

2017

Open-Source Opens Doors: A Case Study on Extending ArchivesSpace Code at UNLV Libraries

Cyndi Shein

University of Nevada, Las Vegas Libraries, c.shein@yahoo.com

Carol Ou

University of Nevada, Las Vegas Libraries, carol.ou@unlv.edu

Karla Irwin

University of Nevada, Las Vegas, karla.irwin@unlv.edu

Carlos Lemus

University of Nevada, Las Vegas Libraries, carlos.lemus@unlv.edu

Follow this and additional works at: <http://elischolar.library.yale.edu/jcas>

 Part of the [Archival Science Commons](#)

Recommended Citation

Shein, Cyndi; Ou, Carol; Irwin, Karla; and Lemus, Carlos (2017) "Open-Source Opens Doors: A Case Study on Extending ArchivesSpace Code at UNLV Libraries," *Journal of Contemporary Archival Studies*: Vol. 4 , Article 2.

Available at: <http://elischolar.library.yale.edu/jcas/vol4/iss1/2>

This Case Study is brought to you for free and open access by EliScholar – A Digital Platform for Scholarly Publishing at Yale. It has been accepted for inclusion in *Journal of Contemporary Archival Studies* by an authorized editor of EliScholar – A Digital Platform for Scholarly Publishing at Yale. For more information, please contact elischolar@yale.edu.

OPEN-SOURCE OPENS DOORS: A CASE STUDY ON EXTENDING ARCHIVESPACE CODE AT UNLV LIBRARIES

Introduction.

Open-source software is primarily characterized by free access to its code.¹ Implementing such software often involves local customization of the code, which can then be contributed back to the community of users. Motivations for contributing to the development of open-source software range from individual incentives to corporate strategies,² and from altruism to the expectation of reciprocity.³ As of the writing of this article, over three hundred libraries and archives across the globe are paying members of ArchivesSpace, an open-source archival collection management application that is supported by three full-time employees and three registered service providers.⁴ ArchivesSpace's code is open and used by nonmember institutions; however, it is primarily member institutions that participate in the governance of the program, define development priorities, and contribute code to the application.

As a member institution, the University of Nevada, Las Vegas (UNLV), Libraries is allocating staff resources to the development of ArchivesSpace for three main reasons: (1) to move UNLV forward in the implementation of its first archival collection management system; (2) to share code and ideas that will benefit the broader community of users; and (3) to explore functions with potential to inform the development of the master codebase of the application. Dedicating the time and talents of one staff member to extend existing code or develop code that expands the current functions of ArchivesSpace has improved the workflows and productivity of staff across two departments, enabling them to make Special Collections and Archives' archival resources discoverable and accessible in a timely manner, which is central to UNLV Libraries' mission.⁵ By offering locally developed code back to the ArchivesSpace community, UNLV advances local development and also shares concepts that have the potential to move work forward on the application itself.

Unlike the majority of ArchivesSpace's early adopters, UNLV's path to implementation did not involve migrating from either of ArchivesSpace's predecessors, Archivists' Toolkit or Archon, making UNLV's fundamental needs different from the needs of those driving the development of the application. When UNLV began using ArchivesSpace in 2014, only a small percentage of UNLV's archival collection descriptions were machine-readable, and those files were neither valid Encoded Archival Description (EAD) nor DACS-compliant.⁶ At that time, ArchivesSpace

¹ For a more complete explanation, see the Open Source Initiative definition, <https://opensource.org/definition>.

² Josh Lerner and Jean Tirole, "The Open Source Movement: Key Research Questions," *European Economic Review* 45 (2001): 821.

³ Michael Heron, Vicki L. Hanson, and Ian Ricketts, "Open Source and Accessibility: Advantages and Limitations," *Journal of Interaction Science* 1, no. 2 (2013): 2.

⁴ For more information on ArchivesSpace membership, governance, and service providers, see ArchivesSpace Mission and History at <http://archivesspace.org/about/mission-and-history/>.

⁵ "In support of the University's mission and shared values, the Libraries contribute to and support learners as they discover, access, and use information effectively for academic success, research, and life-long learning." UNLV Libraries Mission Statement, https://www.library.unlv.edu/about/mission_statement.

⁶ To be "DACS-compliant," archival description must include the mandatory elements prescribed by *Describing*

developers and the majority of its early adopters were concentrating on transforming and migrating EAD files from Archon and Archivists' Toolkit. Meanwhile, UNLV was focused on how to normalize its idiosyncratic legacy data for import into ArchivesSpace and how to support local staff in creating new standardized descriptions directly in the application. While the developers of the master codebase rightly concentrate their attention on the issues ranked most essential by the community as a whole, meeting an immediate local need is best accomplished by enhancing a local instance of the repository.⁷ UNLV implemented a locally hosted instance of ArchivesSpace that can be modified to address its own requirements. Adding locally developed plugins to the local instance, rather than revising the codebase itself, offers distinct advantages:⁸

- Modifying the codebase of a local instance inevitably has negative ramifications when moving to new releases, but upgrades to new releases are generally not impaired by plugins (although plugins may need to be revised to accommodate new releases);
- Plugins can easily be shared by their authors and adopted by others in the community;
- Functions/features that gain traction through the community's use of a certain plugin become candidates for addition to the master codebase; and
- A plugin can easily be deprecated if/when the plugin's functions have been replicated or superseded in a new release of the master codebase.

Literature review.

