OCLC Data Sync Reports with Python

Colin Bitter
Head of Cataloging and Metadata
R. Barbara Gitenstein Library
The College of New Jersey

Catalog Management Interest Group
ALA Core Interest Group Week
February 3, 2021

https://github.com/colinbitter

TCNJ THE COLLEGE OF NEW JERSEY

OCLC Data Sync Collections

- Replaces batchload services
- Part of WorldShare Collection Manager
- Allows libraries to keep holdings aligned with WorldCat
- Several options for ongoing collections
  - **Bibliographic collection**
  - Local holdings records (LHR) collection
  - Non-MARC numeric search key collection
  - Non-MARC patterned data collection
- Reclamation (one-time project)
- More information available from OCLC https://help.oclc.org/Metadata_Services/WorldShare_Collection_Manager/Choose_your_Collection_Manager_workflow/Data_sync_collections?sl=en

TCNJ Publishing Profiles (Alma)

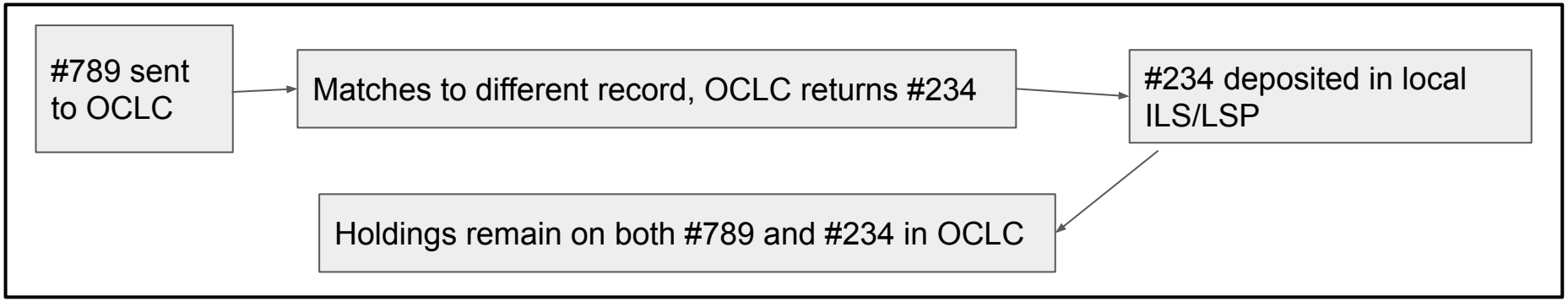
- Two publishing profiles in Alma
  - Electronic titles (all electronic institutional records with OCLC#)
  - Physical titles (all physical records set to “publish bib”)
- Files from each profile sent daily via FTP
  - Added, deleted, updated (determined by Alma logic)
- Average ≈3,000 records daily between both profiles

OCLC Datasync Settings

- Two datasync collections in OCLC, corresponding to Alma publishing profiles (physical/electronic)
- Local system number (Alma MMS ID) located in 001
- OCLC number located in 035 $a
- MARC record delivery disabled (only receive reports)
- Used number matching up until July 2020
- Currently use record matching
- Likely to resume number matching when available

OCLC Datasync Issues

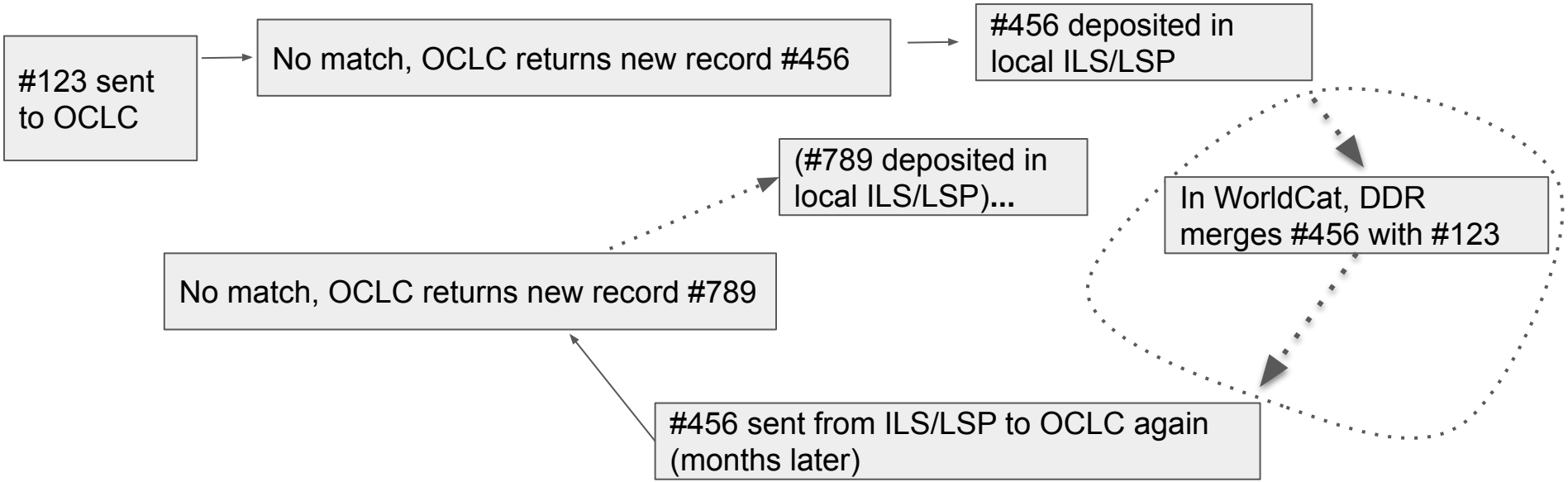
- Record matching can leave stray holdings in WorldCat



