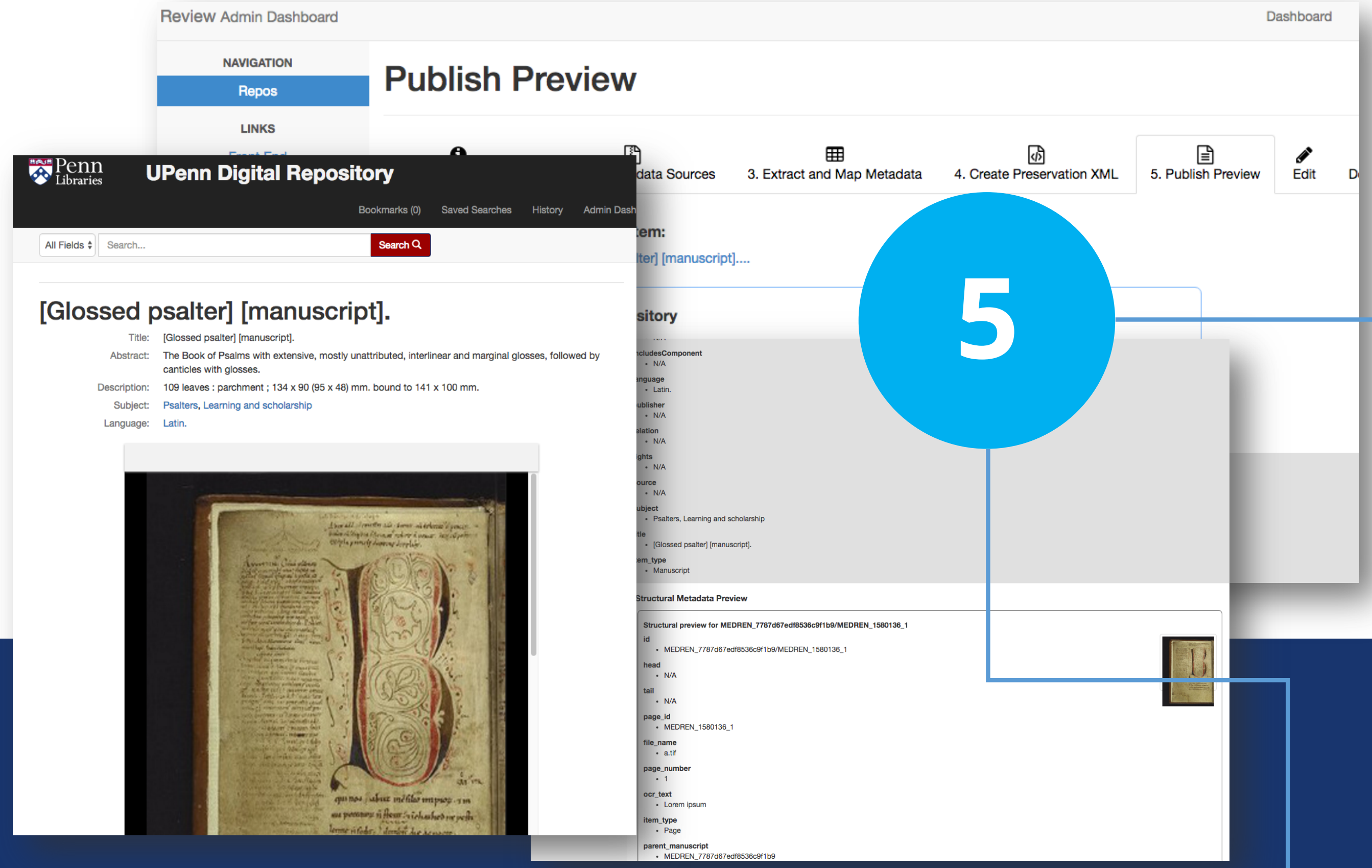
Your generate_xml job for MEDREN_7787d67edf8536c9f1b9, started on 2016-09-15T16:11:05-04:00, is complete

- /data/metadata/preservation.xml

XML Sample for /data/metadata/preserva

```
<?xml version='1.0' encoding='UTF-8'?>
<root>
 <record>
  <uuid>
   MEDREN_7787d67edf8536c9f1b9
  </uuid>
  <title>
   [Glossed psalter] [manuscript].
  </title>
  <description>
   109 leaves : parchment ; 134 x 90 (95 x 48) mm. bound to 141 x 100 mm.
  </description>
  <abstract>
   The Book of Psalms with extensive, mostly unattributed, interlinear and
   marginal glosses, followed by canticles with glosses.
  </abstract>
  <language>
   Latin.
  </language>
  <subject>
   Psalters
  </subject>
  <subject>
   Learning and scholarship
```

The application generates an XML document that represents the data and field mappings in their purest abstract form, which the user can then spot-check in the interface. This document is added to the object’s Git repository and is updated as metadata and field mappings change.


Review Admin Dashboard

Publish Preview

UPenn Digital Repository

[Glossed psalter] [manuscript].

Title: [Glossed psalter] [manuscript].
Abstract: The Book of Psalms with extensive, mostly unattributed, interlinear and marginal glosses, followed by canticles with glosses.
Description: 109 leaves : parchment ; 134 x 90 (95 x 48) mm. bound to 141 x 100 mm.
Subject: Psalters, Learning and scholarship
Language: Latin.

Structural Metadata Preview

The application provides two review interfaces: Blacklight, which represents the data as it will be seen in production, and a stripped-down admin interface where all metadata associated with an object is displayed alongside relevant assets for targeted review. This interface also features a persistent log of any asset-related problems detected during ingest, such as missing or corrupt files, and functionality to add working notes to the object if desired.

Using the generated XML file as a layer of data abstraction, the application ingests a representation of the object into Fedora and indexes it to Solr/Blacklight. Assets in the Git repository that are referenced in the structural metadata are picked up for verification, derivative creation, and ingestion. The object is now ready for review.

Repository Ecosystem

Colenda is intended for use as part of a larger repository ecosystem. The team at Penn Libraries is currently in the process of creating and test-deploying an architecture using a Ceph storage cluster, git-annex for versioning of large files, and Hydra/Rails for management of object files and metadata representation.

- **Colenda (Samvera & Rails)**
  Colenda uses Hydra for metadata aggregation in Fedora, as well as search and discovery in Solr/Blacklight. Additional functionality to curate and manage assets and metadata is provided in the application through Rails.
  Objects’ descriptive metadata is ingested to Fedora, and structural metadata referenced for display via the Rails application.
- **Git & git-annex**
  Git repositories created through Colenda’s workflows use git-annex, a tool that extends Git to version files for long-term preservation without checking the binaries into Git; rather, git-annex versions references to file content and location. Binaries are stored in a key/value store where the key is derived from a SHA-256 checksum of file content. This provides a layer of abstraction that allows users to flexibly rearrange items on the file system without breaking ingestion workflows. Files are stored on separate storage identified as a git-annex special remote. A special remote is an abstraction that handles storage of binary content but not versioning metadata, and integrates with the Git ecosystem.
- **Ceph Storage Cluster (S3)**
  The binaries referenced in Colenda’s Git repositories live on remote storage that runs as a Ceph storage cluster. Ceph is a hardware-agnostic software storage platform that prioritizes data replication across multiple nodes for high availability, high replication for fault-tolerance, and scalability. The versions of the files targeted for long-term preservation are stored in one place and referenced using git-annex and Fedora plugins on the remote, using an S3-compatible gateway API.