The initiation of open-source projects and the implementation of open-source systems are not new to libraries. In 1999, Daniel Chudnov discussed then-current examples of open-source efforts in libraries and advocated for libraries to use and participate in the development of open-source systems. He noted that “open source software depends on community effort—a striking similarity to the economics of libraries.”⁹ In 2003, authors from the Massachusetts Institute of Technology (MIT) Libraries and Hewlett-Packard Labs discussed their collaboration to develop DSpace, an open-source digital repository for libraries. In developing DSpace, one of their goals was to build a system that “would be immediately useful at MIT, and hopefully at other institutions.”¹⁰ A 2008 discussion described one of ArchivesSpace's predecessors, Archon, developed by the University of Illinois, as an “open-source collections management software program [intended] to meet the descriptive and access needs of small academic and institutional archives and special collections libraries,” specifically helping them adhere to standards while creating a searchable public interface for their collections. In this conference, authors from the University of Illinois expressed their hopes that “the international user community will grow and assist us in the development” of Archon.¹¹ The value of user communities in the support and development of open-source systems is a common theme in the literature.

Archives: A Content Standard. For more information, see the Society of American Archivists' website, <http://www2.archivists.org/groups/technical-subcommittee-on-describing-archives-a-content-standard-dacs/dacs>.

⁷ Here, “master codebase” refers to the master ArchivesSpace repository and core code maintained by LYRASIS. LYRASIS serves as the organizational home for ArchivesSpace.

⁸ A plugin (or plug-in) is “a software component that adds a specific feature to an existing computer program.” Wikipedia, s.v. “Plug-in.”

⁹ Daniel Chudnov, “Open-Source Software: The Future of Library Systems?” *Library Journal* 124, no. 13 (1999): 41.

¹⁰ MacKenzie Smith et al., “DSpace: An Open-Source Dynamic Digital Repository,” *D-Lib Magazine* 9, no. 1 (2003), <http://dlib.org/dlib/january03/smith/01smith.html>.

¹¹ Scott W. Schwartz, Chris Prom, Kyle Fox, and Paul Sorensen, “Archon: Facilitating Global Access to Collections

The literature also includes recent discussions of other ArchivesSpace implementations. Arizona State University Libraries were charter members of ArchivesSpace, and Elizabeth Dunham outlines Arizona State's experiences migrating its data to the new system, pointing out how available local technical expertise assisted in implementing and maintaining the software. She also noted a local inability to customize ArchivesSpace via plugins since the organization lacked staff with the necessary skillset.¹² The ArchivesSpace implementation at West Carolina University's Hunter Library was accomplished through a collaborative workflow among multiple library departments, necessitated in part because the library did not have the technical resources to facilitate the wholesale import of existing finding aids. As described by Paromita Biswas and Elizabeth Skene, their lack of technical infrastructure also led to utilizing a hosted instance of ArchivesSpace contracted with LYRASIS. "Under this arrangement, LYRASIS provides server support, technical assistance, and system upgrades for ArchivesSpace," as well as some limited customization. With regards to the ArchivesSpace user community, the authors list a challenge related to a "seeming absence of peer institutions with whom to compare workflows and learn," since Hunter Library was neither migrating from another archival collection management system nor capable of hosting and customizing the software itself.¹³ Mackenzie Brooks and Alston Cobourn describe an ArchivesSpace implementation at Washington and Lee University, one that occurred seemingly early. While the system has bugs, "the application continues to improve and will only get better as more people contribute." They laud the experience of collaborating with other departments and libraries as gratifying. In addition, they specifically highlight the plugin architecture of ArchivesSpace, which "means that various features can be developed, shared, and implemented to create an application right for each institution."¹⁴

Staff at Harvard University and the Bentley Historical Library at the University of Michigan likewise discussed their ArchivesSpace experiences with specific descriptions of what can be achieved when programming resources are available. As Dave Mayo and Kate Bowers note, the migration of EAD to ArchivesSpace at Harvard led to the development of several locally used tools as well as other contributions to the community. They reported a number of issues related to the importer and also contributed code to ArchivesSpace via GitHub pull requests, including code that was originally part of their Custom Importer Plugin.¹⁵ Max Eckard, Dallas Pillen, and Mike Shallcross describe a grant-funded project to integrate several open-source systems, including ArchivesSpace, DSpace, and Archivematica, an open-source digital preservation system. For this project, staff from the Bentley Historical Library and the University of Michigan Library worked with Artefactual Systems (the developer of Archivematica) to outline development that would be

in Small Archives" (presentation, World Library and Information Congress: 74th IFLA General Conference and Council, Québec, Canada, August 10-14, 2008), https://archive.ifla.org/IV/ifla74/papers/159-Schwartz_Prom_Fox_Sorensen-en.pdf

¹² Elizabeth Dunham, "Implementing ArchivesSpace at Arizona State University," *Journal of Digital Media Management* 4, no. 3 (2016): 280–92.

¹³ Paromita Biswas and Elizabeth Skene, "From Silos to (Archives)Space: Moving Legacy Finding Aids Online as a Multi-Department Library Collaboration," *The Reading Room: A Journal of Special Collections* 1, no. 2 (2016): 72, 78–79.

¹⁴ Mackenzie Brooks and Alston Cobourn, "ArchivesSpace at W&L: Why We Didn't Wait," *Mid-Atlantic Archivist* 43, no. 4 (2014): 4–5.

