Does Working in Batch Mean sacrificing quality metadata?

How tools like MarcEdit, OpenRefine, Excel, and Python can help improve access and discovery

Presentation for the ALCTS CaMMS Catalog Management Interest Group Meeting at ALA Annual 2019.
Jennifer M. Eustis

What I’ll Cover Today

Introduction

Types of Electronic Resources

Common Issues Encountered

Metadata Evaluation, Requirements & Meeting Those Requirements

Matching Potential Solutions to Common Issues

Examples

Access & Discovery

Takeaways

Introduction

- A little bit about myself
  - New to UMass Amherst
  - Have worked with electronic resources for many years in Voyager & Alma
- A tale of 5 institutions
  - Five College Consortium
- The story of the tower
  - UMass Amherst

Image 1

Workflow for electronic resources

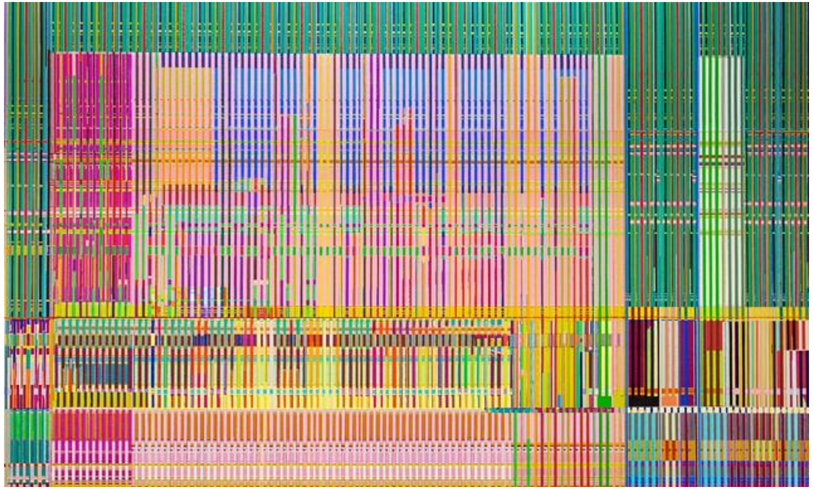


Image2

- The Workflow
  - Use CORAL to track and manage administrative data
  - Use SFX, HLM, EDS, & Aleph to either enable electronic packages and/or provide access/discovery to electronic resources
  - SFX, HLM, & EDS provide access and discovery to electronic packages, databases, journals
    - Resources can only be found in the discovery layer
  - Aleph provides access to those electronic packages, databases, journals not in SFX, HLM or EDS and individual titles if those title sets of MARC records are available for batch import into Aleph
    - Resources can be found in the OPAC and discovery layer

Types of Records that are Batch Loaded

Title sets of MARC records are loaded into Aleph monthly.

Titles are all electronic and include primarily streaming video, streaming audio, and electronic books.

Title sets of MARC records come from 2 sources:

