# Automate and Expedite

## Using Claude AI to Speed up an E-reserves Migration

*By Challen R. Wright and Shelly Johnson*

### INTRODUCTION

Content management system (CMS) migrations are quite common in academic libraries. Since CMSs store, showcase, and preserve digital collections to allow objects to be available to the public, migrat-ing systems is important to stay up to date with changing technologies and enhanced cyber security features. While many migrations are planned ahead of time, the University of Nevada, Reno came across a unique situation where a migration of our e-reserves platform was required with little warning and needed to be accomplished within a short time frame. This case study will review the ways in which the University of Nevada, Reno's Metadata Librarian and Metadata Technician used AI, specifically Claude, as a tool to expedite and automate parts of a large-scale migration of over 42,000 e-reserves objects and their metadata.

### BACKGROUND

The University of Nevada, Reno previously hosted our e-reserves platform on Ares, which was used to present resources for student access through the university's student course platform, Canvas. Day-to-day operations were managed by the Access Services department, while IT/Tech Support was managed through the Digital Services department. In September 2024, the system was compromised due to an attempted hack by an outside source. Once this event was identified by the University Libraries team, they quickly replicated the e-reserves structure in Box as a quick fix until a more permanent solution could be obtained. At that time, more than 100,000 digital objects (primarily PDFs of physical resources) existed in UNR's e-reserves platform.

The Libraries selected Leganto, a platform hosted and maintained by Ex Libris, as the successor for e-reserves content. The platform was more secure, maintained by a well-known vendor, and offered integration for Canvas. In Spring 2025, the authors were asked to aid in migrating the metadata from Ares to Leganto. The goal was to have Leganto ready for professors by Fall 2025 for the start of the semester.

Since the size of the collection was large and the timeline to complete the migration was limited, it was decided that only reading lists for courses taught over the last five years would be included in the migration to Leganto. This compromise seemed fitting, as this would allow the most current courses and their reading lists to be ready for professors to use by Fall 2025 while weeding out older reading lists that may have changed due to developments of course materials and current instructional trends. This left the team with approximately 42,000 objects that needed to be migrated.

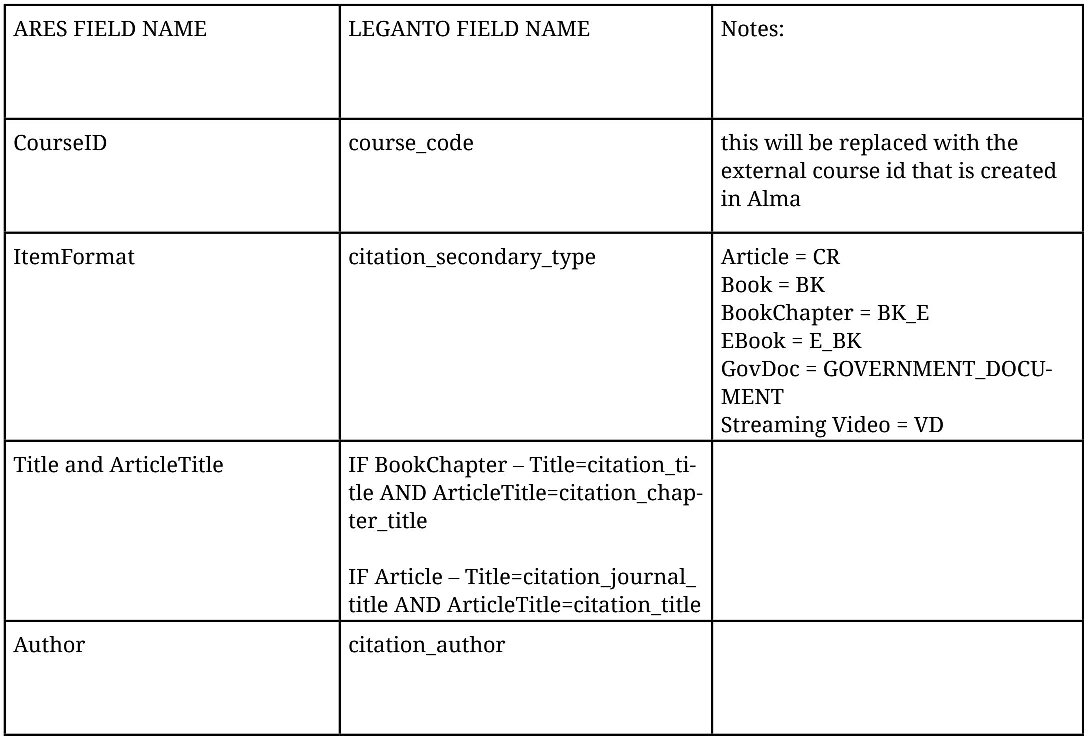
*Challen R. Wright is Metadata Librarian, University of Nevada, Reno. Shelly Johnson is Metadata Technician, University of Nevada, Reno.*

