

# **Transforming External Legacy Metadata for ArchivesSpace**

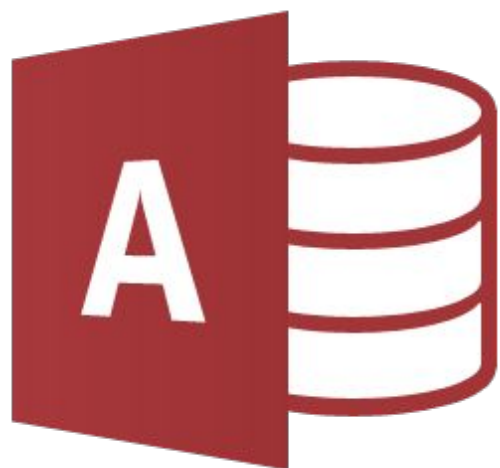
*A labor of love*

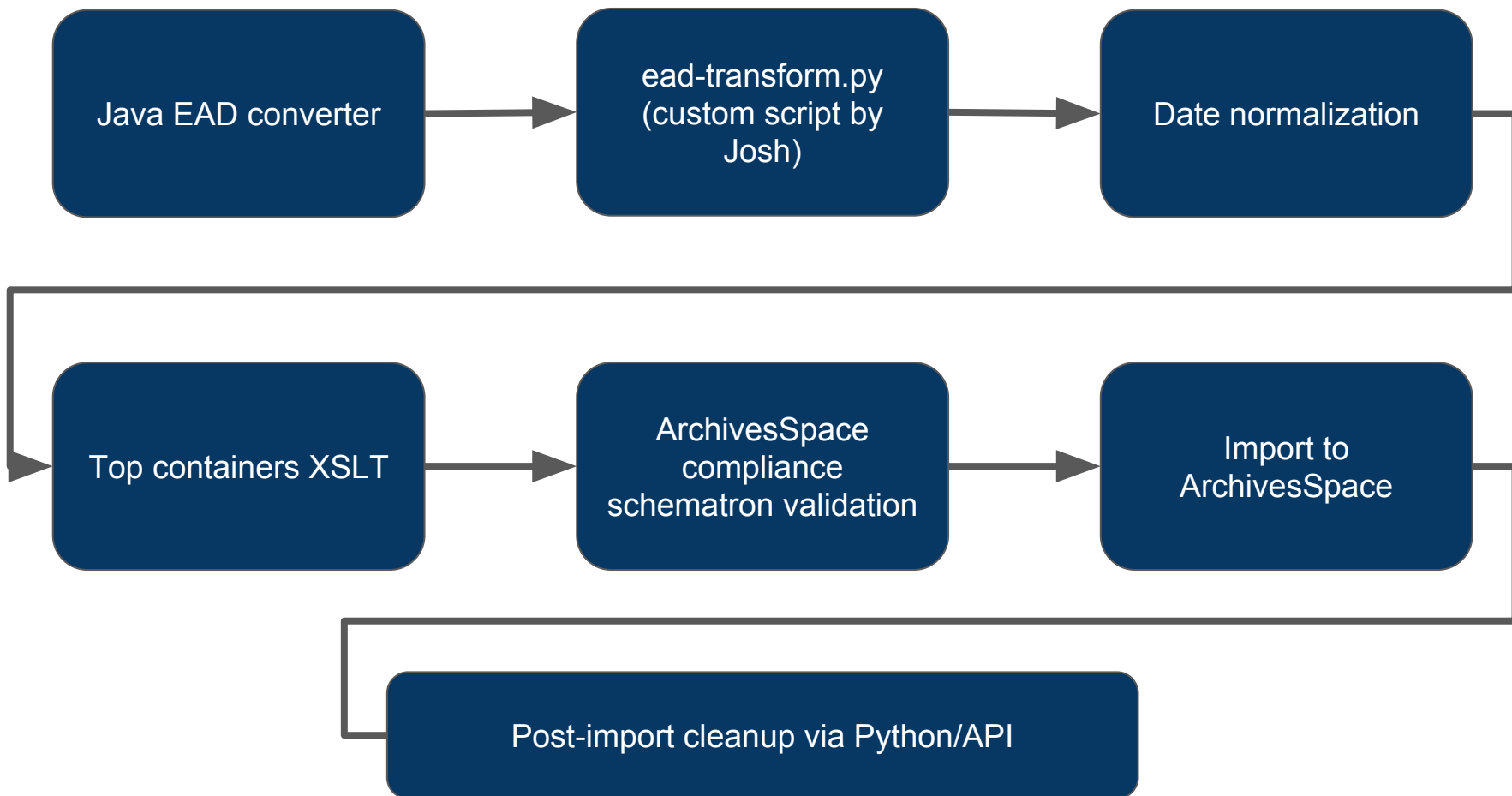
*Bria Parker, Discovery and Metadata Services*

