Introduction to Metadata Power Tools for the Curious Beginner

Maureen Callahan, Yale University
Regine Heberlein, Princeton University
Dallas Pillen, University of Michigan

Cleveland, August 20, 2015

Introduction

Maureen Callahan

I. Preaching the gospel of power tools - why we use these, how they should be approached, how they ultimately result in a better researcher experience

A. Why we use these

1. So much messy data, so little time

a) Getting strategically lazy

Basic Principles of Working with Power Tools

Dallas Pillen

Create a Sandbox Environment

- Backups
- Make it truly not matter if something goes wrong
- It’s okay to break things -- you will learn in the process

Think Algorithmically

- Think of archival data as data
- Envision success
  - What does the data look like now?
  - What should it look like when you're finished?
- Break a big problem down into smaller steps
  - Write out the steps in natural language

Choosing a Tool

- A lot of metadata clean-up can be accomplished with a variety of tools
- Often, there is no ‘best’ tool for the job
- The best tools are the ones that work for your particular issue and existing skill set
  - What is your issue?
  - What tools do you already know?
  - How can you apply those tools to the issue?
  - If necessary, what tools do you need to learn?

Document...

- Successes
- Failures
- Procedures
- If you tried Googling something and couldn’t find an answer, that means there is a documentation gap. Fill it!

The Gospel of Metadata

Regine Heberlein

1. Know Thine Data Like Thyself

uncontrolled value

siblings

AT-generated

string

```
<c level="file" id="C1455_c300">
  <did>
    <unittitle>Personal Correspondence</unittitle>
    <container id="cid264983012" type="box">18</container>
    <container parent="cid264983012" type="folder">21</container>
    <unitdate type="inclusive" normal="1970/1979">1970-1979</unitdate>
    <physdesc>
      <extent type="computed">1 folder</extent>
    </physdesc>
  </did>
</c>
```

2. Know Your Heart's Desire

```
<c level="file" id="C1455_c300">
  <did>
    <unittitle>Personal Correspondence</unittitle>
    <container type="folder" parent="C1455_i18">21</container>
    <unitdate type="inclusive" normal="1970/1979">1970-1979</unitdate>
    <physdesc>
      <extent type="computed">1 folder</extent>
    </physdesc>
  </did>
</c>

<c level="otherlevel" otherlevel="item" id="C1455_i18">
  <did>
    <container type="box">18</container>
    <unitid type="barcode">32101080829854</unitid>
    <physloc type="code">rcpxm</physloc>
  </did>
</c>
```

3. Focus on the Logic, Not the Tools

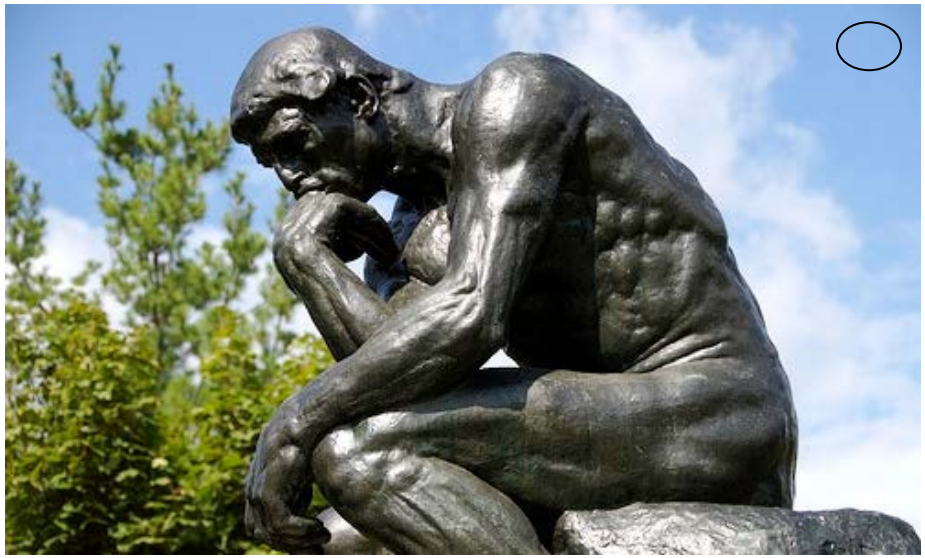
A

```
<c level="file" id="C1455_c300">
  <did>
    <unittitle>Personal Correspondence</unittitle>
    <container id="cid264983012" type="box">18</container>
    <container parent="cid264983012" type="folder">21</container>
    <unitdate type="inclusive" normal="1970/1979">1970-1979</unitdate>
    <physdesc>
      <extent type="computed">1 folder</extent>
    </physdesc>
  </did>
</c>
```

B

```
<c level="otherlevel" otherlevel="item" id="C1455_i18">
  <did>
    <container type="box">18</container>
    <unitid type="barcode">32101080829854</unitid>
    <physloc type="code">rcpxm</physloc>
  </did>
</c>
```

Hmmm…



https://www.flickr.com/photos/stevenfettig/1390275600

4. Choose the Path of Least Resistance

XSLT?

XQuery?

OpenRefine?

Find/Replace?

Ha!
Just what
I need!