¹⁵ Dave Mayo and Kate Bowers, "The Devil's Shoehorn: A Case Study of EAD to ArchivesSpace Migration at a Large University," *Code4Lib Journal* 35 (2017), <http://journal.code4lib.org/articles/12239>.

needed to support this integration. Code completed by Artefactual Systems for this joint project will be included in Archivematica 1.6.¹⁶

Archivists' Toolkit is a widely adopted and robust open-source archival collection management system that preceded ArchivesSpace; its development offers some lessons regarding the importance of a user community that is enabled and empowered to participate. Sibyl Schaefer discusses specific challenges related to making the Archivists' Toolkit open-source project sustainable past initial grant funding, arguing that "governance of the project needed to be more open, delegating tasks to users whenever possible in order to minimize overhead costs and essentially becoming a true collaborative and community-based open-source venture." Schaefer then outlines several missed opportunities where the project did not fully open up development or successfully incorporate user volunteers for product testing and other tasks. It was also not until near the end of Archivists' Toolkit's development that the project added a plugin framework, thereby providing a mechanism to provide "basic means for code contribution without forking the code."¹⁷

Themes emerging from the literature highlight the advantages of having in-house technical expertise to support implementation of open-source systems and confirm the essential role of user communities in supporting and developing these systems.

Background.

The UNLV Libraries is a center for scholarship and lifelong learning for the diverse and dynamic southern Nevada community. The Libraries includes one main library and three branches, and employs more than 120 faculty and staff. The Special Collections and Archives Division stewards and provides public access to more than thirteen thousand linear feet of archives, manuscripts, and photographs; over thirty thousand rare books, maps, government documents, and serials; over three thousand oral histories; and over seventy thousand online, digitized items. Special Collections and Archives' mission focuses on supporting the interdisciplinary study of Las Vegas, southern Nevada, and gaming.¹⁸ In support of that mission, the Discovery Services Department (Collections, Acquisitions and Discovery Division) and the Special Collections and Archives Technical Services Department (Special Collections and Archives Division) work together to foster discovery and access, and to safeguard collections for future generations.

In 2013, the UNLV Libraries formally recognized its critical need for an archival collection management system. Thousands of accession records, source files, finding aids, and inventories describing its archival collections had been created over time in a variety of formats and were dispersed across different print and electronic environments. Improving staff and public access to this information required that the records be normalized, centralized, and enhanced. When

¹⁶ Max Eckard, Dallas Pillen, and Mike Shallcross, "Bridging Technologies to Efficiently Arrange and Describe Digital Archives: The Bentley Historical Library's ArchivesSpace-Archivematica-DSpace Workflow Integration Project," *Code4Lib Journal* 35 (2017), <http://journal.code4lib.org/articles/12105>.

¹⁷ Sibyl Schaefer, "Challenges in Sustainable Open-Source: A Case Study," *Code4Lib Journal* 9 (2010), <http://journal.code4lib.org/articles/2493>.

¹⁸ For more detail, see the UNLV University Libraries Special Collections and Archives Mission webpage, <https://www.library.unlv.edu/speccol/about/mission>.

considering the options, decision-makers cited their positive experiences with Archivists' Toolkit at previous institutions and noted that commercial software was cost-prohibitive. Archivists' Toolkit and Archon were widely adopted but no longer grant-supported, and a number of respected peer institutions had committed to moving that work forward by becoming charter members of ArchivesSpace.¹⁹ This indicated that the profession was moving in the direction of community-based applications, and UNLV wanted to join that active and innovative community. Although ArchivesSpace was known to be underdeveloped, UNLV viewed it as the most promising option for the foreseeable future. UNLV Libraries became a paying member of the ArchivesSpace community and began implementation in 2014; as of this writing, UNLV is using version 1.5.4.

UNLV Libraries has a Library Technologies Division; to date its role in ArchivesSpace implementation has been for the Systems Department staff to install test and production instances of the application on a local server, add files (plugins) upon request, re-index upon request, and upgrade to new releases. All other responsibilities are left to librarians. The first year of implementation focused on populating ArchivesSpace: a librarian standardized and imported legacy EAD files into ArchivesSpace, and inexperienced paraprofessional and student interns began manually entering other legacy information, bringing descriptions up to minimal DACS standards as they went. Throughout this first year, staff noted specific shortcomings in ArchivesSpace and envisioned functions that would create efficiencies during implementation. Since Library Technologies' application developers were overextended and lacked familiarity with Ruby (the object-oriented programming language on which ArchivesSpace is built), other means of support for local application enhancements were sought.

Defining and meeting local needs.

As UNLV began using ArchivesSpace, staff soon came up with a wish list of functions to support local implementation. Priorities identified early in the implementation process included

1. Transforming legacy data for import into the application to ensure that all archival collections are represented in ArchivesSpace,
2. Creating efficiencies for repurposing metadata across departments and systems,
3. Cleaning up name and subject headings prior to launching the public user interface, and
4. Making the display of PDFs of finding aids/resource records easier for researchers to interpret and understand.

Since priorities two and three involved shared interests between Technical Services and Discovery Services, the heads of those departments collaborated to propose the hire of a temporary application programmer in support of an exploratory, cross-departmental project. Internal funding was obtained to support a part-time, eleven-month position; due to ongoing need and the progress demonstrated during the first eleven months, the position was renewed for a second term. Recruiting for the position focused on students from UNLV's College of Engineering, which

¹⁹ Official development of Archivists' Toolkit and Archon ceased September 30, 2009; the original developers stopped providing user support and bug fixing for these applications in September 2013. For more information, see <http://archivespace.org/about/mission-and-history/>. As of the writing of this article, seven institutions have collaboratively funded an update of Archon and formed a user group that is described here: <https://sites.google.com/denison.edu/archonupdateproject/about>.

resulted in hiring a skilled and self-directed undergraduate student to investigate the capabilities of ArchivesSpace and come up with ways to meet the needs articulated by staff. Collaboration between the librarians and the programmer led to the development of plugins that enable the following efficiencies:

- Creating resource records (collection descriptions),
- Cleaning up messy metadata in the Agent and Subject modules,
- Repurposing exported metadata for other systems, and
- Displaying exported collection descriptions in a way that is more meaningful to researchers.