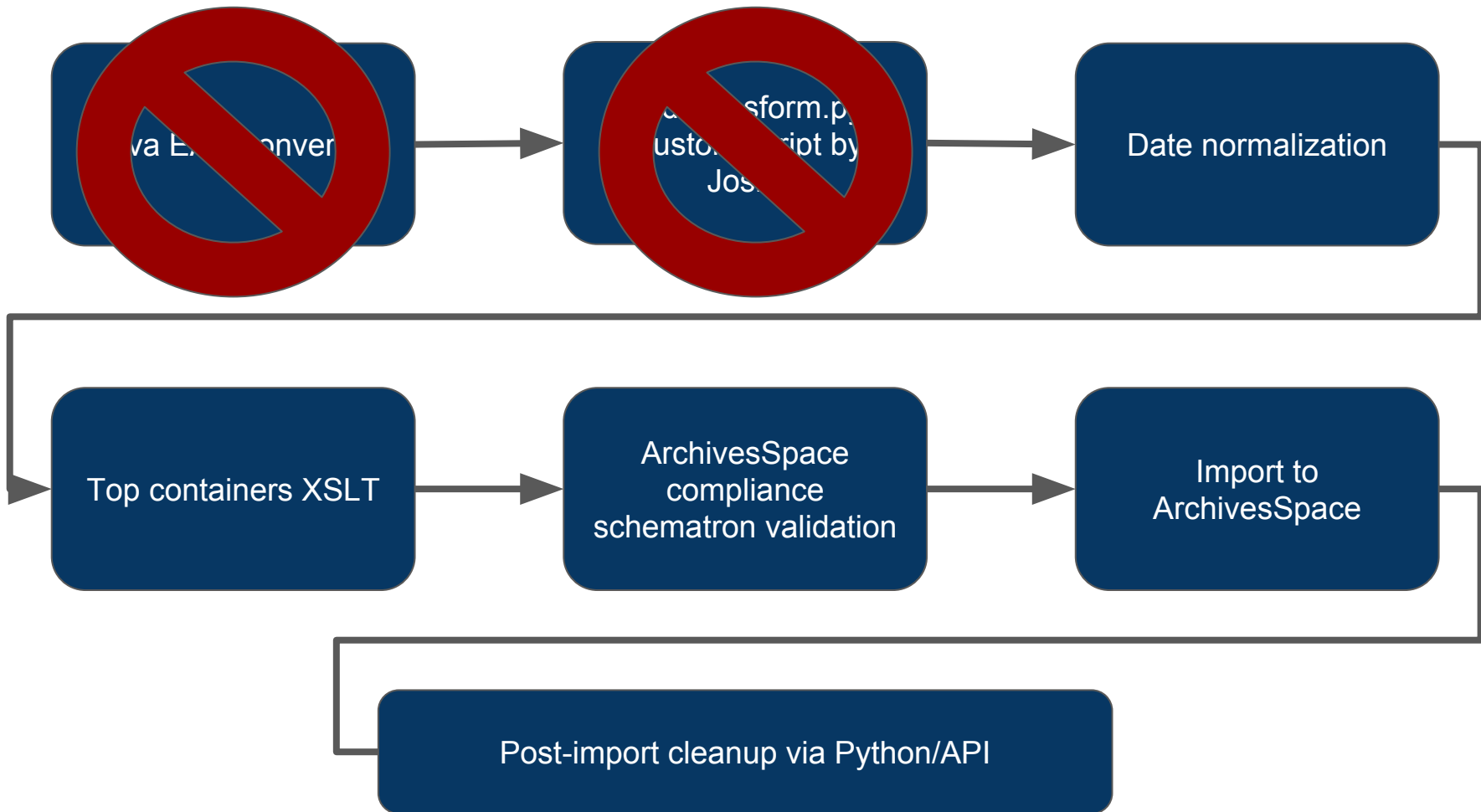


ArchivesSpace

a community served by ✦ LYRASIS



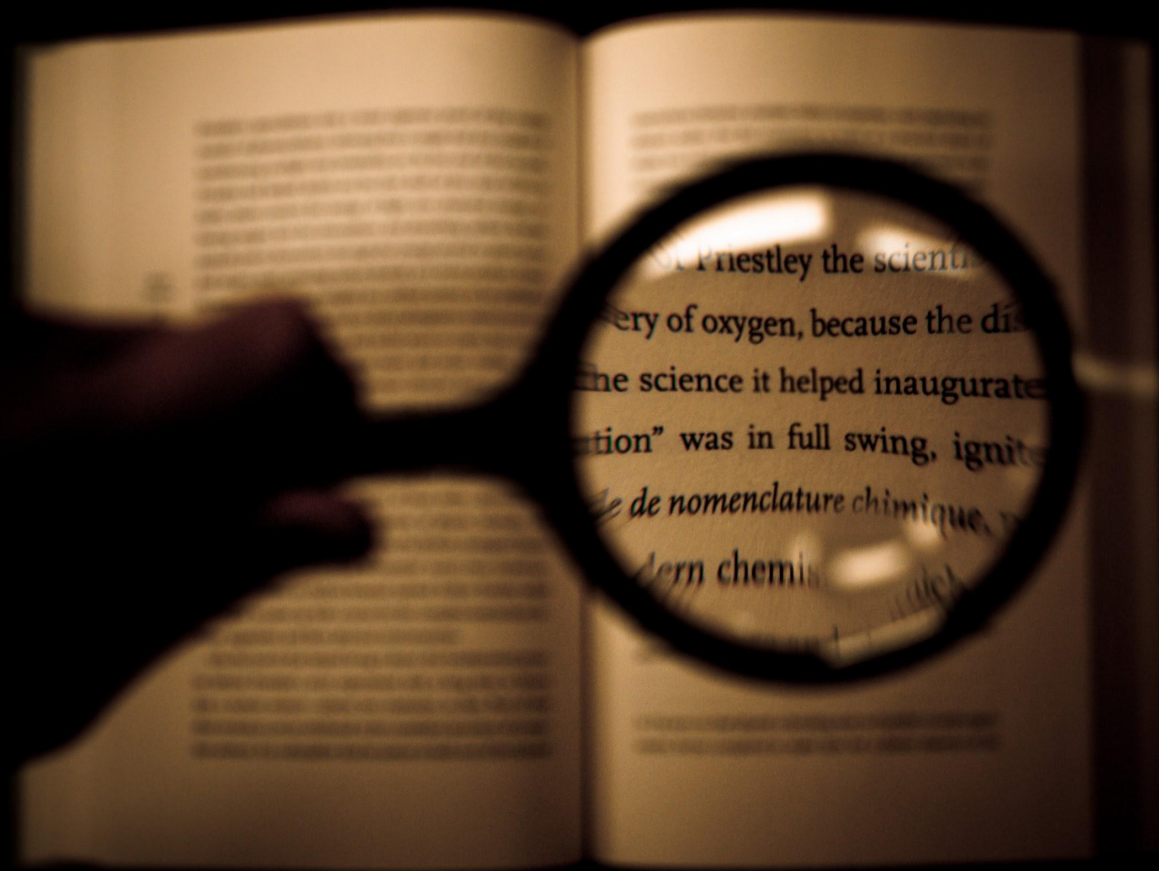






Woman at International Ladies Garment Workers Union march - <http://hdl.handle.net/1903.1/32553>

[illegible]



# Challenges

- Spreadsheets not in the same format as those from The Beast
  - (plus, we had a converter we could use to get from The Beast to EAD)
- Database exports needed to be separated out from the master tables by collection and merged.
  - This was already done by SCUA
- No .csv import of finding aid data function native in ArchivesSpace
  - Plugins available, but that required developer time that was not available

# Steady converts CSV into EAD XML

## Upload

Choose a CSV file to upload. The file must have headers which conform to the [stead schema](#). If you have your container list data in Excel, you will need to export it as CSV before uploading it.

You can try it out with a [sample CSV file](#).

Name\*

Email\*

Organization

CSV\*

No file chosen

To import into ArchivesSpace and have instances created properly, you may need to make some other changes to the EAD. You can select and fill out the following options to try to do that for you. This is currently an experimental feature.

Add identifiers to containers for parent/child relationships? ☐

archdesc/did/unitid

## About

### Use Cases

In most cases tools like ArchivesSpace are preferred because they centralize and normalize the creation of archival description and arrangement. We have two main use cases for maintaining container lists in spreadsheets instead of working directly in ArchivesSpace.

