code{4}lib
JOURNAL

WMS, APIs and LibGuides: Building a Better Database A-Z List

At the American University of Sharjah, our Databases by title and by subject pages are the 3rd and 4th most visited pages on our website. When we changed our ILS from Millennium to OCLC's WorldShare Management Services (WMS), our previous automations which kept our Databases A-Z pages up-to-date were no longer usable and needed to be replaced. Using APIs, a Perl script, and LibGuides' database management interface, we developed a workflow that pulls database metadata from WMS Collection Manager into a clean public-facing A-Z list. This article will discuss the details of how this process works, the advantages it provides, and the continuing issues we are facing.

by Veronica Ramshaw, Véronique Lecat and Thomas Hodge

The database A-Z list is a staple of many academic libraries' online presences, allowing students and faculty to quickly see which familiar databases are available and to directly access specific resources. Traditionally, our own database A-Z list has existed as two pages fed by the same XML document: Databases by Title and Databases by Subject. According to our site's Google Analytics, these database pages are very important to our users as they represent the third and fourth most visited pages on our website.

After the American University of Sharjah (AUS) Library migrated its Integrated Library System (ILS) and discovery interface to OCLC WorldShare Management System (WMS) in 2016, we needed to find a new way to synchronize database data. This article will outline how we automated this process using WMS APIs and LibGuides.

Old procedure

Prior to 2016, the flow of information from our Millennium system to the two database pages was automated. Millennium Resource and License records contained the necessary metadata such as database name, description, URL, and subject keywords. The data was harvested from a list of all Resource and License record numbers via /xrecord, a facility of Millennium's public access catalog that displayed the records in XML format. A script harvested these records into a Dublin Core XML serialized file. This was done whenever changes were made to the Resource or License records. The resulting file, titled databases.xml, was then placed on our website's server where the two public database pages, Databases by Title and by Subject, would be dynamically generated from that data.

In this way, we were able to synchronize the database information stored in Millennium to the pubic database pages, ensuring information was kept up-to-date and consistent.

After the WMS Migration

In 2015, the AUS Library began the process of migrating from Millennium to WMS and in 2016 had stopped using Millennium completely. These changes meant it was no longer possible to generate our public database pages using the old procedure. As a result, any changes to the details of a database, such as a new name, modified URL, or updated description, had to be changed manually in WMS Collection Manager, and in the databases.xml file. In addition to the amount of time this took, the duplication of work and dependence on manual changes created the potential for human error to introduce inconsistencies between the systems.

Our new discovery interface, WorldCat Discovery, does provide a list of active collections through the A-Z Titles interface, so we considered it as potential replacement for our Database A-Z. Unfortunately this tool has a number of problems. The largest problem is the sheer volume of duplication due to the fact that it lists every collection your library has activated in WMS Collection Manager. For example, we have almost 200 individual collections of Springer eBooks for each year and each subject we've purchased (Springer Behavioral Science and Psychology eBooks 2016, Springer Behavioral Science and Psychology eBooks 2017, Springer Economics and Finance eBooks 2017, and so on) which makes for a cumbersome list to browse through. Another problem is that we have a number of collections active in Collection Manager that we wouldn't consider databases. For example, we have access to a handful of journals from the University of Chicago Press from individual subscriptions, but this is far to small to be considered a database. These problems made WorldCat Discovery's Journal Finder interface too cluttered and unwieldy to a useful replacement for our Database A-Z list.

Finding a New Way

Our goals for a new database A-Z list were that it should:

- **Be Automated:** To reduce the duplication of work and possibility of human error.

- **Be Clean:** Human readable with no duplicate content cluttering out the useful information.

- **Be Consistent:** Every platform (website, LibGuides, discovery interface) should have the same metadata. Users should see the same names for the same databases across every library platform to minimize confusion.

- **Maintain Continuity:** Similarities between the old A-Z database pages and the new one should help avoid user disorientation in the transition.