Efficiently spawning resources from accession records.

The top priority of the UNLV ArchivesSpace implementation team was (and still is) to import or create a record for each archival collection, so that all collections are represented in ArchivesSpace and all collection description is centralized. While paraprofessional staff and students continue to manually create ArchivesSpace resource records for manuscript collections that have no machine-readable records, a librarian is working to clean up and import descriptions of over three thousand oral history interviews using legacy data from a homegrown database. The challenge in creating finding aids for the interviews is that their item-level descriptions are minimal, not DACS-compliant, and structurally do not parallel EAD. CSV (Comma Separated Values) files exported from the homegrown database can only be imported into ArchivesSpace's Accession module. Resource records can only be imported as EAD files. UNLV will be providing public access to collections through resource records but not through accession records, which are created and used for internal administrative purposes only. Given the inconsistencies in the data, converting the interview descriptions from CSV into EAD prior to import proved too labor-intensive. Since there was no clear way to bulk import the legacy data into the Resource module, UNLV imported the oral history interviews as individual accessions and investigated ways to efficiently generate resources from the accessions.

By default, ArchivesSpace has a "spawn" feature that generates a resource record from information found in an accession record. Unfortunately, resource records must be spawned one at a time, which is impractical when faced with spawning thousands of records. Exploration of the built-in spawn function revealed two additional shortcomings: not all essential fields transfer over into the spawned resource record, and it is not possible to apply the "pre-populate" function to any of the fields. In order to create resource records for its oral history interviews, UNLV needed to spawn resource records from accession records more efficiently by creating multiple records simultaneously, transferring all public fields from the accession record to the resource record during spawning, and auto-populating fields that contain boilerplate values.

To address this need, the application programmer created the UNLV Spawn Plugin, which allows staff to search accessions by keyword, select multiple accession records, and then spawn multiple resources from all the selected accessions simultaneously (see appendix figs. 1 and 2). Once spawned, each resource record must be manually edited and saved individually, but the plugin eliminates the step to create resource records one by one from each accession record. The biggest time savings gained by this plugin is the ability to auto-populate additional necessary fields. When

an accession record is spawned, ArchivesSpace copies the values in the Title, Dates, Extent, Agent, and Scope and Contents fields from the accession into the spawned resource. The UNLV Spawn Plugin enhances this function—it automatically transfers values from additional fields, copying them from the accession record to the spawned resource. The plugin also auto-populates boilerplate notes that are not in the accession record but are required in a resource record per DACS (e.g., Conditions Governing Access and Conditions Governing Use notes) based on local standardized text. To complete the resource record, the plugin also automatically adds a local Classification for oral histories and the Art and Architecture Thesaurus’s subject “oral histories (document genres)” to each spawned resource record.

The UNLV Spawn Plugin expedites the local implementation of ArchivesSpace by establishing a smoother workflow for creating thousands of oral history resource records. It also allows UNLV to maintain the item-level discoverability of these frequently requested materials as UNLV transitions from the homegrown database to ArchivesSpace. The local modifications, tailored to spawn oral history records, can be edited or disabled to help staff efficiently create resource records for all types of archival collections (manuscripts, photographs, etc.) that have accession records in ArchivesSpace. Settings can easily be edited within the staff interface as needed. The subject, local classification, and access and use notes can all be customized to accommodate the needs of each set of records that are being spawned (see appendix fig. 3).

Transforming MARCXML export for use in other systems.

While the UNLV Spawn Plugin focuses on efficiently creating collection records within ArchivesSpace, the MARCXML Exporter Plugin focuses on customizing exported data to facilitate creating collection records for other systems—OCLC WorldCat and the UNLV Libraries’ online catalog. UNLV Libraries currently describes archival collections using two encoding standards: EAD for finding aids and MARC for bibliographic records. Finding aids are generated from ArchivesSpace and published online as PDFs. MARC records are created as original cataloging records in OCLC WorldCat using the Connexion client, then downloaded to the Libraries’ local catalog. The finding aids are created by the Technical Services Department (Special Collections and Archives Division), and the cataloging is done by the special collections cataloger in the Discovery Services Department (Collections, Acquisitions and Discovery Division).

The current workflow for MARC cataloging of archival collections begins when the finding aid is completed, published as a PDF, and forwarded from Technical Services to the special collections cataloger. The cataloger then creates the MARC catalog record in OCLC Connexion using descriptive information from the finding aid combined with additional metadata required by the MARC standard and the UNLV Libraries’ local cataloging policies. She refers to the Library of Congress authority file as well as the catalog’s local authority file to confirm or create name and subject headings, and then adds them to the MARC record. Prior to fall 2015, the inclusion of descriptive metadata from the finding aid in the MARC record was largely a manual copy-and-paste process. In 2015, however, the Discovery Services Department began to experiment with importing the default MARCXML exports from ArchivesSpace directly into OCLC Connexion. Although the raw imported MARCXML record did not initially meet MARC or the Libraries’ local cataloging standards, the department was able to develop Connexion macros to handle many

common edits, such as reformatting fields, inserting standard values, and deleting additional descriptive information that would not normally be included in the cataloged MARC record. This new process of importing the MARCXML record and employing Connexion macros for standard edits replaced the former, tedious process of cutting and pasting from the PDF finding aid, and allowed the special collections cataloger to focus instead on the more complex authority work, subject cataloging, and other proofreading required for each record. As of fall 2015, this procedure had been fully adopted for all original cataloging of archival collections.²⁰