1. Sometimes donors provide container lists for their collections as a spreadsheet. In other cases hired appraisers will provide detailed spreadsheets which could also be used as container lists. We want to be able to edit those as little as possible to allow them to form the basis of a preliminary inventory. In this way in MPLP-style we can make the collection accessible as quickly as possible.
2. NCSU Libraries stores some of its architectural collections like drawings in off-site storage. With poor wireless coverage in some areas it is more convenient to enter container information in a spreadsheet off-line. Also this allows several people to be working on a single collection resource at one time. The spreadsheets can then be merged, exported to CSV, and transformed to EAD XML.

### Code

Steady is a Ruby on Rails application and is available as open source software. [See the full source code](#).

### Contact

This little utility was developed by Jason Ronallo as part of his work at NCSU Libraries.

Email: [jronallo@gmail.com](mailto:jronallo@gmail.com)



by owlpacino ([CC-BY-ND 2.0](https://creativecommons.org/licenses/by-nd/2.0/))  
<https://flic.kr/p/6vwycW>



## Additional arguments

- Collection level
  - Resource id
  - Extent
  - Scope
  - Normalized dates

```
#!/usr/bin/env ruby
```

```
$LOAD_PATH.unshift File.join(File.dirname(__FILE__), '..', 'lib')  
require 'pp'  
require 'stead'  
require 'trollop'
```

```
opts = Trollop::options do  
  banner <<--EOS
```

This script takes a csv file with a name in the format <eadid>\_container\_list.csv and creates a stub EAD XML document.

Usage:

```
csv2ead --csv /path/to/<eadid>_container_list.csv [options]
```

where options are:

EOS

```
opt :csv, "A CSV file", :required => true, :type => String  
opt :baseurl, 'Base URL for adding on the eadid', :type => String  
opt :url, 'Full URL for this collection guide', :type => String  
opt :template, 'Specify using a different EAD XML template', :type => String  
opt :ncsu, 'Use NCSU specific template'  
opt :extra, 'Full path to a Stead::Extra file to add in other data', :type => String  
opt :output, 'Save the file by specifying the filename', :type => String  
opt :pretty, 'If --output is specified this will pretty indent the container list.'  
opt :stdout, 'Output full EAD to terminal'  
opt :idcontainers, 'Add id attributes to containers to show parent child relationships among containers in same com  
opt :unittitle, 'Add a unittitle to ead/eadheader/archdesc/did/unittitle', :type => String  
opt :unitid, 'Add a unitid to ead/eadheader/archdesc/did/unitid', :type => String  
opt :extent, 'Add an extent to ead/eadheader/archdesc/did/physdesc/extent', :type => String  
opt :unitdate, 'Add a unitdate to ead/eadheader/archdesc/did/unitdate', :type => String  
opt :begindate, 'Add a start date to ead/eadheader/archdesc/did/unitdate/@normal', :type => String  
opt :enddate, 'Add an end date to ead/eadheader/archdesc/did/unitdate/@normal', :type => String  
opt :scope, 'Add a scopecontent note to ead/archdesc/scopecontent/p', :type => String
```

```

def collectiondate_node
  @ead.xpath('//xmlns:archdesc/xmlns:did/xmlns:unitdate').first
end

def add_normal_date
  if @begindate
    collectiondate_node['normal'] = @begindate + "/" + @enddate
  end
end

def scope_node
  @ead.xpath('//xmlns:archdesc/xmlns:scopecontent/xmlns:p').first
end

def add_scope
  if @scope
    scope_node.content = @scope
  end
end

def to_ead
  @ead = template.dup
  add_eadid
  add_eadid_url
  add_archdesc_unittitle
  add_title_proper
  add_archdesc_unitid
  add_eadid
  add_archdesc_extent
  add_normal_date
  add_archdesc_unitdate
  add_scope
  @dsc = @ead.xpath('//xmlns:archdesc/xmlns:dsc')[0]
  if series?
    add_series
    add_subseries if subseries?
  end
  @component_parts.each do |cp|
    c = node(file_component_part_name(cp))
    c['level'] = 'file'
  end
end