- Records constantly merging in WorldCat via DDR (Duplicate Detection Resolution) and Member Merge Project
- Lag between data sync and report processing
  - Reports contain new OCLC#s, but merges can occur after report delivery
- Duplicate records in local ILS/LSP can cause problems

OCLC Datasync Issues

- *Note: Data Sync Record Matching Algorithm different from DDR. Can result in "loop" behavior*



Reports from OCLC

- Numerous reports delivered after each sync (WorldShare>>Metadata>>My Files>>Downloads)
- Reports generated within hours of Alma publishing job
- TCNJ uses two reports
  - Bibliographic Processing Report (BibProcessingReport.txt)
  - Bibliographic Exception Report (BibExceptionReport.txt)
- Your mileage may vary
- **Use Python to enhance the Bibliographic Processing Report**
- Manually process the Bibliographic Exception Report

BibProcessingReport.txt

- Pipe delimited
  bibKey | **local system number** | **input OCLC** | **output OCLC** | bibAction
  NJT_156489|**9992589134051 91**|**1098222884**|**1098222884**|match

- Desired outcome:
  - Keep all OCLC#s up to date in Alma
    - Maintain links from WorldCat.org
    - OCLC# is match point for dozens of record sources/vendors
  - Keep holdings aligned with WorldCat

OCLC-Datasync-Postprocessing.py

- Intended for use with the Bibliographic Processing Report
- Input (from BibProcessingReport.txt)
  NJT54689 | 991146803405191 | 830722503 | 1232418344 | create
- Output (oclc_result.csv)

DATASYNC	MMSID	FILEINPUT	FILEOUTPUT	ACTION	FILEMATCH	BATCH	ORIGBATCHMATCH	NEWBATCHMATCH	HELD
NJT54689	991146803405191	830722503	1232418344	create	FALSE	1232418344	FALSE	TRUE	TRUE

OCLC-Datasync-Postprocessing.py

- Windows OS
- Libraries/modules
  - pandas, numpy, glob, path
  - bookops-worldcat developed by Tomasz Kalata (NYPL)
    https://pypi.org/project/bookops-worldcat/
    (Requires WorldCat Metadata API)
- Targets first text file in downloads folder (e.g., C:\Users\Colin\Downloads)
- Creates dataframe from text file

OCLC-Datasync-Postprocessing.py

- Dataframe
- Input (BibProcessingReport.txt)

  NJT54689 | 991146803405191 | 830722503 | 1232418344 | create

  Output (initial dataframe)

DATASYNC	**MMS ID**	**FILEINPUT**	**FILEOUTPUT**	**ACTION**
NJT54689	991146803405191	830722503	1232418344	create

- If FILEOUTPUT is null, replace with FILEINPUT
- Initial comparison (**FILEMATCH**) between FILEINPUT and FILEOUTPUT

DATASYNC	MMSID	FILEINPUT	FILEOUTPUT	ACTION	**FILEMATCH**
NJT54689	991146803405191	830722503	1232418344	create	**FALSE**

- Take all values from FILEOUTPUT and query WorldCat using WorldCat Metadata API

OCLC-Datasync-Postprocessing.py

- Use FILEOUTPUT to query WorldCat for:
  - OCLC# (currentOclcNumber)
  - Holdings status (holdingCurrentlySet)
- Append returned values to dataframe

DATASYNC	MMSID	FILEINPUT	FILEOUTPUT	ACTION	FILEMATCH	**BATCH**	**HELD**
NJT54689	991146803405191	830722503	1232418344	create	FALSE	**1232418344**	**FALSE**

- Add two comparisons: FILEINPUT==BATCH, FILEOUTPUT==BATCH

DATASYNC	MMSID	FILEINPUT	FILEOUTPUT	ACTION	FILEMATCH	BATCH	**ORIGBATCHMATCH**	**NEWBATCHMATCH**	HELD
NJT54689	99118...	830722503	1232418344	create	FALSE	1232...	**FALSE**	**TRUE**	FALSE

- Output dataframe to csv (oclc_result.csv) to downloads folder

oclc_result.csv

- Generally looking for FALSE values
- Held: If not held in OCLC (FALSE), confirm in local ILS/LSP
  - Two records with same OCLC#, one is deleted>>>unheld in WorldCat
- Comparisons: FILEMATCH, ORIGBATCHMATCH, NEWBATCHMATCH
  If FALSE, take any number of actions:
  - Examine records between Alma and WorldCat
  - Batch new OCLC#s numbers back into Alma
  - Be sure holdings are correct (holdings set on new OCLC record does not mean holdings were removed on previous OCLC record)

Questions

bitterc1@tcnj.edu
https://github.com/colinbitter