Secrets of the Save File:

Using Connexion and Microsoft Access for Efficient Batch Cataloging Projects

1. [Title slide]

Hello and welcome.

Slides and transcript are available at tiny.cc/MOUGSaveFile.

I'm going to be talking today about using Connexion and Microsoft Access for efficient batch cataloging projects. Along the way you will get to learn more about the inner workings of Connexion's local save file. To start off, I'd like to introduce myself and give you a little more context about how I ended up here, talking about this topic today.

2. About Me

My name is Rebecca French, and I began my career in libraries as a music cataloger. My first full-time position was as the Music & Media Metadata Specialist at James Madison University, where, as you can probably guess, I did music and media cataloging. In that role I was also responsible for batch loads for e-resources, which allowed me to become more familiar with ways to work with records in larger quantities. I then moved into the position I hold now as Metadata Analyst Librarian, where I design workflows and tools to efficiently work with metadata at scale. I get to work with everything from digital and archival collections to e-resources and traditional MARC cataloging.

The reason I'm giving you that background on my career is that this presentation describes a process I developed while in my music cataloging job, before I had developed many of the technical skills I now use in my current role. So while now I would approach a batch cataloging project using tools like OpenRefine, APIs, and scripting, I recognize that not everyone has those skills or tools available to them. There's definitely still a place for methods that have a much lower barrier to access and use only commonly available software, like the one I'm going to share today.

3. "See, I've got a really good system…" [Workaround XKCD comic]

I mentioned workflow development as part of what I do in my job, and it often feels something like this. The person in the comic is saying, "See, I've got a really good system: if I want to send a YouTube video to someone, I go to File --> Save, then import the saved page into Word, then I go to "Share This Document" and under "Recipient" I put the email of this video extraction service…" Like the friend in the comic, we understand that this is a pretty convoluted and roundabout way of accomplishing this task. I work to replace these types of workflows with something a bit more streamlined, or at least remove the unnecessary steps to make things more efficient.

The workflow I'm about to describe might seem very similar to this one (it does have quite a few steps), but my hope is that I'm able to show you how we're actually avoiding a situation like this by letting the computer do some of the work for us.

4. The Jazz LP Project [section header]
I've structured this talk as a walkthrough of our project, starting with an overview of the collection and why we needed to catalog it. Then I'll discuss each of the steps we took and I'll also take some time in the middle to go over some features of the local save file database in more detail.

5. JMU's Jazz LP Collection
Our jazz LP collection was a donation from the local public radio station prior to 1990. It contains just under 3,500 LPs, which are arranged in alphabetical order by performer or composer. At some point after acquisition, a brief record for each item was added to our library catalog. These brief records contained title, performer or composer, and publisher name and number. Each item was also assigned an accession-based call number.

6. The Jazz LP Project
This minimal metadata meant that these items weren't very discoverable in the catalog, and our Music Library also had closed stacks, so patrons weren't able to find these materials through browsing. We wanted to improve the discoverability of these LPs by enhancing our existing metadata with records from OCLC. In 2015, I started working on this project and got about a third of the collection cataloged.

7. The Jazz LP Project, On Hold
But then a number of things changed in our organization that resulted in us putting this project on hold. The Music Library opened its stacks, so it was now less of a priority to catalog these items for discoverability. I also changed jobs, moving into my current role, and most of the projects I was working on as Music & Media Metadata Specialist were put on hold until we could fill that position.

8. The Jazz LP Project, Round 2
In 2017, we hired a new music cataloger, Allison Lyttle. Then, in early 2018, we found out that the LPs were going to be moved to off-site storage, which meant that again the collection would not be browsable. This move also gave us a deadline – if we were going to catalog these it would be best if we could do that before the

summer move. So at this point I trained Allison on how to use this method and she completed the rest of the project.

9. Planning the Batch Workflow

Back in 2015 when I was figuring out how we could catalog these items efficiently, I started by considering the existing metadata that was already recorded. It didn't follow any consistent rules, and there were definitely some typos and other errors, but we were able to construct search strings from this metadata and perform batch searches in Connexion. At that point, we still needed to evaluate the search results with items in hand in order to select the correct record for each search.

10. How to Match?

Once we had a set of records, though, we needed to figure out a way to match them with the bib numbers of our existing brief records in order to import the new records into the catalog. Ideally, this is something that wouldn't need to be done by hand.