```

```
def add_series
  add_arrangement
  series = @component_parts.map do |cp|
    [cp['series number'], cp['series title'], cp['series dates']]
  end.uniq
  series.each do |ser|
    add_arrangement_item(ser)
    # create series node and add to dsc
    series_node = node('c01')
    @dsc.add_child(series_node)
    series_node['level'] = 'series'
    # create series did and add to series node
    series_did = node('did')
    series_node.add_child(series_did)
    unitid = node('unitid')
    unitid.content = ser[0]
    unittitle = node('unittitle')
    unittitle.content = ser[1]
    unitdate = node('unitdate')
    unitdate.content = ser[2]
    series_did.add_child(unitid)
    series_did.add_child(unittitle)
    series_did.add_child(unitdate)
  end
end
```

```

def add_series
  add_arrangement
  series = @component_parts.map do |cp|
    [cp['series number'], cp['series title'], cp['series start'], cp['series end'],
     cp['extent'], cp['scopecontent'], cp['persname'], cp['corpname'], cp['subject']]
  end.uniq
  series.each do |ser|
    add_arrangement_item(ser)
    # create series node and add to dsc
    series_node = node('c01')
    @dsc.add_child(series_node)
    series_node['level'] = 'series'
    series_node['id'] = 'series' + ser[0]
    # create series did and add to series node
    series_did = node('did')
    series_node.add_child(series_did)
    unitid = node('unitid')
    unitid.content = ser[0]
    unittitle = node('unittitle')
    unittitle.content = ser[1]
    series_did.add_child(unitid)
    series_did.add_child(unittitle)

    unitdate= node('unitdate')
    if ser[2]
      unitdate.content = ser[2] + '-' + ser[3]
      unitdate['normal'] = ser[2] + '/' + ser[3]
      series_did.add_child(unitdate).attr('normal')
    end
  end
end

```

```

physdesc = node('physdesc')
extent= node('extent')
if ser[4]
  extent.content = ser[4]
  series_did.add_child(physdesc)
  physdesc.add_child(extent)
end

scopecontent = node('scopecontent')
series_node.add_child(scopecontent)
if ser[5]
  serieshead = node('head')
  scopecontent.add_child(serieshead)
  serieshead.content= 'Scope and Contents'
  p = node('p')
  scopecontent.add_child(p)
  p.content = ser[5]
end

controlaccess = node('controlaccess')
series_node.add_child(controlaccess)
if ser[6]
  ser[6].split(';').each do |c|
    persname = node('persname')
    controlaccess.add_child(persname)
    persname << c
  end
end

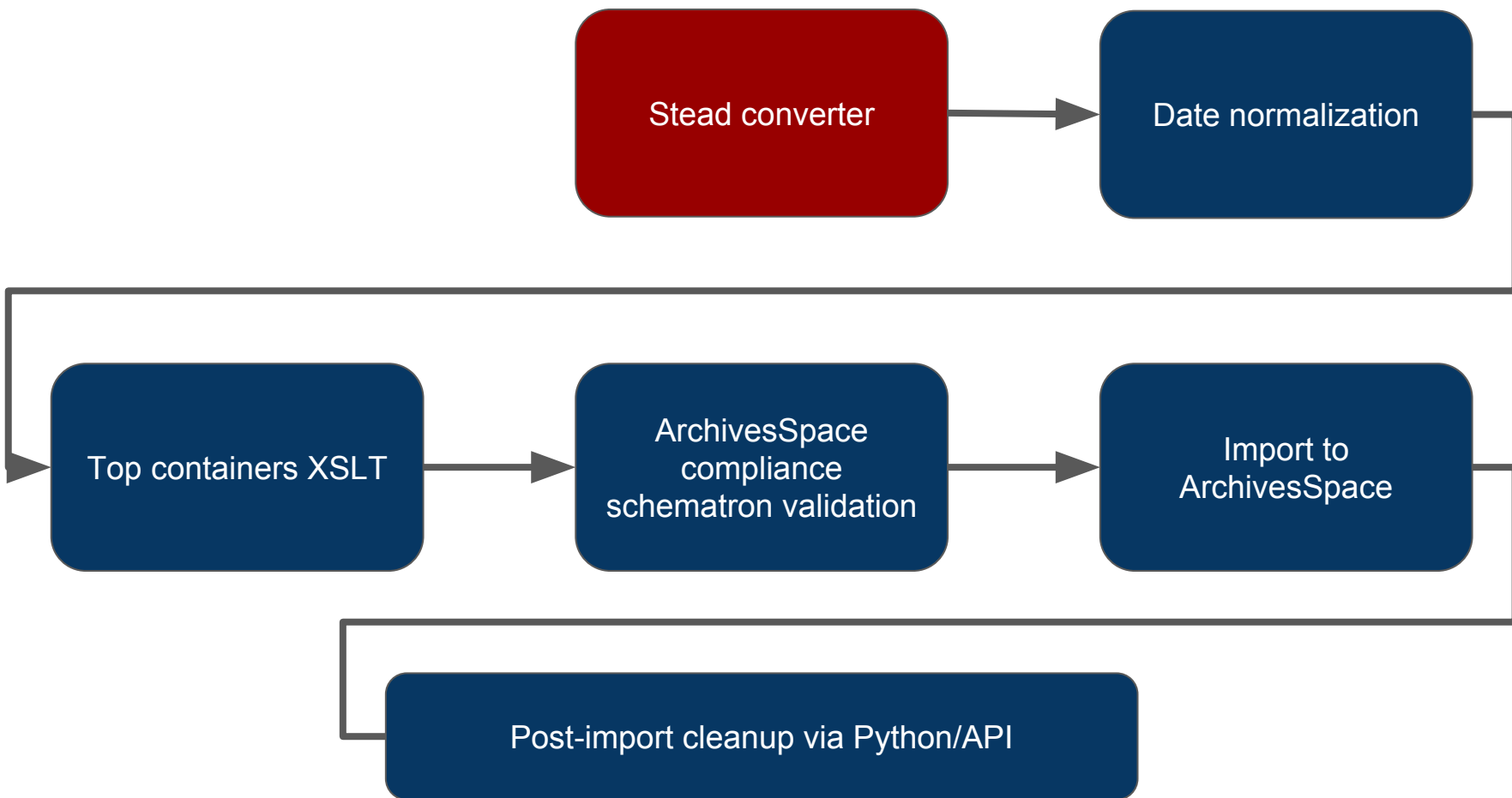
if ser[7]
  ser[7].split(';').each do |c|
    corpname = node('corpname')
    controlaccess.add_child(corpname)
    corpname << c
  end
end

if ser[8]
  ser[8].split(';').each do |c|
    subject = node('subject')
    controlaccess.add_child(subject)
    subject << c
  end
end