Although repurposing the default ArchivesSpace MARCXML export worked well, staff quickly identified and began to explore additional improvements with the potential to streamline the new procedure. Two improvements promising the greatest efficiencies were (1) customizing the ArchivesSpace MARCXML export so fewer edits would need to be made to the record in Connexion, and (2) exporting multiple MARCXML records as a single file to decrease the number of clicks and keystrokes required to export each archival collection from ArchivesSpace and import it into Connexion.

Toward these improvements, the application programmer developed a plugin for ArchivesSpace that allows staff to customize the MARCXML export via the ArchivesSpace staff interface. The plugin allows staff to toggle the export of specific MARCXML fields. It also permits certain locally standard batch edits such as replacing the period in the collection identifier with a dash and customizing the finding aid note in the MARC 555 field (see appendix fig. 4). The UNLV MARCXML Exporter Plugin was implemented in the Libraries' production instance of ArchivesSpace in December 2016, and the customized MARCXML output now allows the special collections cataloger to use a smaller and faster set of Connexion macros. Thanks to ArchivesSpace REST (representational state transfer) APIs (Application Programming Interface), certain functionalities can also be facilitated or repurposed using Python (a programming language) outside of the ArchivesSpace directory. The application programmer wrote a Python script (Multi Marc Exporter) to batch export MARCXML records from ArchivesSpace as a single file. This script is currently being tested and will soon be adopted for production use.

Cleaning up agent and subject records.

While the special collections cataloger leverages her professional expertise and years of experience to create authorized names and subjects in the MARC records that describe archival collections, no staff members earlier in the description workflow have the training or experience needed to assign or establish authorized headings in the finding aids they create. Adding to the chaos of names and subjects that have been manually created in UNLV's local instance of ArchivesSpace, the legacy EAD files imported into ArchivesSpace during initial implementation were not consistently subject to authority control and still need cleanup. Furthermore, during import, names that were embedded in EAD records imported into a single data field in ArchivesSpace and subjects imported as a single string, with subfields separated by hyphens but no indication as to the nature (topical, temporal, geographic, etc.) of each subfield.²¹ Due to the limited number of

²⁰ Carol Ou, Katherine L. Rankin, and Cyndi Shein, "Repurposing ArchivesSpace Metadata for Original MARC Cataloging," *Journal of Library Metadata* 17, no. 1 (2017): 19–36.

²¹ The authors suspect unparsed names and subjects imported into ArchivesSpace to be a fairly common problem in the archives community. Although EAD accommodates subfields associated with names and subjects, previous tools,

names visible in the built-in dropdown to create agent records in the Resource module, it is not always apparent that a name already exists; until staff identified this flaw, an unknown number of duplicate names were mistakenly created. Duplicate records for some names and subjects were also automatically created during ingest of accessions (CSV) and resources (EAD). UNLV needs to not only clean up its Agent and Subject modules in ArchivesSpace but also to establish procedures that support inexperienced staff in creating names and subjects going forward.

As of the writing of this article, of the 4,715 names in UNLV's instance of ArchivesSpace, over half of them are unauthorized and in need of review and revision:

- 2,712 Unspecified ingest source (need authority control)
- 1,771 Local source (have been researched and established locally)
- 232 NACO Authority File

Similarly, of the 1,128 subject headings, well over half of them need review and revision:

- 660 Unspecified ingested source (need authority control)
- 50 Local source (have been researched and established locally)
- 361 Library of Congress Subject Headings
- 57 Art & Architecture Thesaurus

To assist with de-duplication, cleanup, and improvement of name and subject creation workflows, UNLV adopted and/or created three plugins: a UNLV Custom Reports Plugin, an LC Authority Import Plugin, and an Overlay Plugin.

The UNLV Custom Reports Plugin facilitates export of reports (JSON, CSV, XLSX, or PDF) sorted alphabetically by agent name or sorted alpha-numerically by Authority ID. UNLV is using this plugin to export data to an Excel spreadsheet and custom-sort several columns to identify duplicate names, anomalies in names, and names without authority control (Source = ingest). The report helps target names for cleanup.

UNLV adopted and adapted an existing LCNAF Plugin, shared through the open-source community, to help inexperienced staff create authorized names and subjects.²² The community plugin opens within ArchivesSpace in a user-friendly interface through which staff are able to search Library of Congress headings directly, select appropriate headings, and import headings via an API call to the Library of Congress Linked Data Service. At the time UNLV implemented this community plugin, it utilized the default MARC importer of ArchivesSpace, which did not include the essential Authority ID field. The UNLV programmer created a custom MARC importer that includes the Authority ID and then extended the community's LCNAF plugin to work with UNLV's custom MARC importer, calling this local plugin the LC Authority Import Plugin.

such as Archivists' Toolkit, had only one data field in which to enter names, and no data fields to enter subfields for subjects.

²² UNLV's application programmer adapted an existing LCNAF plugin found on the ArchivesSpace GitHub profile at <https://github.com/archivesspace/archivesspace/tree/master/plugins/lcnaf>.

