Publishing without Perishing
Alma Publishing Profiles for Fun and Profit

ELUNA 2023
Martin Patrick: martin.patrick@library.gatech.edu
Stacie Traill: trail001@umn.edu
Publishing Profiles 101
The Basics

- Slightly text heavy slides today, more in the speaker notes and appendix
- Data export options in Alma
- Pub profiles background and use cases
The Basics: Alma Export Jobs

- Export Authority records
- Export Bibliographic records
- Export Digital titles
- Export Electronic portfolios
- Export Physical items

But how to combine them all?
The Basics: Alma Analytics Exports

You could take tabular data and use a tool like MARCEdit to turn it into MARC records, or you could use pymarc to explore it.
Pros and Cons of Jobs and Analytics Export

Pros
- Data comes out exactly as it is
- Basic setup to export from Alma
- If you want just MARC data, tabular data, or MARCXML

Cons
- Cumbersome to combine bibs and holdings and items, etc.
- Data comes out exactly as it is
- Lack of output targets
- Analytics doesn’t have every field indexed and requires several weird tricks to combine things
The Basics: Alma Publishing

- Not just for sending data to Primo or BrowZine
- Flexible output options: FTP servers or OAI
- Ability to normalize data on the fly
- Bibliographic data is customizably enrichable with holdings, items, and portfolios
  - Choice of publishing one bibliographic for each item, or one bibliographic record with all attached items
Publishing Profiles: Best Practices (ioho)

- Publishing On option
- Publishing Order of Operations
- Enrichment Options
- Filtering
- Normalizing
The Basics: Publishing Settings
The Basics: Alma Publishing Enrichment

<table>
<thead>
<tr>
<th>Bibliographic Normalization</th>
<th>Physical Items Enrichment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct the data using normalization processes</td>
<td>Add Items Information</td>
</tr>
<tr>
<td>Linked Data enrichment</td>
<td>Committed to retain subfield</td>
</tr>
<tr>
<td></td>
<td>Retention note subfield</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bibliographic Enrichment</th>
<th>Electronic Inventory Enrichment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Management Information</td>
<td>Add Electronic Portfolio Information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Related Records Enrichment</th>
<th>Digital Inventory Enrichment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add related records information</td>
<td>Add Digital Representation Information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authority Enrichment</th>
<th>Collection Enrichment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Authority Information</td>
<td>Add Collection Information</td>
</tr>
</tbody>
</table>

| Physical Holdings Enrichment | |
|-----------------------------| |
Analyzing Bibliographic and Item Data: The Mysterious Case of the Retention Notes
Background

- Retention notes for HathiTrust and Scholar’s Trust had:
  - Been added to 583 in the Bibliographic Record in all cases
  - Been added to the Holdings record in some cases
  - Had the Retention Commitment flag turned on in zero cases

- The list of which actual items by barcode were committed was not made available, however I was assured the bib records were accurately marked. 😐
Questions to address

- How many of the bib records had more than one holdings record?
- How many holdings records had more than one item?
- Which holdings records *didn’t* have a 583 note?
- From a set of bib records, how can I best get a set of barcodes so I can update the Retention Flag and notes in the correct item record itself?
# Setting up Publishing

<table>
<thead>
<tr>
<th>Physical Items Enrichment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Items Information</td>
</tr>
<tr>
<td>Repeatable field</td>
</tr>
<tr>
<td>Item PID subfield</td>
</tr>
<tr>
<td>Copy ID subfield</td>
</tr>
<tr>
<td>Item Policy subField</td>
</tr>
<tr>
<td>Barcode subfield</td>
</tr>
<tr>
<td>Material type subfield</td>
</tr>
<tr>
<td>Item status subfield</td>
</tr>
</tbody>
</table>
Processing the Output

.mrc file → MARC Edit → List of Barcodes
Why would you want to?

- You need to set holdings for multiple symbols at a location level
- You wish to customize data output in ways not possible through out-of-the-box OCLC publishing.
- You want to control which records are sent to OCLC by means other than the Export to WorldCat flag (using logical set creation and/or filtering in publishing).
- You want to publish to more than one OCLC DataSync collection.
- You want more control over OCLC holdings deletion than the out-of-the-box job offers.
Process Flow

START
Records are automatically added to logical set when 909 field is added indicating that cataloging is completed

OCLC DataSync incremental publishing job runs, including all records in 909 set with changes since last run

Indication rule filters out selected brief records

END
MARC file is generated and ready for upload to OCLC FTP

Normalization process acts on MARC data (including enrichments) to further refine data output

Holdings and Item enrichment maps selected values to MARC records
Sets and Filtering