Vendor

OCLC Knowledge Base Collections

Image 3

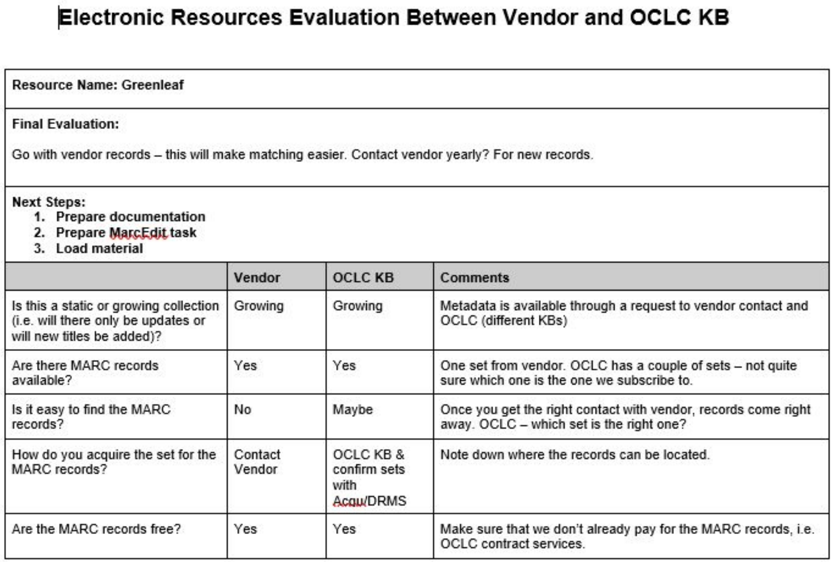
Common issues encountered

- URLs
  - Missing
  - Incorrect urls
  - Dead
  - Lead to wrong resource
- Missing information
  - URLs
  - Tites
  - Standard numbers
  - Publisher information

- Case
  - All uppercase
  - All lowercase
  - All sorts of cases
- Character encoding
  - MARC8 vs UTF-8
- Records
  - Electronic in file, OCLC master has a print record
  - Wrong records
  - Distinguishing correct set to match subscription
  - Inability to get a set your institution subscribes to

HOW DO YOU MEET YOUR MINIMUM REQUIREMENTS?

Meeting Minimal Metadata Requirements



Electronic Resources Evaluation Between Vendor and OCLC KB

Resource Name: Greenleaf			
Final Evaluation: Go with vendor records – this will make matching easier. Contact vendor yearly? For new records.			
Next Steps: 1. **Prepare documentation** 2. **Prepare MarcEdit task** 3. **Load material**			
	Vendor	**OCLC KB**	**Comments**
Is this a static or growing collection (i.e. will there only be updates or will new titles be added)?	Growing	Growing	Metadata is available through a request to vendor contact and OCLC (different KBs)
Are there MARC records available?	Yes	Yes	One set from vendor. OCLC has a couple of sets – not quite sure which one is the one we subscribe to.
Is it easy to find the MARC records?	No	Maybe	Once you get the right contact with vendor, records come right away. OCLC – which set is the right one?
How do you acquire the set for the MARC records?	Contact Vendor	OCLC KB & confirm sets with Acqu/DRMS	Note down where the records can be located.
Are the MARC records free?	Yes	Yes	Make sure that we don't already pay for the MARC records, i.e. OCLC contract services.

Requirements depend on:

- Your users
  - How do they search electronic resources?
  - What is the primary access/discovery point? (Discovery, catalog, A-Z lists)
  - What information do your colleagues need?
- Your discovery solution
  - What do you need to consider?
- Your catalog
  - Does this interfere with discovery or help?
- Best practices & national standards

Reality of Meeting Requirements

Vendor & OCLC Knowledge Base Collections need to be massaged. Using my evaluation, I assign one of 3 levels to the level of messaging needed:

- Low

  The set needs minimal cleanup so that it meets local needs for access, discovery, and best practices. Typically this is handled through a single MarcEdit Task and a visual spot check. The visual spot check is to check URLs, local fields (949), and sample 856s. The spot check can be done in Excel using Highlight Cell Deduplication or OpenRefine.

  Example: eDuke Latin American Studies (OCLC KB Collection) / Document without shelves (Marcive)

Reality of Meeting Requirements Continued

- Medium

  This set needs some extra work. There is the work to ensure that it meets minimal requirements for access, discovery, and best practices. This set might also need its own MarcEdit task or an additional one.

  More time needs to be spent on the URLs.

  Example: O'Reilly Safari Online Learning Platform (Vendor Provided) / NAXOS (OCLC KB Collection)

Reality of Meeting Requirements Continued

- High - Very High

This set requires significant cleanup. First it’s necessary to ensure the set meets minimal standards. Then it is necessary to check URLs in particular. An option is using Python to check not only for status of a URL but whether it leads to the resource. It is important to ask if the time needed to enhance this set is worth the effort.

Example: TRAIL - Technical Reports of archives and image library (OCLC Query Collection)

Excel vs OpenRefine

Excel

Excel has the ability to separate data into separate columns, highlight duplicate cells, and if you know visual basic macros, mundane tasks can be easier.

It is good for small sets that need to be spot checked.

It's difficult with large sets or when you have to make changes based on conditions.

