



ArchivesSpace

a community served by ✦ LYRASIS

ArchivesSpace Plug-ins

Laney McGlohon

(from a workshop originally developed with
Lora Woodford)



ArchivesSpace

a community served by ✦ LYRASIS



Agenda

- What is an ArchivesSpace plug-in?
 - Overview
 - When - and when not - to use one
 - Examples
 - Finding plug-ins
- What if I can't use plug-ins?
- Getting help with developing plug-ins and finding a wider audience





What is an ArchivesSpace plug-in?

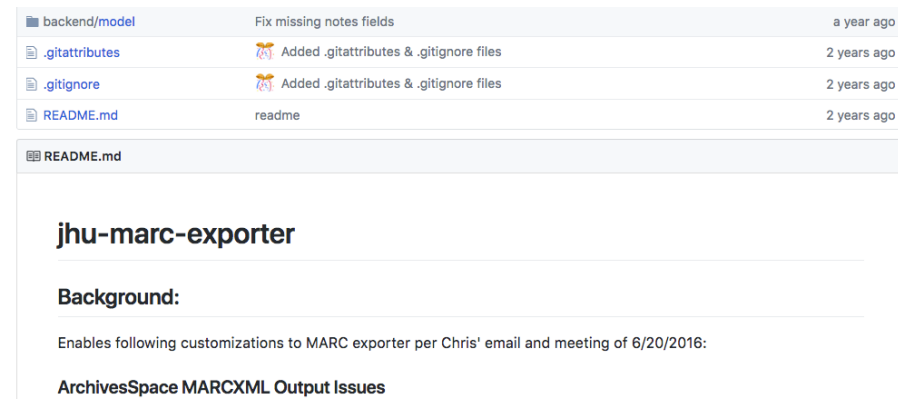
ArchivesSpace plug-ins provide a mechanism to customize ArchivesSpace by overriding or extending functions without changing the core codebase. As they are self-contained, they also permit the ready sharing of packages of customization between ArchivesSpace instances.





When should you use a plug-in?

- To add **institution-specific** extended functionality, style, branding, and customizations
- Examples:
 - Customizing exporters
 - Setting required fields
 - Auto-generating DOIDs, accession numbers, etc.
 - Modifying labels
 - Customizing staff or public display



When shouldn't you use a plug-in (in a perfect world)

- Obvious bugs
 - Bugs should be reported to the program and fixes incorporated into the core code
 - You may still wish to expedite the fix by implementing a plug-in in the interim, and may even wish to submit a pull request back to the core code!
- One-off data needs
- When you can't commit to a maintenance plan





Maintenance considerations

- New releases may impact the efficacy (or even the need for) your plug-in
- Pay attention to release notes and always confirm your plug-ins are still working as intended before upgrading
- Example:
 - JHU MARC exporter initially included bug fix for [AR-1189](#)
 - JHU PUI plug-in (not surprisingly) broken post-2.1





Customizing field and option labels

The screenshot shows the ArchivesSpace interface. At the top left is the ArchivesSpace logo with the tagline "a community served by LYRISIS". To the right are navigation menus: "Select Repository", "System", and a user profile dropdown for "admin". The user dropdown menu is open, showing options: "My Global Preferences", "My Repository Preferences", "Become User" (highlighted with a blue box), and "Logout". Below the navigation is a search bar with "Search All Records" and a search icon. The main content area displays "Welcome to ArchivesSpace" and "Your friendly archives management tool." At the bottom right, there is a footer: "Visit ArchivesSpace.org | v2.2.2 | Send Feedback or Report a Problem".

This screenshot is identical to the one above, but the user dropdown menu is open and the "Become Another User" option is highlighted with a blue box. The rest of the interface, including the logo, navigation menus, search bar, and footer, remains the same.





Moving branding image from right to left

ArchivesSpace Public Interface

Repositories Collections Digital Material Unprocessed Material Subjects Names Record Groups Q

Welcome to ArchivesSpace
Search across our collections, digital materials, and more.

Search The Archives

Enter your search terms Search all record types Keyword Years: From To To +

Search

ArchivesSpace Public Interface

Repositories Collections Digital Material Unprocessed Material Subjects Names Record Groups Q

Welcome to ArchivesSpace
Search across our collections, digital materials, and more.

Search The Archives

Enter your search terms Search all record types Keyword Years: From To To +

Search

Staff Interface | Visit ArchivesSpace.org | v2.2.2 | Send Feedback or Report a Problem





Autogenerating Digital Object IDs

Save

Digital Object Digital Object

Basic Information

Title *

Identifier *

Publish?

VRA Core Level

Type

Language

Restrictions?

Original functionality

Must enter ID to save

Save

Digital Object Digital Object

Basic Information

Title *

Identifier -- auto-generated upon save --

Publish?

VRA Core Level

Type

Language

Restrictions?