- Publishing to OCLC is managed on the basis of a local field defined to record Cataloging Status (MARC 909).
- No record is published to OCLC unless its MARC 909 indicates that cataloging has been completed.
- A logical set based on 909 data is the basis of incremental publishing.

```
909  __ |a cat |b MNU |c 2023-04-04
```

**Set name** OCLC Datasync July 2022 and onward with OCN  **Set type** logical where (Other System Number contains keywords "OCoLC" AND c.search.index_names.local_field_909 contains keywords "2022*") OR (Other System Number contains keywords "OCoLC" AND c.search.index_names.local_field_909 contains keywords "2023*")
Some brief records for special projects that wouldn't otherwise be excluded from the logical set are excluded on the basis of an indication rule in the publishing profile.

```plaintext
rule "909 brief projects"
when
(exists "909.d.iso_std" OR exists "909.d.nrl_gl")
then
set indication."true"
end
```
Customizing Data Output

- Normalization routine for OCLC DataSync includes rules to:
  - Delete local fields.
  - Delete OCLC symbol mapping fields for records carrying a locally-designated “no export” field.
  - Map locations in a statewide storage facility to the appropriate OCLC symbols.
  - Remediate some specific problems with Alma MARC output that have caused issues with OCLC in the past (010, 880).

- Normalization happens after holdings and item enrichment, so normalization can act on the basis of data added through those enrichment processes.
Publishing to more than one DataSync collection

- U of MN uses separate OCLC DataSync collections for records with OCLC numbers and records without OCLC numbers in order to use different matching algorithms.
  - Records with OCLC numbers match on that basis only
  - Records without OCLC numbers use OCLC's full match algorithm
- There is one logical set and one general publishing profile for each DataSync collection.
- Both publishing profiles use the same filtering and data customizations.
Holdings Deletes

- Holdings deletes are handled separately, through Alma Analytics reports.
- This process makes it easier to ensure that holdings are only deleted for the symbols corresponding to libraries who have withdrawn their last copy of a title.
- Delete reports are fetched on a schedule by a script through the Analytics API and uploaded to OCLC’s FTP.
Producing metadata for external repositories
Background

- Locally held items are digitized and made accessible through an instance of ContentDM using a homegrown metadata schema
- **Goal:** extract and transform MARC records for physical items to metadata suitable for ingest to ContentDM
- Multi-step workflow using Alma Publishing + Python allows transformations and enrichments at various points throughout the process.
Process Flow

START

Digitization department submits a list of barcodes for items to be digitized

Barcode list is used to create an itemized physical items set in Alma

The set is published at the item level

MARC file is fetched from local server and processed with Python script to map MARC elements to custom repository metadata schema in CSV file

MARC file is generated and output to local SFTP server

Normalization process acts on MARC data (including enrichments) to further refine data output

Holdings and item enrichment maps selected values to MARC records

OCLC numbers from MARC records are searched via HathiTrust API to determine whether equivalent items exist in HT

CSV file is enriched with HathiTrust links, if any were found

CSV file is returned to digitization staff for final checking and ingest to the repository

END
Enrichment: Holdings and Item Data

- The publishing profile produces item-level output (1 record per item).
- Holdings and item data needed in the repository is added to records through Physical Holdings Enrichment in the publishing profile.
Transformations, part 1: Alma Normalization Rules

- An Alma normalization routine was created which is used only in the publishing profile for this workflow.
- The NR removes fields not needed in the repository metadata and remaps some fields to alternative tags to make further transformations easier. For example, subjects are handled with less granularity in the repository than they are in MARC, so:

```plaintext
rule "UMedia subjects mapping"
when
  (TRUE)
then
copyField "600" to "699.{0, }"
copyField "610" to "699.{1, }
copyField "611" to "699.{1, }"
copyField "650" to "699.{5, }" if (exists "650.{ ,0}")
copyField "651" to "698"
end
```
Transformation example: subject fields

Alma original record:

600 10 |a La Salle d'Offémont, Adrien Nicolas Piédefer, |c marquis de, |d 1735-1818

650 _0 |a Slavery |z Haiti |v Early works to 1800.

650 _0 |a Black people |z Haiti |v Early works to 1800.

Published transformed record:

699  0\$aLa Salle d'Offémont, Adrien Nicolas Piédefer,\$cmarquis de,\$d1735-1818

699  5\$aSlavery\$zHaiti\$vEarly works to 1800.