OpenRefine

OpenRefine has all these abilities of Excel but in my mind is easier to see thanks to its facet function and tools to work with cells and columns.

It is good for large sets to be spot checked.

If you don't know jython, making edits based on conditional logic can be difficult

MarcEdit Find All results can be copied to the clipboard as a tab delimited file. This can be copied as a tsv in OpenRefine or Excel.

Excel, OpenRefine, And Python

Excel and OpenRefine

These are excellent tools for:

- Spot checking
- Moving data into separate columns
- Finding and replacing data
- Finding duplicates
- Determining trends

Python

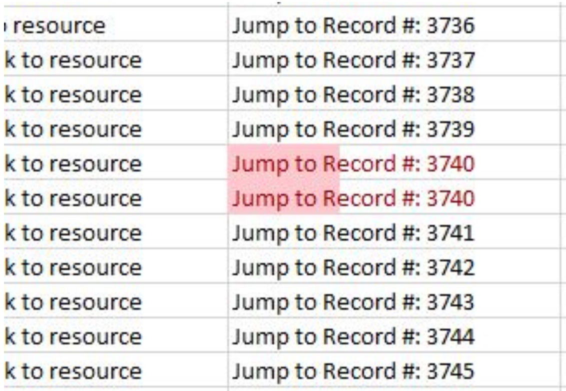
This is useful when:

- Conditional logic is needed
- Checking URLs

Examples

Excel

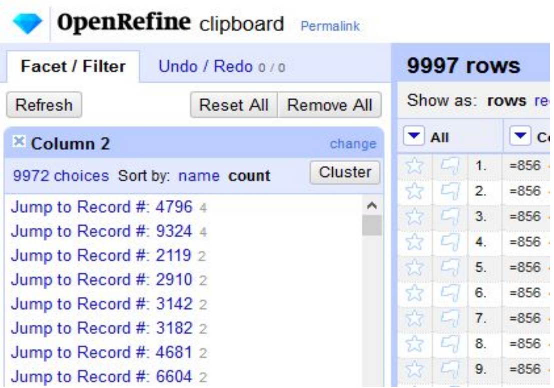
Conditional Formatting-> Highlight Duplicate Values



resource	Jump to Record #: 3736
k to resource	Jump to Record #: 3737
k to resource	Jump to Record #: 3738
k to resource	Jump to Record #: 3739
k to resource	Jump to Record #: 3740
k to resource	Jump to Record #: 3740
k to resource	Jump to Record #: 3741
k to resource	Jump to Record #: 3742
k to resource	Jump to Record #: 3743
k to resource	Jump to Record #: 3744
k to resource	Jump to Record #: 3745

OpenRefine

Facet by text-> Sort by count

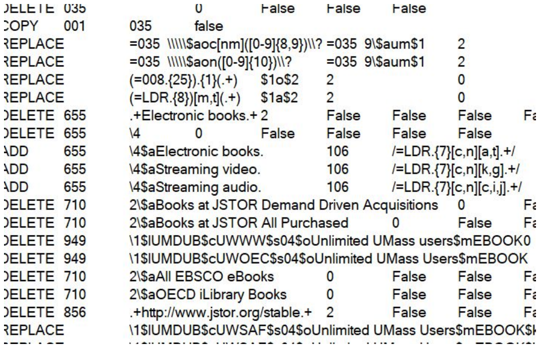


Jump to Record #: 4796 4
Jump to Record #: 9324 4
Jump to Record #: 2119 2
Jump to Record #: 2910 2
Jump to Record #: 3142 2
Jump to Record #: 3182 2
Jump to Record #: 4681 2
Jump to Record #: 6604 2