Around the time the migration began, the Libraries also became interested in ways in which we could incorporate AI into the workplace, both in daily work and in special projects. The Libraries purchased enterprise memberships to ChatGPT and Claude. These memberships offered users all features from the pro membership plans along with enhanced security features to protect institutional data. Due to the size of the Leganto migration and the quickly approaching deadline of the project, the Metadata Librarian and Metadata Technician decided to try using Claude as a way to expedite the migration process.

This case study will go through the authors' experience using AI to aid in the metadata mapping and migration process. For both authors, this was the first project where AI played a major role in its execution. The following sections will describe the processes of using Claude to create Python scripts that map the original metadata to the new schema, clean up metadata, and implement APIs to further enhance metadata with the goal of automating much of the migration process and minimizing human intervention and error.

## CLEANUP SCRIPT 1

The first goal of the project was to map the original metadata from Ares to the new metadata schema for Leganto. The authors decided to create two Python scripts for the migration with the first script mapping the metadata to the new schema and the second performing additional cleanup tasks. For the first script, a metadata crosswalk was created to match Ares fields to their Leganto field equivalent (see Figure 1). Fields in both the Ares and Leganto schemas included controlled vocabularies that had to be mapped as well (for example, Item Format used different values for object types). Additionally, some fields in Ares like Title and Article Title had to be mapped to different fields in Leganto depending on the values used in other fields.



| ARES FIELD NAME | LEGANTO FIELD NAME | Notes: |
|---|---|---|
| CourseID | course_code | this will be replaced with the external course id that is created in Alma |
| ItemFormat | citation_secondary_type | Article = CR<br>Book = BK<br>BookChapter = BK_E<br>EBook = E_BK<br>GovDoc = GOVERNMENT_DOCUMENT<br>Streaming Video = VD |
| Title and ArticleTitle | IF BookChapter – Title=citation_title AND ArticleTitle=citation_chapter_title<br>IF Article – Title=citation_journal_title AND ArticleTitle=citation_title | |
| Author | citation_author | |

**Figure 1:** Metadata Crosswalk Table from Ares to Leganto

Creating the crosswalk was the first step before using AI to develop a script. In order for Claude to create an accurate script that completed the necessary mapping functions, a very detailed prompt was required. Additionally, sample metadata spreadsheets for both schemas were created to allow Claude to analyze what each input and output file should look like when running through the Python script.

Initially, the Metadata Librarian decided to give Claude information conversationally, rather than with one single prompt. In the first chat with Claude, the Metadata Librarian started by explaining the concept of the project and slowly introducing the sample files and crosswalk table. Conversation continued until the author determined that Claude understood the details of what needed to be included in the script. The script was then created and tested on a sample of the data. The first script was unsuccessful and continued to put out error messages that stopped the script from running. After Claude made several revisions to the script that were also unsuccessful, the author decided to start a new conversation.

The second conversation yielded better results. This time, the authors provided Claude with all metadata templates and a well-defined prompt at the start, rather than supplying information periodically. The first script that Claude wrote was somewhat successful and produced an output file of the mapped metadata, along with a .txt log file that recorded all changes made in each row. There were some issues, though, such as the date field values becoming decimals. The authors identified issues like these and relayed them to Claude, which was then able to revise the Python script accordingly. Additional adjustments were made to the script during this conversation for small changes to the mapping and cleanup of the metadata. This second conversation was successful and created a proper script that did everything the authors required it to do.