```



by gotcredit ([CC-BY 2.0](https://creativecommons.org/licenses/by/2.0/))  
<https://flic.kr/p/SDDTLb>



# Real talk



```
csv2ead -c afl-cio-inventory.csv  
-d RG99-001 -l "[collection title]"  
-g [start date] --enddate [end date]  
-x "extent"  
--scope "[scope and contents note]"  
> output.xml
```

All	Column 1	Column 2	expression	normal
1.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[1]/did/unitdate	January 1956-June 1956	1956-01/1956-06
2.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[2]/did/unitdate	July 1956-January 1957	1956-07/1957-01
3.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[3]/did/unitdate	May 1956-June 1957	1956-05/1957-06
4.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[4]/did/unitdate	March 1958	1958-03
5.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[5]/did/unitdate	April 1958	1958-04
6.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[6]/did/unitdate	January 1956-June 1956	1956-01/1956-06
7.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[7]/did/unitdate	August 1956-December 1956	1956-08/1956-12
8.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[8]/did/unitdate	January 1957-February 1957	1957-01/1957-02
9.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[9]/did/unitdate	April 1957-August 1957	1957-04/1957-08
10.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[10]/did/unitdate	January 1958-February 1958	1958-01/1958-02
11.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[11]/did/unitdate	1959	
12.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[12]/did/unitdate	January 1960-October 1962	
13.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[13]/did/unitdate	December 1955-April 1958	
14.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[14]/did/unitdate	December 1955-April 1958	
15.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[15]/did/unitdate	February 1956-May 1956	
16.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[16]/did/unitdate	February 1956-April 1957	
17.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[17]/did/unitdate	March 1957-June 1957	
18.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[18]/did/unitdate	February 1956	
19.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[19]/did/unitdate	1957	
20.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[20]/did/unitdate	August 1956-December 1956	
21.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[21]/did/unitdate	November 1957-April 1958	
22.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[22]/did/unitdate	December 1955-April 1958	
23.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[23]/did/unitdate	August 1956-October 1957	
24.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[24]/did/unitdate	December 1955-February 1957	
25.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[25]/did/unitdate	February 1956-May 1958	
26.	RG13-005.xml-transform.xml	/ead/archdesc/dsc/c01[1]/c02[26]/did/unitdate	June 1957-January 1958	

```

import csv
from lxml import etree
from os.path import join

path = 'rg97' # Enter the path to your EAD directory here

normalized_file = '97-002-dates.csv' # Enter the path to the exported CSV here

with open(normalized_file, 'rb') as csvfile:
    reader = csv.reader(csvfile)
    next(reader, None) # Skip the header row
    for row in reader: # Loop through the rows in the csv
        filename = row[0]
        date_path = row[1]
        # uncomment the line below if you also cleaned up the date expression
        expression = row[3]
        normalized = row[4]
        ead = open(join(path, filename)) # Open the EAD
        tree = etree.parse(ead)
        date = tree.xpath(date_path)
        # uncomment the line below if you also cleaned up the date expression
        date[0].text = expression
        date[0].attrib['normal'] = normalized # Add a normal attribute with the normalized date
        outfile = open(join(path, filename), 'w')
        outfile.write(etree.tostring(tree, encoding="utf-8", xml_declaration=True)) # R
        outfile.close()

print "Normalization complete" # Woo!