1.	=856
2.	=856
3.	=856
4.	=856
5.	=856
6.	=856
7.	=856
8.	=856
9.	=856

MarcEdit Tasks

Triage file with streaming video, streaming audio, and ebooks

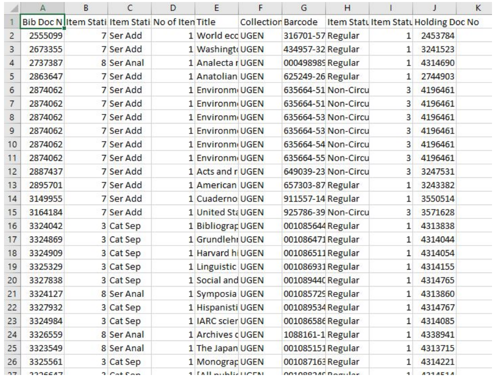


```
ELETE  035              0          False      False      False
OPY    001      035     false
EPLACE          =035  \\\\\$aoc[nm]([0-9]{8,9})\\? =035  9\$aum$1      2
EPLACE          =035  \\\\\$aon([0-9]{10})\\?      =035  9\$aum$1      2
EPLACE          (=008.{25}).{1}(.+)     $1o$2      2                   0
EPLACE          (=LDR.{8})[m,t](.+)     $1a$2      2                   0
ELETE  655      .+Electronic books.+ 2             False      False      False      Fa
ELETE  655      \4              0       False      False      False      False
DD     655      \4$aElectronic books.              106        /=LDR.{7}[c,n][a,t].+/
DD     655      \4$aStreaming video.               106        /=LDR.{7}[c,n][k,g].+/
DD     655      \4$aStreaming audio.               106        /=LDR.{7}[c,n][c,i,j].+/
ELETE  710      2\$aBooks at JSTOR Demand Driven Acquisitions          0          Fa
ELETE  710      2\$aBooks at JSTOR All Purchased         0          False      Fa
ELETE  949      \1$lUMDUB$cUWWW$s04$oUnlimited UMass users$mEBOOK0
ELETE  949      \1$lUMDUB$cUWOEC$s04$oUnlimited UMass Users$mEBOOK
ELETE  710      2\$aAll EBSCO eBooks       0          False      False      Fa
ELETE  710      2\$aOECD iLibrary Books    0          False      False      Fa
ELETE  856      .+http://www.jstor.org/stable.+    2  False      False      Fa
EPLACE          \1$lUMDUB$cUWSAF$s04$oUnlimited UMass Users$mEBOOK$k
```

Python Example: Create Aleph Bibliographic and Holdings Records Sys Numbers

CSV Incoming Data

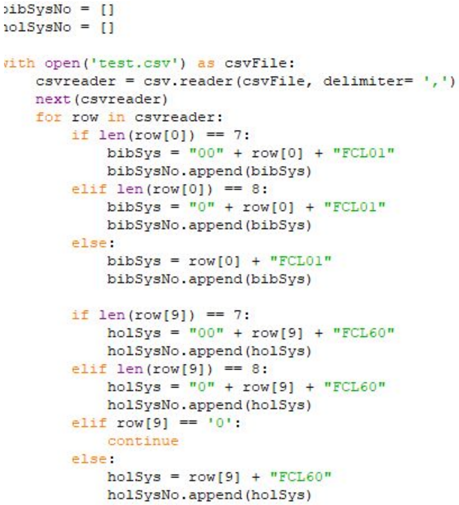
Has both bib and holdings sys numbers but not in format accepted by our ILS. Example: 2555099 needs to be 002555099FCL01