Added functionality

No need to enter before save





Customizing the MARC exporter

```
3 <record>
4 <leader>00000npc a2200000 u 4500</leader>
5 <controlfield tag="008">180124i2018
6 xx eng d</controlfield>
7 <datafield ind1=" " ind2=" " tag="852">
8 <subfield code="a">Repository Organization
9 Agency Code</subfield>
10 <subfield code="b">test database</subfield>
11 <subfield code="c">xyz.1234.abc.890</subfield>
12 </datafield>
13 <datafield ind1=" " ind2=" " tag="040">
14 <subfield code="a">Repository Organization
15 Agency Code</subfield>
16 <subfield code="c">Repository Organization
17 Agency Code</subfield>
18 <subfield code="e">dacs</subfield>
19 </datafield>
```

Default exporter

```
3 <record>
4 <leader>00000npcaa2200000 i 4500</leader>
5 <controlfield tag="008">180124i2018
6 dcu eng d</controlfield>
7 <datafield ind1=" " ind2=" " tag="852">
8 <subfield code="a">Code4Lib 2018</subfield>
9 <subfield code="b">test database</subfield>
10 <subfield code="e">2500 Calvert St NW,
11 Washington, DC 20008</subfield>
12 <subfield code="c">xyz.1234.abc.890</subfield>
13 </datafield>
14 <datafield ind1=" " ind2=" " tag="040">
15 <subfield code="a">pse1s</subfield>
16 <subfield code="b">eng</subfield>
17 <subfield code="c">pse1s</subfield>
18 <subfield code="e">dacs</subfield>
19 </datafield>
```

Overridden exporter





Some simple plug-in examples

- Autogenerating Digital Object IDs - <https://github.com/lorawoodford/autogenerate-doid>
- Customizing MARC exports - <https://github.com/lorawoodford/custom-marc-exporter>
- Enabling imports from CSV - <https://github.com/harvard-library/aspace-import-excel>





Finding plug-ins

- If you have a need and it's not met in the application, chances are someone else has had it too - and may have created a plug-in for it
 - Look at <https://github.com/archivesspace/archivesspace/tree/master/plugins>, <https://github.com/archivesspace-labs>, and <https://github.com/archivesspace/awesome-archivesspace>
 - Search github for “archivesspace”
 - Lots of cool development going on by many community members





What do I do next?

- Download the plugin
- Put it in the plugins directory in your ArchivesSpace installation
- Enable the plugin
- Restart ArchivesSpace





Plug-in directory structure

The directory structure within a plug-in is similar to the structure of the core application. The following shows the supported plug-in structure. Files contained in these directories can be used to override or extend the behavior of the core application.

```
local ..... Enabled by default
backend ..... Database and API
  controllers ..... backend endpoints
  model ..... database mapping models
  converters ..... classes for importing data
  job_runners ..... classes for defining background jobs
  plugin_init.rb ..... if present, loaded when the backend first starts
frontend ..... Staff Interface
  assets ..... static assets (such as images, javascript) in the staff interface
  controllers ..... controllers for the staff interface
  locales ..... locale translations for the staff interface
  views ..... templates for the staff interface
  plugin_init.rb ..... if present, loaded when the staff interface first starts
public ..... Public User Interface
  assets ..... static assets (such as images, javascript) in the public interface
  controllers ..... controllers for the public interface
  locales ..... locale translations for the public interface
  views ..... templates for the public interface
  plugin_init.rb ..... if present, loaded when the public interface first starts
migrations ..... Database migrations
schemas ..... JSONModel schema definitions
search_definitions.rb Advanced search fields
```





What if I can't use plug-ins?

If you don't host your own ArchivesSpace or if your IT support is skeptical or overwhelmed, what then?

- Talk to your IT folks/host
 - Build trust/confidence with small steps
 - Try to work with them to use a plug-in that does something small or is only needed one time
- Submit individual feature requests to build what you want into the application
 - <https://archivesspace.atlassian.net/wiki/spaces/ADC/pages/19202060/How+to+Request+a+New+Feature>





What if I can't use plug-ins?

- Build community support for what you want in the application
- Build community partnerships to convert a plug-in that does what you want to a standard distribution plug-in (the way the LCNAF plug-in works) or core code





Getting help

- The ArchivesSpace community wants to help make your plug-in dreams come true
 - Core Committers
 - Users Group listserv and Google Group
- Slack channels, especially Archivists Working with Archival Data (shoes-untied.slack.com)
- Wider GitHub community





Getting a wider audience

- Think about whether your plug-in could be helpful beyond your own institution (most can!)
- If it can
 - Promote it via the listservs
 - Consider a write-up for a monthly update or blog post
 - Bring it up in an open call
 - Work with the community to make the case for it to be part of the standard distribution of ArchivesSpace or core code





Plug-ins resources

- https://github.com/archivesspace/archivesspace/blob/master/plugins/PLUGINS_README.md
- <https://archivesspace.atlassian.net/wiki/spaces/ADC/pages/17137734/Plugins+and+Scripts>
- <https://github.com/archivesspace/awesome-archivesspace>
- <https://www.youtube.com/watch?v=hWP430Q5EWM>
- <http://campuspress.yale.edu/yalearchivesspace/category/what-archivesspace-does/>
- <http://libraryblogs.is.ed.ac.uk/librarylabs/tag/archivesspace/>
- <http://archival-integration.blogspot.com/2015/07/archivesspace-donor-details-plugin.html>
- <https://guides.nyu.edu/archivesspace/development>
- <https://blogs.library.duke.edu/bitstreams/2016/09/21/archivesspace-api-fun/>
- <https://blogs.harvard.edu/archivaldescription/2017/01/26/spreadsheet-to-ead-to-as/>
- <https://rubyexample.com/user/djpillen>
- <https://library.osu.edu/blogs/it/category/archivesspace/>
- Turning a plug-in into core code:
<https://archivesspace.atlassian.net/wiki/spaces/ADC/pages/349995159/Turning+an+ArchivesSpace+Plugin+into+Core+Code>





Thank you!

Any questions?

Contact Information:

Laney McGlohon – laney.mcglohon@lyrasis.org



ArchivesSpace
a community served by ✦ LYRASIS