The bib numbers were connected to our existing metadata that we had exported from the catalog, and that was connected to the search strings that we created from that metadata in the same spreadsheet. The catalog metadata wasn't a good match for fields in the OCLC records in most cases due to differences in descriptive practices and punctuation.

We had used the search strings to retrieve each record in the save file, so what was needed was a way to see which search string was used to retrieve each save file record.

11. How To Match

This is where Microsoft Access and the local save file database come into play. The database contains all the records in the local save file, along with the search string used to retrieve that record and a lot of other information.

Before I go into more detail about the steps to take to access the information in the save file database, I'm going to give a brief overview of the process so you have an idea of where we're headed.

12. [Workflow diagram]

It starts with performing a batch search in Connexion, and the results are saved to the local save file. You can then winnow the save file results to eliminate records you don't want. This step is optional, as there are also later opportunities to select or discard records based on other criteria.

Then you open the local save file database in Microsoft Access, import any other data you might need to work with, and create relationships to specify how that data links to the items in the save file. Finally, you can run queries to further narrow your result set and export the data you need to continue your project.

13. Batch Search
To begin our jazz LP project, we exported the metadata that was in the brief catalog records and then used that to create a search string for each item.

Each search string included the material type “lps” and English as the language of cataloging. You can see some examples of search strings on the slide. We used the publisher numbers when those were available, sometimes with the publisher as an additional search term. For items without a publisher number, we typically searched on the title and publisher.

We used a high maximum number of matches to download so we’d be able to review all the potential records for each item, and we performed between 50 and 300 searches in each batch.

14. Spreadsheet with batch search strings
Here’s what our spreadsheet looked like, with bib numbers and the metadata from the catalog on the left and the search strings we created on the right in column E. [pause]

15. Winnow Save File
The next step after performing these batch searches was to winnow the results. We compared each item to the records returned for its search and selected the best record that met our criteria. Even if there was only one search result, we still compared it to the LP to make sure we had the correct record.

To simplify this step, and because we weren’t taking the time for full cataloging of each item, we only considered certain fields when evaluating records. These were the dates in the 008, the 028, 245, 260 or 264, and the 300 field. All the other records except the best match were deleted from the save file.

I want to note here that depending on your particular project, this step may not be necessary. One benefit of working with the save file database in Access is that you can use queries to do some of the record selection instead of manually reviewing the save file in Connexion. Which of these you use depends on what criteria you’re using as well as your level of comfort creating queries in Access, which I’ll talk more about later.

16. Open Save File in Access
Everything so far has likely been familiar if you’re a Connexion user; now this is where it gets interesting!

The local save file is just a database file saved on your computer; you can see the default location here on the slide. This is a hidden folder, so you may have to change your Windows folder options in order to see it. If you used the default local save file, the file name you're looking for is DefaultBib.bib.db.

17. Local save file location [screenshot]
I recommend you make a copy of this file to work with instead of using this one that Connexion uses, because we will be making some changes to it. Then you'll open the copy in Microsoft Access.

18. Local Save File Database [section header]
I'm going to pause in our workflow for a bit to talk about some of the features of this database in more detail. There are 15 tables in this database; I'm going to talk about the three that are potentially the most useful. We didn't use all of these in this particular project, but this might give you some ideas for other ways you could use the information in the save file database.

19. tblSaveFile
The first table is named tblSaveFile. It contains the fields that you see in the Connexion client when you're looking at a list of records in the local save file. tblSaveFile also has some additional fields that don't appear in that list, information that is only visible in Connexion when you're looking at an individual record. One such field is Holdings Count, the number of libraries that have holdings set on a particular record. Information on GLIMIR clusters is another example.

20. [tblSaveFile chart 1]
Here's a partial list of the fields in tblSaveFile, along with the corresponding labels used in Connexion. We're seeing the save file number, 1XX, 245, OCLC number, format. Nothing too surprising here yet.

21. [tblSaveFile chart 2]
The table on this slide and the previous slide include all the fields in the Connexion local save file results list as well as the fields from tblSaveFile that don't appear in Connexion except when accessed from individual records (Holdings Count and the fields related to GLIMIR clusters).

22. Connexion local save file [screenshot]
Here's our save file in Connexion. This probably looks very familiar to you. We have the contributor, title, OCLC control number, date, call number, and when each record was added to the save file. None of these items are held by our library. They're all sound recordings, and some of the records are Library of Congress cataloging, as indicated by DLC.