Thus far, UNLV's Overlay Plugin has proven the most useful tool for cleaning up local names and subjects. This plugin was developed locally to de-duplicate name records without losing content or relationships between records. In version 1.5.4 of ArchivesSpace, the built-in "merge" function does not actually merge the records. Although it merges the resource and accession records associated with an agent or subject, it does not merge the data in the selected agent or subject records—it destroys the "victim" record and overwrites it entirely with the "target" record. When creating agent records, UNLV staff include handcrafted biographical notes, agent relations, and other unique, locally created descriptions, resulting in values that need to be retained in certain fields. The Overlay Plugin permits staff to import an authorized agent record from the Library of Congress, and, if it duplicates a handcrafted record, to overlay only the existing unauthorized agent heading and Authority ID in the handcrafted record, protecting all other existing fields. The Overlay Plugin can also be used to genuinely merge agents or subjects that are already in ArchivesSpace.

Even with the efficiencies created by the plugins, cleanup of name and subject headings is a major undertaking because many of the steps require human judgment and experience in authority control. This is a critical step that must be completed before UNLV Libraries can consider using the public user interface of ArchivesSpace.

Customizing the PDF finding aid.

Since its implementation of ArchivesSpace in 2014, UNLV Libraries has been using the Resource module to create and edit collection descriptions, but as of the writing of this article, UNLV has not implemented the public user interface. Collection-level descriptions are made publicly available through the library catalog and through the aforementioned homegrown database, both of which provide an actionable link to a PDF of a finding aid that contains the fuller collection description and inventory. Through interactions with patrons, UNLV staff identified several areas of the PDFs generated by ArchivesSpace that were confusing to users, affecting their basic understanding of the collection contents. Modifying the display involved manipulating the EAD as well as the XSLT stylesheet that transforms the EAD to PDF.

When ArchivesSpace generates a PDF finding aid, the content of each PDF is drawn from an ArchivesSpace EAD export. Prior to UNLV's hiring of the student application programmer, preparing finding aids for public display required several steps. The librarian created a local stylesheet, which included UNLV branding and edits designed to present the information more clearly to users.²³ As each finding aid was completed, the librarian edited the XML in each ArchivesSpace-generated EAD file and then manually converted the EAD to a PDF by applying the local stylesheet (using oXygen's transformation function). Having one librarian performing manual EAD-to-PDF conversions for each finding aid was labor-intensive; UNLV needed a way to divide the labor and streamline the process.

After the application programmer was added to the team, he developed a plugin to modify the EAD export, adding publisher and copyright information, rendering human-readable enumerations, and adding human-readable relator translations. He also made custom changes to

²³ UNLV's initial PDF modifications were based on the Getty Research Institute's changes to the ArchivesSpace XSLT stylesheet. Additional local changes to the XSLT have since been made by UNLV's programmer.

the XSLT to alter the way information from the EAD is displayed. The UNLV EAD Export Plugin and the modified stylesheet work in conjunction in the backend of ArchivesSpace to automate the production of customized PDFs.

The programmer built on the librarian's initial stylesheets and EAD modifications; their changes combine to make the information in the PDF easier for researchers to interpret:

- Title page: display finding aid title rather than filing title (filing titles are inverted for indexing and not user-friendly), replace ArchivesSpace credit with finding aid authors' names, add branding (UNLV logo), add copyright symbol and statement
- Front matter: spell out label abbreviations in creator field (e.g., display "contributor" rather than "ctr") so labels are easier to understand, add parentheses around the text in the container summary field so information is conveyed more clearly
- Inventory: adjust font styles for series/subseries titles and notes to reflect hierarchical description, increase the width of the container table to reduce unnecessary blank space and accommodate more text per line, adjust margins to visually represent relationships between nested components and their associated notes, reduce unnecessary line breaks between labels and the notes the labels reference, change the label over the box numbers from "Instances" to "Containers"

The EAD Exporter Plugin, combined with the customized XSLT stylesheet, makes collection descriptions easier for users to understand and automates the production of ready-to-publish PDFs directly from ArchivesSpace's Background Jobs function. This not only streamlines production; it also empowers all staff to generate PDFs, freeing the librarian for other duties.

Lessons learned.

During its implementation of ArchivesSpace, UNLV learned several lessons:

- Staff experienced firsthand the degree of improvement a programmer can bring to both workflows and work product;
- Staff was somewhat surprised to learn that programming was not the only labor gap it needs to fill along the path to full implementation; and
- Staff confirmed that student employees are capable of immense contributions to the process.

As demonstrated throughout this article, the student application programmer added much-needed skills to the ArchivesSpace implementation team. Even with the addition of this capable programmer, however, the pace of full implementation of ArchivesSpace has been slower than anticipated. The programmer often develops solutions faster than library staff can test and implement them. Implementation of the programmer's code revealed an unforeseen staffing gap in some areas—for example, the UNLV Libraries has installed plugins in the production instance of ArchivesSpace that will help perform efficient resource spawning and legacy metadata cleanup, but it lacks a workforce to undertake these large projects. Furthermore, while the LC Authority Import Plugin greatly assists inexperienced staff in populating ArchivesSpace with authorized LC names, expertise is still needed to perform authority control on the majority of names in UNLV's

Agent module, notably names that are specific to the region and not found in LCNAF. UNLV will need to fill this gap before implementing the public user interface for ArchivesSpace.