	A	B	C	D	E	F	G	H	I	J	K
1	Bib Doc N	Item Stati	Item Stati	No of Iten	Title	Collection	Barcode	Item Statu	Item Statu	Holding Doc No	
2	2555099	7	Ser Add	1	World ecc	UGEN	316701-57	Regular	1	2453784	
3	2673355	7	Ser Add	1	Washingt	UGEN	434957-32	Regular	1	3241523	
4	2737387	8	Ser Anal	1	Analecta r	UGEN	000498989	Regular	1	4314690	
5	2863647	7	Ser Add	1	Anatolian	UGEN	625249-26	Regular	1	2744903	
6	2874062	7	Ser Add	1	Environm	UGEN	635664-51	Non-Circu	3	4196461	
7	2874062	7	Ser Add	1	Environm	UGEN	635664-51	Non-Circu	3	4196461	
8	2874062	7	Ser Add	1	Environm	UGEN	635664-53	Non-Circu	3	4196461	
9	2874062	7	Ser Add	1	Environm	UGEN	635664-53	Non-Circu	3	4196461	
10	2874062	7	Ser Add	1	Environm	UGEN	635664-54	Non-Circu	3	4196461	
11	2874062	7	Ser Add	1	Environm	UGEN	635664-55	Non-Circu	3	4196461	
12	2887437	7	Ser Add	1	Acts and r	UGEN	649039-23	Non-Circu	3	3247531	
13	2895701	7	Ser Add	1	American	UGEN	657303-87	Regular	1	3243382	
14	3149955	7	Ser Add	1	Cuaderno	UGEN	911557-14	Regular	1	3550514	
15	3164184	7	Ser Add	1	United Sta	UGEN	925786-39	Non-Circu	3	3571628	
16	3324042	3	Cat Sep	1	Bibliograp	UGEN	001085644	Regular	1	4313838	
17	3324869	3	Cat Sep	1	Grundlehr	UGEN	001086471	Regular	1	4314044	
18	3324909	3	Cat Sep	1	Harvard hi	UGEN	001086511	Regular	1	4314054	
19	3325329	3	Cat Sep	1	Linguistic	UGEN	001086931	Regular	1	4314155	
20	3327838	3	Cat Sep	1	Social and	UGEN	001089440	Regular	1	4314765	
21	3324127	8	Ser Anal	1	Symposia	UGEN	001085729	Regular	1	4313860	
22	3327932	3	Cat Sep	1	Hispanisti	UGEN	001089534	Regular	1	4314767	
23	3324984	3	Cat Sep	1	IARC scier	UGEN	001086586	Regular	1	4314085	
24	3326559	8	Ser Anal	1	Archives c	UGEN	1088161-1	Regular	1	4338941	
25	3323549	8	Ser Anal	1	The Japan	UGEN	001085151	Regular	1	4313715	
26	3325561	3	Cat Sep	1	Monograp	UGEN	001087163	Regular	1	4314221	

Python File

Uses conditional logic to create the correct format for the number



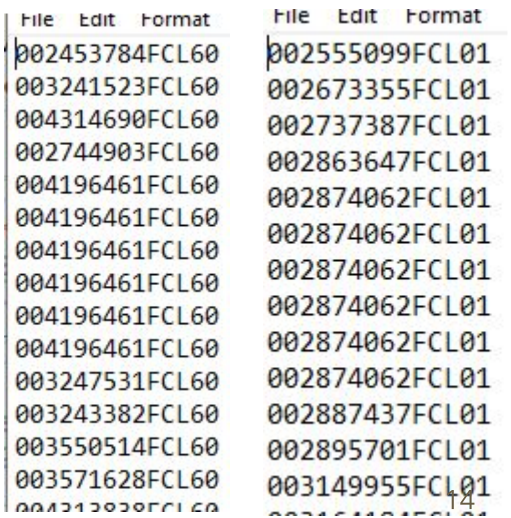
```python
bibSysNo = []
holSysNo = []

with open('test.csv') as csvFile:
    csvreader = csv.reader(csvFile, delimiter= ',')
    next(csvreader)
    for row in csvreader:
        if len(row[0]) == 7:
            bibSys = "00" + row[0] + "FCL01"
            bibSysNo.append(bibSys)
        elif len(row[0]) == 8:
            bibSys = "0" + row[0] + "FCL01"
            bibSysNo.append(bibSys)
        else:
            bibSys = row[0] + "FCL01"
            bibSysNo.append(bibSys)

        if len(row[9]) == 7:
            holSys = "00" + row[9] + "FCL60"
            holSysNo.append(holSys)
        elif len(row[9]) == 8:
            holSys = "0" + row[9] + "FCL60"
            holSysNo.append(holSys)
        elif row[9] == '0':
            continue
        else:
            holSys = row[9] + "FCL60"
            holSysNo.append(holSys)
```

Results

2 texts files one for holdings (FCL60) and one for bib records (FCL01)



```
002453784FCL60
003241523FCL60
004314690FCL60
002744903FCL60
004196461FCL60
004196461FCL60
004196461FCL60
004196461FCL60
004196461FCL60
004196461FCL60
003247531FCL60
003243382FCL60
003550514FCL60
003571628FCL60
```

```
002555099FCL01
002673355FCL01
002737387FCL01
002863647FCL01
002874062FCL01
002874062FCL01
002874062FCL01
002874062FCL01
002874062FCL01
002874062FCL01
002887437FCL01
002895701FCL01
003149955FCL01
```

Access & Discovery

Access & Discovery are at the heart of all this work. How users and colleagues access and discover these resources are crucial aspects to formulating metadata requirements and deciding best ways to prepare files for batch load.