```

<unitdate normal="1957-08/1958-05">August 1957-May 1958</unitdate>

```

        <xsl:attribute name="label">
          <xsl:copy-of
            select="concat('audio ', '(', //ead/eadheader/eadid, '.', self::container[@type
          </xsl:attribute>
        <xsl:attribute name="id">
          <xsl:copy-of
            select="concat(self::container[@type = 'box'], '.', self::container[@type = 'bo
          </xsl:attribute>
        <xsl:attribute name="type">box</xsl:attribute>
        <xsl:copy-of select="self::container[@type = 'box']"/>
        <xsl:value-of select="."/>
      </xsl:copy>
    </xsl:when>
    <xsl:when test="@label = 'mixed_materials'">
      <xsl:copy>
        <xsl:attribute name="label">
          <xsl:copy-of
            select="concat('mixed_materials ', '(', //ead/eadheader/eadid, '.', self::conta
          </xsl:attribute>
        <xsl:attribute name="id">
          <xsl:copy-of
            select="concat(self::container[@type = 'box'], '.', self::container[@type = 'bo
          </xsl:attribute>
        <xsl:attribute name="type">box</xsl:attribute>
        <xsl:value-of select="."/>
      </xsl:copy>
    </xsl:when>
  </xsl:when>

```

## Log

```
Created: /repositories/import/archival_objects/import_1b69ad08-3ff6-4f34-a978-9b3144012628
Created: /repositories/import/archival_objects/import_4ec1a509-bbdd-4f4e-b40d-25c1fd5518e7
Created: /repositories/import/archival_objects/import_210f3c23-9965-4d53-8d93-c794e307ec8f
Created: /repositories/import/archival_objects/import_79057e3e-a1f4-4179-bcc6-72e5fd5d5052
Created: /repositories/import/archival_objects/import_ff38b7f3-0ead-4f90-94fb-836d94c72e87
Created: /repositories/import/archival_objects/import_1b0b377d-0715-4fda-a4e6-a719ef080465
6. DONE: Saving records: cycle 2
7. STARTED: Saving records: cycle 3
Created: /repositories/import/archival_objects/import_a71de3dc-b538-484c-a435-27eeddfd1450
Created: /repositories/import/archival_objects/import_41b0067c-1287-4355-801b-cde0fb3a6846
Created: /repositories/import/archival_objects/import_4ebc5f5b-029e-4ec8-aa39-cf9e088b15eb
Created: /repositories/import/archival_objects/import_e521069d-5562-49a9-88c0-109c6ff7bc35
Created: /repositories/import/archival_objects/import_cf1dd47c-9dd5-46bf-95d2-153f30577048
Created: /repositories/import/archival_objects/import_ab17f3f2-03b0-4985-9b14-b91d130c8e35
Created: /repositories/import/archival_objects/import_caa53699-8329-4cb9-8c6c-b088c5c2bbf9
Created: /repositories/import/archival_objects/import_3caaea72-16cb-4fd9-902e-eb6a688c432b
Created: /repositories/import/archival_objects/import_07867f59-7b84-4811-a719-aed7022c0cf3
```

## New & Modified Records

- Товарищи конец
- International Labor News Service clipsheet 28
- La Chienlit C'est Lui (He is the chaos)
- International Labor News Service clipsheet 29
- Työmaana maailma (translation: [The worksite is the world])
- International Labor News Service clipsheet 30
- Valitse Puhtaat Ympäristö (translation: Choose a healthier environment)

[Libraries](#)[Collections](#)[Digital Material](#)[Help](#)

## AFL and AFL-CIO Organization and Field Services Department, State Charter records

 **Collection 0021-LBR-RG28-005**



Citation



Request



Print

[Special Collections and University Archives](#) | AFL and AFL-CIO Organization and Field Services Department, State Charter records

[Collection Overview](#)[Collection Contents](#)[Container Inventory](#)

The Organizing Department was given responsibility for administering charter records in the 1970s. This collection contains copies of AFL, CIO, and AFL-CIO state charters and related correspondence gathered or generated by the AFL-CIO Organization and Field Services Department. AFL charter materials date 1890-1955, CIO materials from 1937-1956, and AFL-CIO materials from 1956-1985. The bulk of all materials date from the 1930s to the 1950s ...[See more >](#)

### Dates

1890-1985

### Conditions Governing Access

This collection is open to the public and must be used in the Special Collections reading room. Researchers must register and agree to copyright and privacy laws before using this collection.