To begin we needed to get a thorough understanding of now WMS handles database data. It does this through two different modules: Collection Manager and License Manager. Collection Manager contains the metadata about the database (e.g. description, URL) as well as a list of every ejournal or ebook contained in the database and the date range covered. License Manager handles the renewal dates, prices, terms of use, and allows us to upload renewal documentation.

Next, we looked at using the LibGuides Database A-Z tool. For those unfamiliar with the product, LibGuides is used to make Subject and Course specific guides to library resources. Since our initial purchase of the LibGuides system, the vendor SpringShare added a public-facing Database A-Z, however we did not have any use for it up to this point because we were already generating our own Database A-Z from data harvested from Millennium. With this project, we decided to use the LibGuides Database A-Z page to replace our existing database lists. For the sake of clarity, in this article we will refer to it as the LibGuides Database A-Z.

New procedure

The following is an outline of how our new procedure works. It is divided into three sections: (1) The preparation work that needs to be done in WMS and LibGuides before the script is run ; (2) Running the script which exports database metadata from WMS License Manager and Collection Manager ; and (3) Loading the resulting file to LibGuides.

1. Preparation

    a. WMS License Manager

        i. License statuses must be Current (as opposed to Expired or Pending)

        ii. Licenses must contain the correct linked collections [see Figure 1]

    b. WMS Collection Manager

        i. Select a representative collection for each database. For example, we have 3 separate collections for ProQuest Ebook Central: our main subscription, our DDA subscription, and our perpetually owned titles. We selected one of these, our main collection, as the representative collection. This prevents unnecessary duplication in the database list.

        ii. Ensure the collection title is appropriate. This can be locally customized by the library. For example, we renamed Ebook Central Academic Complete as ProQuest Ebook Central

        iii. Ensure the collection has description text

        iv. Verify the Collection URL is correct

        v. Include the LibGuides content_id in the staff note field

        vi. Include the subject heading(s) in the staff note field [see Figure 2]

    c. LibGuides

        i. Clean up the existing database list in LibGuides. In our case, we had a lot of links that were saved as databases in LibGuides that we either changed to link assets, or deleted completely.

        ii. Ensure each database is saved in LibGuides and is assigned a content_id. The script only updates existing database assets, it does not create new ones.

        iii. Classify the Database Type as Abstract-only or Full-text, as needed.

2. Running the script [see Figure 3]

    a. Using the OCLC License Manager API, the script runs through all our licenses. For every license, it gets the license status (Current, Expired, or Pending) and the list of linked collections.

    b. Using the OCLC Collection Manager API, the script provides the details of the collections linked to the licenses, such as name, description, URL, proxy yes/no, content_id, and subject headings.

    c. The script searches for the linked collections via the Collection Manager API.

    d. It creates a brief delimited record for each license or collection which should be included in the A-Z list.

    e. It outputs a tab-delimited .txt file [see Figure 4]

3. Loading to LibGuides

    a. Check the tab-delmited .txt file to ensure data integrity.

    b. Go to LibGuides > Content > A-Z Database List

    c. Click "Import / Update Databases" and upload the file.