Examples:

- Local field 949 Subfield k
- 655 _ 4 $a Electronic books.
- Field 856 subfield z

Image 4

Takeaways

Saying Yes to new tools doesn’t mean putting old tools away. Use the tool or method you’re comfortable with and Experiment with what you feel you can handle.

Not all sets are created equal. Evaluate metadata quality based on your requirements and record decisions, tool(s) to apply to the set, time to process each set.

Say No to sets that don’t meet your requirements. Examples for UMass Amherst include LION & HistoryMakers.

Takeaways

Set goals to learn a new tool. It doesn’t have to be the entire project. Take a piece and use the new tool while relying on the tools you already know.

Get a sense of which tool works in which situations. Are you dealing with a hammer or screwdriver?

Don’t sacrifice quality just to get any data in your system! This will work against access and discovery.

Be kind and patient with yourself while you learn.

Questions?

jeustis@umass.edu

References

MarcEdit: https://marcedit.reeset.net/

OpenRefine: http://openrefine.org/

Data Carpentry Introduction to OpenRefine: https://datacarpentry.org/OpenRefine-ecology-lesson/

Python: https://www.python.org/

Code Academy & Python: https://www.codecademy.com/learn/learn-python-3

W3schools & Python: https://www.w3schools.com/python/

Pymarc: https://github.com/edsu/pymarc

Introduction to Pymarc Session I: http://www.ala.org/alcts/confevents/upcoming/webinar/101817

Introduction to Pymarc Session II: http://www.ala.org/alcts/confevents/upcoming/webinar/102517

OCLC API & MarcEdit Integration: https://help.oclc.org/Metadata_Services/WorldShare_Collection_Manager/Troubleshooting/How_do_I_set_up_Marc_Edit_OCLC_Integration

Z39.50 & MarcEdit Operations: https://marcedit.reeset.net/batch-marc-record-retrieval-using-z39-50

---> For the Z39.50: (Remember to add your OCLC Authorization & password in the z39.50 settings)

Image Credits

Image 1: Surkam, Jim. “5_courthouse went up in 1836”. CC BY-NC 2.0, Retried from https://www.flickr.com/photos/jimsurkamp/15102453307/

Image 2: Lee, See-min. “Painting by LIU Wei: Truth Dimension No. 7, 2013 (oil on canvas)”. CC BY-NC 2.0. Retreived from https://www.flickr.com/photos/seeminglee/8921779798/

Image 3: Beckwith, Michael D. “Kelvingrove Art Gallery and Museum”. CC0 1.0 Universal. Retrieved from https://www.flickr.com/photos/118118485@N05/18551513659/

Image 4: Gage, Tim. “MacLeod’s Books”. CC BY-SA 2.0. Retrieved from https://www.flickr.com/photos/timg_vancouver/39363030394

Links to Resources on Evaluating Electronic Resources

- Rutgers: “Evaluating Bibliographic Records Sets for Electronic Resources”, Rev. 2011, Retrieved from
  - https://www.libraries.rutgers.edu/rul/staff/technical_services/cataloging/eval_bib-record_sets.pdf
- Akron, Ohio: “Managing Electronic Resources Collections and Batch Loads”, 2018, Retrieved from
  - https://www.ohiug.org/uploads/6/1/3/5/61351715/2018_managing.pdf
- CARLI: “Batch Loading Bibliographic Records for Electronic Resources”, Rev. 2013, Retrieved from
  - https://www.carli.illinois.edu/sites/files/i-share/documentation/eresbatch.pdf
- Code4Lib Article: “Leveraging Python to improve metadata ebook selection, ingest, and management”, 2018, Retrieved from
  - https://journal.code4lib.org/articles/12828