23. tblSaveFile [screenshot]
For comparison, here's the same local save file when opened in Access, where it's called tblSaveFile. I've hidden some of the less interesting fields here to save space. We see all of the same fields that were visible in Connexion. [brief pause]

24. tblSaveFile [screenshot, highlighted]
There's also the Holdings Count field which doesn't display in Connexion, which tells us how many libraries hold the item represented by each record.

So that's tblSaveFile.

25. tblBatchSearch
Another table in the local save file database is called tblBatchSearch. It contains information about the batch searches that were used to save records to the local file. In addition to the OCLC number, title, and ID or save file number, this table also contains a field with the search strings used in a particular batch and the number of records returned for each search. There is also a Result field that indicates whether records were found or if there were too many matches or another error.

26. tblBatchSearch [screenshot]
Here's what tblBatchSearch looks like in Access. This is very similar to what appears in the batch search reports in Connexion, although the information in those reports isn't really in an actionable form like it is here. On the left you can see the search key that was used to retrieve each record listed on the right.

27. tblBatchSearch [screenshot, highlighted]
For example, you can see that this search for the publisher ECM and title Great Pretender returned four records. [pause]

28. tblRecords
The last table I'm going to talk about is tblRecords, which contains the contents of each record in the save file and the ID or save file number. The records are stored in OCLC's Common Data Format XML. It's difficult to pull info from specific MARC fields because Access isn't really designed to work easily with XML, although for another project I had success looking for a particular keyword that could have appeared anywhere in the record.

29. CDF XML record from tblRecords [screenshot]
tblRecords isn't all that interesting to look at in Access, so here's a screenshot of one of the CDF XML records. If any of you are familiar with MARCXML, you'll see that this looks a bit different, but you can still pick out the fields, indicators, and field contents. [pause]

30. The Jazz LP Project: Back to the workflow… [section header]
Now that we've looked at tblSaveFile, tblBatchSearch, and tblRecords, we're ready to jump back into the workflow. Where we left off, we had done our batch search, winnowed the save file, and opened the database in Microsoft Access.

31. Import External Data
Now we were ready to add in our local data, including the bib numbers, so it could be matched up with the records in the save file. Earlier I mentioned that the search strings were the key to matching up the data from our brief catalog records with what's in the save file. The search strings are a field in tblBatchSearch, as we just saw, and we also have those in our spreadsheet along with the bib numbers.

We imported the spreadsheet as a new table in Access; I called it SierraData. Usually it works just fine to accept the default options in the import wizard.

32. Spreadsheet added as a new table in Access [screenshot]
This is what our data looked like after we imported it as a new table in Access. [pause]

33. Create Relationships
The next step is to create relationships between different tables so Access can tell which fields they have in common. The two relationships we needed to create for this project are linking tblSaveFile and tblBatchSearch on the ID field, which is the save file number, and also linking tblBatchSearch and our SierraData on the search string field.

34. Relationships linking tblSaveFile, tblBatchSearch, and SierraData [screenshot]
This is what it looks like after you've created the relationships. You can see that this gave us a way to connect what's in tblSaveFile, through tblBatchSEarch, to the data from our ILS, which was our goal.

35. Create Query and Export
Our next step in the workflow is to create a query that pulls together the data we need from various tables. As I mentioned earlier, you can also use queries at this point to further refine your result set. One example of that would be if you wanted to select the record that had the highest number of holdings for each search. Because we already winnowed our results earlier, using Connexion, our query for this project only needed to pull together the OCLC numbers and bib numbers.

Access provides three different ways to create a query. The Query Wizard walks you through a series of options, and the Query Design View gives a little more control over the query than the Wizard does. You can also create a query directly in SQL if you're familiar with that query language.

36. Query design view [screenshot]
Here's what the query design view looks like with the query I created for this project. The diagram at the top shows the three tables that this data will be drawn from and how the tables are linked together as a result of the relationships we set up in the previous step. The ID field is the field in common between tblSaveFile and tblBatchSearch, and the search string or search key links tblBatchSearch with the SierraData table. At the bottom, you can see the three fields that will be included in the query: the search key, the OCLC control number, and the Sierra bib record number.

37. Query results [screenshot]
When we ran this query, we got results like this. For each of the searches we did, we now have the OCLC number and the bib number of the record in our ILS that it will be replacing.