    d. All changes made in Collection Manager are now synced to the LibGuides Database A-Z [see Figure 5]
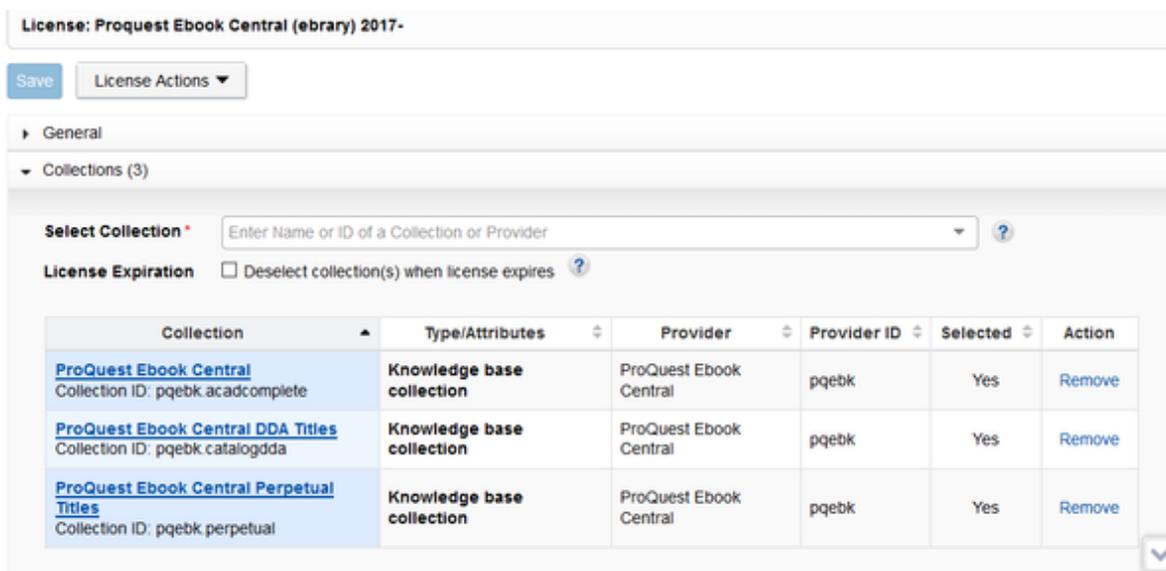
Figure 1.
License of ProQuest Ebook Central contains three linked collection. The script will look at the collections linked in every current license.

Figure 2.
Collection for ProQuest Ebook Central where we added a Description paragraph. In the Staff note field, we added a content_id that links it to it's corresponding LibGuides database asset. We also added subject headings.
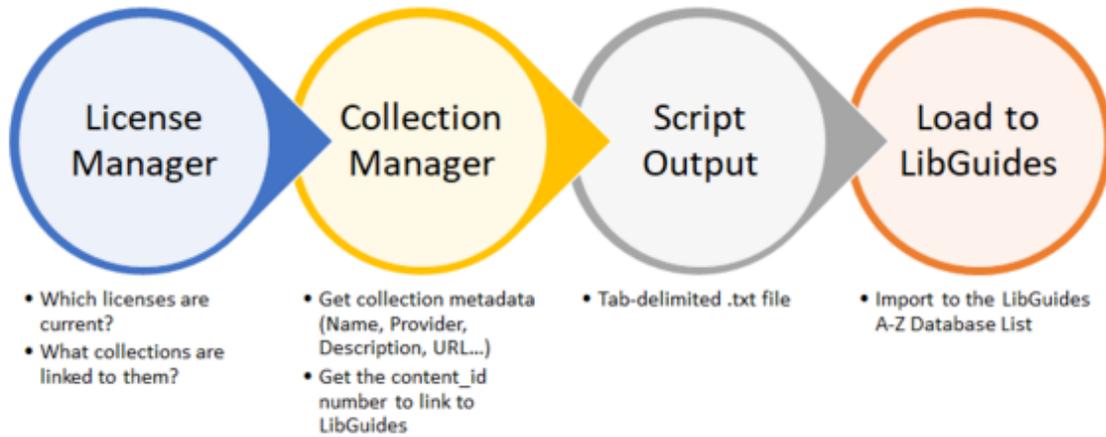
Figure 3.
The internal steps of the script which exports database metadata using the License Manager API and Collection Manager API, and outputs it as a tab-delimited file.

Figure 4. Script output (.txt file imported into Excel)

Figure 5. Our new Database A-Z page. Users can filter by subject or database type (abstract-only or full-text).