### CLEANUP SCRIPT 2

The second script performed additional cleanup, such as removing extraneous prefixes from titles and standardizing the format of ISBNs. It was piloted by the Metadata Technician, who came to this project without having any experience with AI or coding other than an introductory Python class. Therefore, this was a true exercise in how Claude could help even inexperienced coders with script creation. The Metadata Technician began by uploading a sample of the metadata created from the first script and asking Claude to analyze the spreadsheet for ways to clean up the data. Claude pointed out several metadata fields with inconsistent formatting and suggested ways to standardize them. The Metadata Technician attempted to have Claude standardize the author field, but this failed due to the widely differing forms of the names. For example, Claude could not determine whether a value consisted of a single author with first and last name, or two authors' last names.

Many items were missing the ISXN (ISBN for books, or ISSN for serials) and addressing this became the most exploratory and rewarding part of creating the script. The process began by looking for missing ISXNs elsewhere in the metadata. Because several citations were repeated over different years and courses, the Metadata Technician asked Claude to extract all items with an ISBN and return it as a hardcoded dictionary that could be pasted into the script. The same was done for ISSNs. Thus, each new batch of metadata that went through the script had the ISXN added if it was found in the dictionary.

Of course, adding an ISXN that was not already elsewhere in the metadata proved more challenging. When Claude was tasked with searching for missing ISXNs, it returned impressive results for small data sets but could not handle the process for spreadsheets of several thousand items. It suggested setting up API calls to bibliographic databases, and after further querying, Open Library, Google Books, and Crossref were chosen. Fine-tuning included setting up timeouts to avoid overtaxing the APIs and producing confidence scores to accept only those matches with a high confidence level. By taking the mean of the API coverage percentage reported by Claude in the log file for each batch of data, we found that the API calls resulted in an average ISXN coverage improvement of 19.08%. The number of ISSN additions was much higher than that of ISBNs, due to the simpler matching criteria for serials. For books, in addition to title, the matching criteria included author(s) and/or editor(s), edition number, and publication date.

Several factors contributed to the success of the creation of this script. Claude created a modular structure that was easy to amend, and it added comments into each section of the script explaining its function. The log output that Claude set up, which could be seen printed in real time and in a log file, was helpful to understand the process that the script was performing and why it returned the results that it did. For each journal title missing the ISSN, for example, it showed the database to which an API call was made, which journal was found there, the title similarity score, and whether the match was accepted or rejected. Claude, as requested, also wrote a metadata cleanup summary into the log, which provided a helpful overview consisting of total records processed, records with at least one change, changes by type, and ISXN coverage before and after cleanup.

### KEY TAKEAWAYS AND CONCLUSION

While there are many different AI tools on the market, the authors were most experienced with Claude and ChatGPT. During the preliminary stages of the project, the authors found that Claude was better at analyzing data sets and creating Python scripts. While both were conversational and provided quality information, there were fewer errors in Claude's codes than those of ChatGPT.

First, the authors realized that a concise and detailed prompt, including context and desired outcomes, is essential to getting good results from AI. We noticed that using vague language or not asking clarifying questions caused Claude to make incorrect assumptions about what needed to be done and, in turn, produced errors in the output metadata. Taking the time to develop a clear prompt prior to working with an AI tool was helpful in getting a project started on the right track. After the initial prompt, when using Claude to create a multi-function script, it is important to focus on one function at a time, test the script, and refine it through querying Claude. Trying to accomplish too many functions at once makes problems harder to spot. AI tools also need constant human intervention. They are not fully autonomous and often produce results that have errors. It is extremely important to include human checking throughout each step of the project.

At the end of this project, the authors were able to successfully migrate approximately 42,000 objects from Ares to Leganto within a span of 3 months. By utilizing Claude, much of the manual work that would be required to map metadata fields, implement new controlled vocabularies, and quality check/review existing metadata was automated. The two Python scripts created were essential in reducing possible human error and expediting the migration process. Prior to generative AI, a project of this size would have taken a considerably longer amount of time and effort to complete. This project was a great starting point for implementing AI into project workflows and showed the value that a generative AI tool like Claude can bring. During this process, the authors were able to learn some important techniques when working with AI to ensure its effectiveness.