You can export these query results in a number of formats, including text, XML, or an Excel spreadsheet. We exported a spreadsheet for this project.

38. Finishing Up
We turned this spreadsheet into short bib records using MarcEdit. These short records had just the OCLC number in the 001 field and the bib number in the 907. We then merged the bib numbers from these short records into the full OCLC records by matching on the OCLC number. We made a few additional edits to the records, things like deleting call numbers. To load the records into our catalog, I created a custom load table that matched on the bib number, protected our local call number field from being overlaid, and overlaid all other fields in the brief catalog records with the full records from OCLC.

39. Results and Benefits [section header]
Now that I've explained how we did this project, maybe you're wondering how successful it was, so I'd like to share our results and some of the benefits of using this workflow.

40. Records Found
Out of the 3,484 items we searched for, we found records for 95% of them, which leaves 180 that will need individual cataloging. Some of these 180 items might require original cataloging because there's no record for them in OCLC; others might have a record in OCLC that just wasn't found by the search we used.

Most of these LPs were commercial releases, so I don't think it's too surprising that we found records for such a large percentage of them. It's also good to know, though, that the metadata we had to create search strings from was good enough. Some items might not have been found due to incorrect metadata rather than a matching record not existing.

41. How Long?

We found we could process somewhere between 30 and 80 LPs in an hour. That's the full process, from the initial batch search to loading the records into our catalog. I found that the most time consuming part of the process was trying to get the LPs out of their sleeves in order to check them against the OCLC records.

42. Workflow Benefits

We found that this workflow worked well for our project for a number of reasons. It was more efficient to use batch searching and only look at a limited number of fields when evaluating records. Using the save file database to match the OCLC records with our local data allowed us to avoid adding the local information manually or on a record-by-record basis, which increased our accuracy and saved time. Another decision that supported the accuracy of the work was to use the load table to retain the call numbers in existing records, so that was one less piece of data that needed to be moved around. We also found that working in batches helped provide motivation by creating smaller milestones throughout the project.

43. Method Benefits

This method of directly accessing the database underlying Connexion's local save file provides an actionable way to use information that's either inconvenient to get at in Connexion or is not available to the user at all. It offers the ability to link Connexion's metadata to data from another source, as we did with our data from Sierra. It offers querying capabilities beyond what Connexion searches can do, through Access' built-in querying tools or use of SQL. You don't need any special software, and the most basic uses, like exporting a spreadsheet from the database, don't require any particular database skills. On the other hand, those that are more familiar with Access or SQL can use their skills to manipulate the data in more complex ways.

44. Other Projects

In addition to the jazz LPs, we've used this method for two other projects, both of which took advantage of some other options that we didn't need for the LP project.

For some of our e-book collections, we get records without OCLC numbers from vendors. The majority of the time there are a number of duplicate records in OCLC for these e-books, but the desired record is typically provider neutral and has vastly more holdings than any of the other records. So we do a batch search and then use the holdings count field in tblSaveFile to select the record with the greatest number of holdings for each item.

I also did another project to add records to our catalog for the Text Creation Partnership's XML-encoded transcriptions of the Early English Books Online corpus. Some EEBO titles had microform records in OCLC, others had records for the electronic resource, and some items had both. For this project I ended up migrating the database from Access to an SQLite database and running a series of SQL queries to select the desired record for each title. The EEBO records in OCLC did not include links to the TCP texts, so these were then added to the records.

45.Additional Resources

Both of these projects, along with the jazz LP project, are described in more detail in an article I wrote for the Code4Lib Journal, called "Direct database access to OCLC Connexion's local save file."

I've also put the detailed procedures we used for the jazz LP project online.

As I said at the beginning, I'm not claiming that this method is the best option across the board for batch cataloging. But there are certainly situations where it is a good option, depending on who can be involved in the project, the technical skills they have, and the software that's available. For example, if you want to get a count of the number of holding libraries for a list of titles, this is the only way to do that other than using OCLC APIs or looking each item up individually.

46."Only a few of us ever found it." [Undocumented Feature XKCD comic]

I doubt OCLC ever intended for users to directly access the database underlying the local save file (although they didn't have any problems with me doing it when I asked), but as long as we're still using Connexion it remains an option. Now that you're all in on the secret, I'd love to know if any of you end up using this undocumented feature at your own libraries.