699  5\$aBlack people\$zHaiti\$vEarly works to 1800.
Transformations, part 1: Alma Normalization Rules

Another remapping example: non-Latin script titles, authors, and contributors are mapped from 880 fields to their corresponding MARC tags:

```ruby
rule "UMedia 880 mapping"
when
  (TRUE)
then
  copyField "880" to "246.{ , }" if (exists "880.6.245*")
  copyField "880" to "246.{ , }" if (exists "880.6.246*")
  copyField "880" to "190.{1, }" if (exists "880.6.100*")
  copyField "880" to "191.{2, }" if (exists "880.6.110*")
  copyField "880" to "700.{1, }" if (exists "880.6.700*")
  copyField "880" to "710.{2, }" if (exists "880.6.710*")
end
```
Transformation example: non-Latin script fields

Alma original record:

880 10$6245-02$a本草圖譜 : $b[9卷及索引] / $c灌園岩崎常正著 ; 編纂者 飯田藏太郎.

880 1$6100-01$a岩崎灌園,$d1786-1842.

880 1$6700-06$a飯田藏太郎.

Published transformed record:

246  $6245-02$a本草圖譜 : $b[9卷及索引] / $c灌園岩崎常正著 ; 編纂者 飯田藏太郎.

190 1$6100-01$a岩崎灌園,$d1786-1842.

700 1$6700-06$a飯田藏太郎.
Transformations, part 2: MARC to CSV with Python

- The customized MARC output is processed by a Python script that outputs a CSV file suitable for ingest to the repository.
- A second script further enriches the CSV file by adding HathiTrust links when there is an OCLC number match.
Transformations, part 2: MARC to CSV with Python

```python
# Subjects mapping
if field_699:
    subj_list = []
    for f699 in field_699:
        subjs = f699.get_subfields('a', 'x', 'y')
        for subj in subjs:
            subj_list.append(subj.rstrip('.'))
    subjects = '||'.join(subj_list)
    rec_dict['Subject'] = subjects
```
Fin

Martin Patrick: martin.patrick@library.gatech.edu

Stacie Traill: trail001@umn.edu
Appendix
Analyzing OAI Feed with MARCEdit
Why would you even want to?

Some use cases:

- Ensuring your RapidILL/BrowZine feeds are accurate
  - Norm rules are working, things you can’t lend are being properly excluded, etc.
- If you don’t have an FTP server or don’t want to use Ex Libris Secure FTP
  - (or can’t for because it’s a whole thing with IT)
Ok, so what do I need?

What you need:

- Access to the Integration Profiles in Alma Configuration, or a willing Systems Librarian
- OAI DEFINITION enabled
- MARCEdit installed (Mac or Windows)
- A publishing profile that has published
- The profile’s set name
- Your current IP address
- Your OAI Repository Base URL
Integration Profile

OAI REPOSITORY DEFINITIONS

Active * ActiveInactive

Repository Name * UNI3L_OAI

Repository Base URL https://galileo-gatech.alma.exlibrisgroup.com/view/oai/01GALL_GIT/request

Protocol Version 2.0

Admin Email *

Earliest Datestamp * 01/01/2000

Date Format is dd/MM/yyyy

Deleted Record transient

Granularity YYYY-MM-DDTh:mm:ssZ

Metadata Prefixes

Allowed IPs
<table>
<thead>
<tr>
<th>Enabled</th>
<th>IP description</th>
<th>IP Version</th>
<th>IP Match Criteria</th>
<th>Updated By</th>
<th>Last Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Third Iron IP</td>
<td>IPv4</td>
<td></td>
<td>903284307</td>
<td>11/13/2019</td>
</tr>
<tr>
<td></td>
<td>Martin P testing</td>
<td>IPv4</td>
<td></td>
<td>903862682</td>
<td>09/07/2022</td>
</tr>
<tr>
<td></td>
<td>RapidILL 1</td>
<td>IPv4</td>
<td></td>
<td>903862682</td>
<td>09/07/2022</td>
</tr>
<tr>
<td></td>
<td>RapidILL 2</td>
<td>IPv4</td>
<td></td>
<td>903862682</td>
<td>09/07/2022</td>
</tr>
<tr>
<td></td>
<td>Martin P Testing 2</td>
<td>IPv4</td>
<td></td>
<td>903862682</td>
<td>04/20/2023</td>
</tr>
</tbody>
</table>
Server Address
https://galileo-gatech.alma.exlibrisgroup.com/view/oai/01GAL

Set Name
Rapid_Journals_Print_LendableInternational

Metadata Type
MARC21XML

Crosswalk Path:
/Users/martinpatrick/marcedit35/xslt/OAIMARC21XML.xsl

Start:  End:  

Use System Proxy

Process  Close
Computational conflicts: Conflict modeling for distributed intelligent systems, with contributions by numerous experts / Heinz Jürgen Müller, Rose Dieng (eds.).

RapidILL Norm Rule Examples

Martin’s RapidILL Norm Rules: https://github.com/martinpatrick/rapid-norm-rules