Finally, one very clear lesson learned was, when budgets do not permit hiring permanent professional staff, employing bright students can be a very effective way to add skills to the existing workforce. Over the course of twenty-two months, the student worked over 1,600 hours at UNLV Libraries and completed eleven ArchivesSpace-related projects that were put into production. Standing biweekly group meetings and centralized working documents served as the primary means of managing the projects. Once objectives and priorities for each project were clear, the student was largely self-directed, requesting one-on-one meetings with staff only as needed. He documented development decisions by commenting directly in the code and shared the code through GitHub. He also wrote internal reports in laymen's terms to clearly explain the tools for a less technical audience. Some of the characteristics and abilities that made this student successful in developing solutions for ArchivesSpace were solid programming skills; exceptional listening and communication skills; superior problem-solving skills; self-direction; a strong work ethic; good organizational skills; a willingness and ability to learn at the point of need; and an interest in learning library and archives concepts, workflows, and schemas that served to inform his programming work. Offering compensation at the level of a contract employee (rather than student wages) helped attract quality candidates. Hiring an undergraduate from a computer-related discipline added much-needed skills to the ArchivesSpace implementation effort at UNLV Libraries.

Conclusion.

Implementing an emerging, open-source archival collection management system places technical demands on local staff, but it also places control of the application's local development in the hands of those staff, enabling them to modify the application to suit local pace and priorities. Finding that ArchivesSpace's gradual development was not serving immediate local needs, UNLV Libraries built on communal code and authored original local code to facilitate its implementation and use of the application. While there are still several steps before UNLV Libraries accomplishes full implementation of ArchivesSpace, significant local advances have been made by using plugins to

- Enhance the spawn function to efficiently transform accession records into resource records and automatically add mandatory fields;
- Customize the MARCXML export to efficiently create standardized bibliographic records for publication in other platforms;
- Customize reports to identify unauthorized names, adapt a communal LCNAF Plugin to facilitate creation of authorized names and subjects, and develop an Overlay Plugin to de-duplicate agent and subject records without losing content or relationships; and
- Modify the EAD export and associated XSLT stylesheet to automate production of ready-to-publish finding aid PDFs and present researchers with more comprehensible collection descriptions.

For UNLV Libraries, participating in ArchivesSpace's open-source community has opened doors for staff to address issues collaboratively, improve workflows, and make customized collection

descriptions available to users in a timely manner. ArchivesSpace implementation served as a catalyst for valuable collaboration across departments and created opportunities to exchange ideas with the larger community of professionals. During its adoption of ArchivesSpace, UNLV has experienced the power of community-based software through the generosity of fellow professionals who have taken time to answer technical questions via the online Member Forum, openly shared their code through GitHub, and provided encouragement and technical support for the local implementation team. Inspired by this esprit de corps, in addition to sharing locally developed code through GitHub, UNLV and UNLV's application programmer have executed corporate and individual agreements permitting the programmer to collaborate with ArchivesSpace developers and contribute code to the master codebase.²⁴ Although choosing to implement an open-source solution can be a complex decision, choosing to give back to the open-source community should be a simple one.

²⁴ All locally developed ArchivesSpace code is shared through UNLV's GitHub account at <https://github.com/UNLV-Libraries/ArchivesSpace-authority-project>.

Appendix

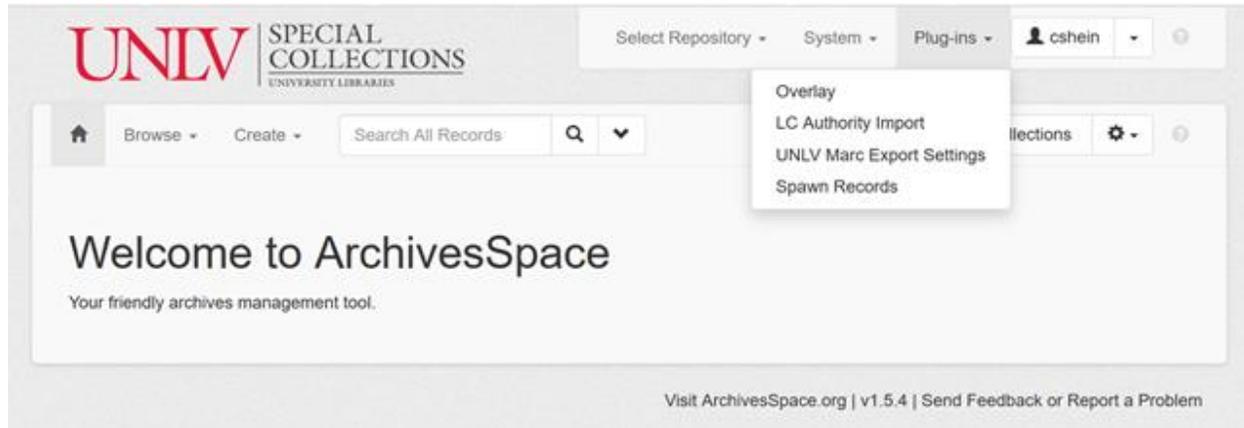


Figure 1. Plugins described in this article are accessed from the top navigation bar of the staff interface of UNLV’s local instance of ArchivesSpace.