Portions of this collection are stored offsite. Request to use this material must be submitted 48 hours in advance of visit.

### Conditions Governing Use

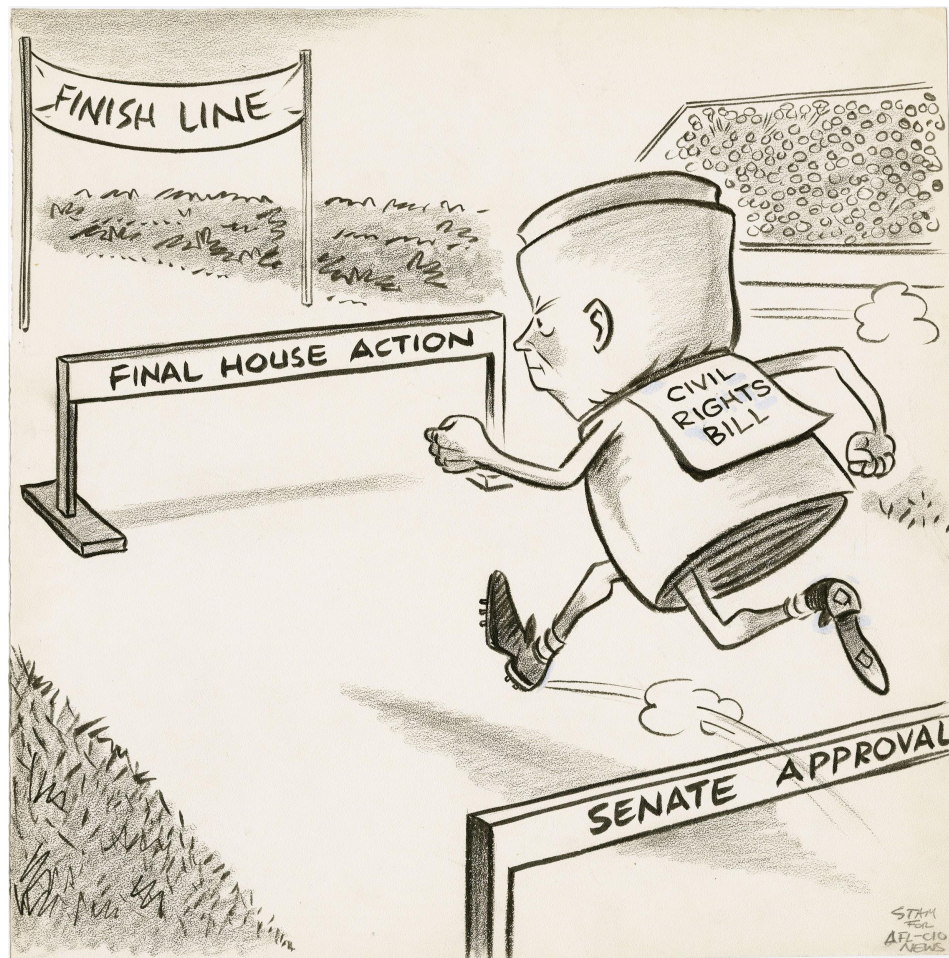
Years:  

### Collection organization

[AFL and AFL-CIO Organization and Field Services Department, State Charter records](#)[Alabama, AFL, October, 1900](#)[Alabama, CIO, December, 1937 - September, 1955](#)[Alabama, AFL-CIO, September, 1956 - 1985](#)[Alaska, AFL, April, 1943 - March, 1955](#)

# Next steps

- Clean up code revisions to Stead
  - Documentation
  - Share back with Jason Ronallo/NCSU
- Problem items
  - Some collections require A LOT of additional manual work to get into the template and go through the Stead conversion
- Final review by Labor.
  - Everything is getting reviewed, and having content that could not be added via conversion added after the fact prior to being published.



Cartoon of Civil Rights Bill...ALF-CIO News, <https://hdl.handle.net/1903.1/40014>



# Acknowledgements

- Megan O'Hern
- Sydney Vaile
- Caitlin Rizzo
- Joanne Archer
- Jen Eidson
- Dave Mayo

# Questions?

Thanks!

[blparker@umd.edu](mailto:blparker@umd.edu)