Pseudo Script

This Perl code snippet represents the core function of the script. The full script is available at: https://github.com/hodgenet/OCLC-LibGuides-Integration

Access to the Perl authenication modules is available by contacting the author at thom@hodgenet.com.

```
1    ## set these to defaults, modify as needed from results:
2    ## when total results falls below itemsPerPage, we have everything
3    while ($startIndex < $totalResults) {
4      ## get the parameters for License Manager API
5      my %queryParams = (
6        'itemsPerPage' => $itemsPerPage,
7        'startIndex'   => $startIndex,
8        'q'            => "licenseStatus:$licenseStatusFilter"
9      );
10     ## Call makeOCLCRequestURL function to make the URL correctly
11     my $requestURL = make_hru(\$apiURL, \%queryParams);
12
13     my $response = $ua->get(
14       $requestURL, 'Authorization' => $atAuthHeader
15     );
16
17     ## get the first bit of the license listing
```

```
18    my $apiRecord = $response->content;
19    ## parseLicenses gets some metadata for the License
20    ## and each of the collections linked to it
21    ($totalResults) = parseLicenses(\$apiRecord);
22    ...
23    foreach my $collection ($entry->child('content')
24                                     ->child('license')
25                                     ->child('collections')
26                                     ->child('collection')) {
27      my ($thisCollectionName, $thisCollectionProvider);
28      my $thisCollectionID = $collection->child('id')->value;
29
30      if (defined $collection->child('name')) {
31        $thisCollectionName = $collection->child('name')->value;
32      } else {
33        $thisCollectionName = 'No Name';
34      }
35
36      if (defined $collection->child('provider')) {
37        $thisCollectionProvider = $collection->child('provider')->value;
38      }
39
40      ## call getCollectionRecord
41      my $thisXMLCollectionRecord = getCollectionRecord(
42        \$thisCollectionID, \$thisCollectionName
43      );
44    }
45    ## increment the start index for next batch of licenses
46    $startIndex = $startIndex + $itemsPerPage;
47  }
```

Ongoing Issues

A major limitation of the current iteration of the script is that it doesn't alert us to the cause of errors. If the script encounters an error, we must hunt for the cause. The common problem we encounter if a collection was deleted in Collection Manager but was not also removed in License Manager, it will cause the current script to fail. Other errors arise from incorrect formatting of the content_id or subject headings, invalid characters, or a license status is not set to "current" in License Manager. We plan to improve the script so it can return human-readable error reports.

One feature of the LibGuides database A-Z tool is the ability to label certain databases as "Popular." We identified a few databases as popular after looking as usage numbers. Unfortunately the LibGuides import process overwrites whether a database is labelled "Popular" in LibGuides, meaning we have to re-label them manually every time we run an update.

Conclusion

After a few months spent developing this procedure, writing the script, and preparing our data, we finally launched our new LibGuides Database A-Z in late May 2018. Based on our experiences so far, we plan to further refine the script and procedure to improve error reporting, troubleshooting and further reduce the amount of manual work involved (such as re-labelling databases as "Popular").

This procedure and script were the products of necessity. After migrating to a new ILS, our old workflow was no longer viable. We were able to exploit the tools we had available — OCLC's APIs (which are free of cost), and the LibGuides Database A-Z — to devise a new procedure. In doing so, we were able to achieve the goals we set forth at the start: an automated procedure that produces a clean output, creates consistency, and maintains continuity. Other libraries using WMS and LibGuides should consider using a similar procedure to produce easily automated Database A-Z pages that work well for their patrons.

About the Authors

Veronica Ramshaw
Web Services Librarian
American University of Sharjah
vramshaw@aus.edu
https://orcid.org/0000-0002-4086-9419

Véronique Lecat
Cataloging and Metadata Librarian

American University of Sharjah.
vlecat@aus.edu
https://orcid.org/0000-0002-1373-8384

Thomas Hodge
Associate University Librarian for Technology and Technical Services
American University of Sharjah.
thodge@aus.edu
https://orcid.org/0000-0002-0026-2532

Subscribe to comments: For this article | For all articles