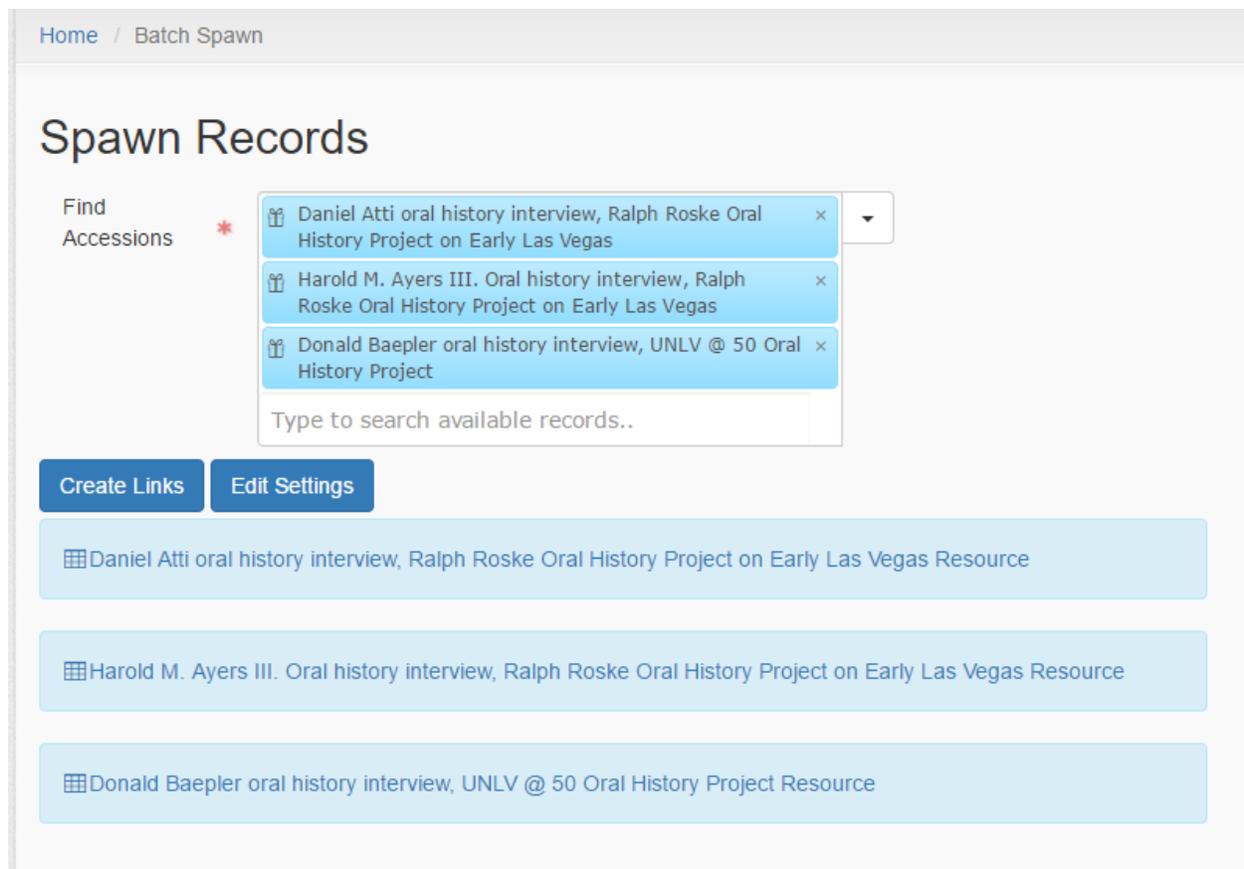


Figure 2. UNLV Spawn Plugin search box showing how oral history accession records are selected for bulk creation of linked resource records

User Spawn Settings Defaults -- UNLV Special Collections

Edit these values to set user Spawn settings defaults for the current repository. ✕

Some tags, if enabled, require the text field to be filled out for the setting to be applied ✕

Enable Custom Settings

Tag name	Enable
Access Restrict	<input checked="" type="checkbox"/>
User Restrict	<input checked="" type="checkbox"/>
Subject Link	<input checked="" type="checkbox"/>
Classifications Link	<input checked="" type="checkbox"/>
EAD ID tag enable	<input checked="" type="checkbox"/>

Custom Settings

Access Restrict Note	Collection is open for research. If use copies of the recordings do not exist, reformatting/prc
User Restrict Note	Materials in this collection may be protected by copyrights and other rights. See <extref xlr
Subject Link	/subjects/1104
Classifications Link	/classifications/2
EAD ID tag *	US::NvLN::

Figure 3. UNLV Spawn Plugin settings

Save

User Marc Export Settings Defaults -- UNLV Special Collections

Edit these values to set user marc export settings defaults for the current repository. These values can be overridden by a user's own marc export settings. ✕

Hover over tags for a description of its mapping to ArchivesSpace and its description in <http://www.loc.gov/marc/bibliographic/>. ✕

Enable Tags Settings

Tag name	Enable
Tag 008	<input type="checkbox"/>
Tag 041	<input type="checkbox"/>
Tag 099	<input checked="" type="checkbox"/>
Tag 245 subfield code f	<input type="checkbox"/>
Tag 245 subfield code g	<input type="checkbox"/>
Tag 351	<input type="checkbox"/>
Tag 500	<input type="checkbox"/>
Tag 506	<input checked="" type="checkbox"/>
Tag 520 with	<input type="checkbox"/>
Tag 541	<input type="checkbox"/>
Tag 555	<input checked="" type="checkbox"/>
Tag 852	<input type="checkbox"/>
Tag 856	<input checked="" type="checkbox"/>

Special Settings

- Strip hard returns
- Replace period with dash in identifier
- Unite qualifier with name for Tag 610
- Search for url in Tag 506 and create additional Tag 856
- Add agent notes

Change label for created Tag 856

Change label for finding aid

Figure 4. UNLV MARCXML Exporter Plugin settings