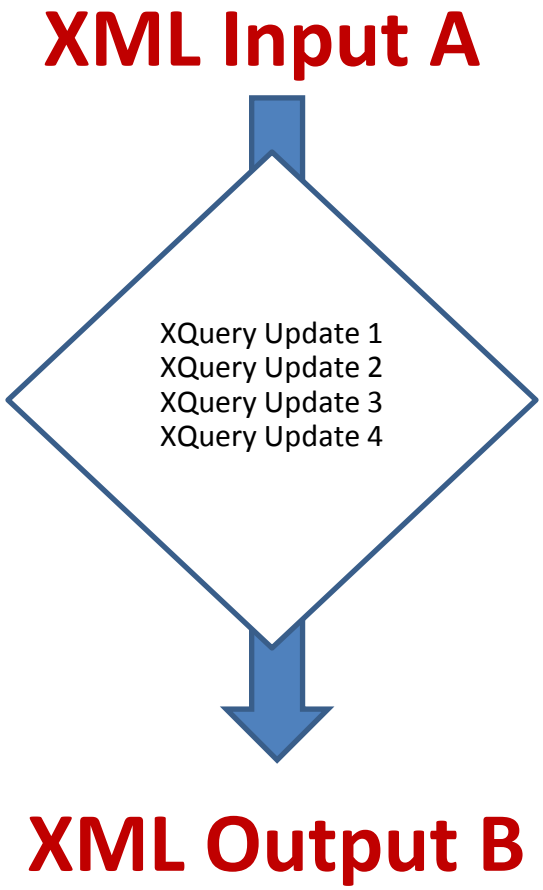
Image: http://www.corporate-eye.com/main/a-macys-mystery-continued/

5. Think About the Future

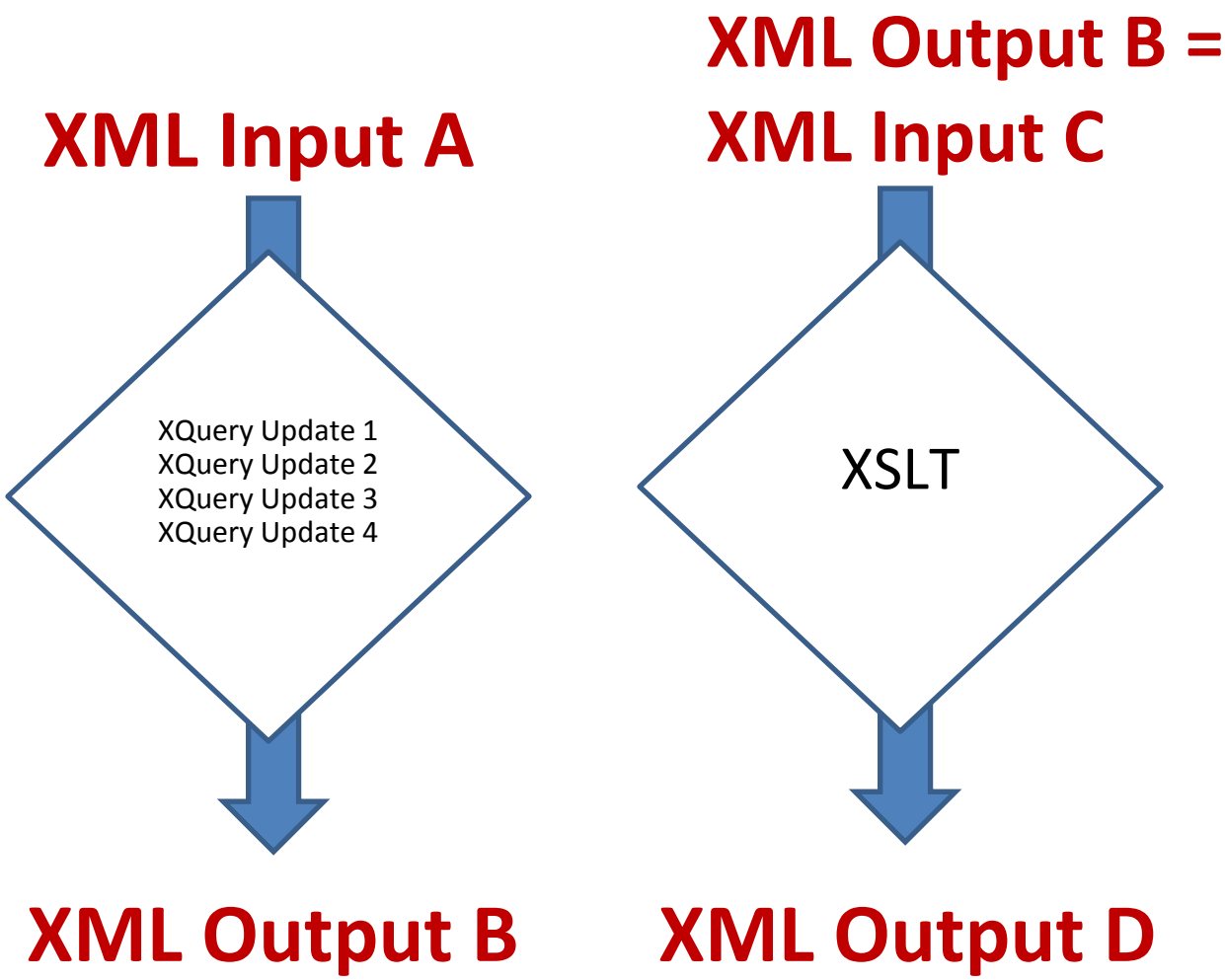
Image: http://www.basf-new-business.com/

- Is learning this tool an investment?
- Is writing this tool an investment?
- Can the tool be used again?
- Does the situation call for a quick, one-off solution or a slower yet permanent tool?
- What is the impact on staff skills?
- How about other workplace consequences?

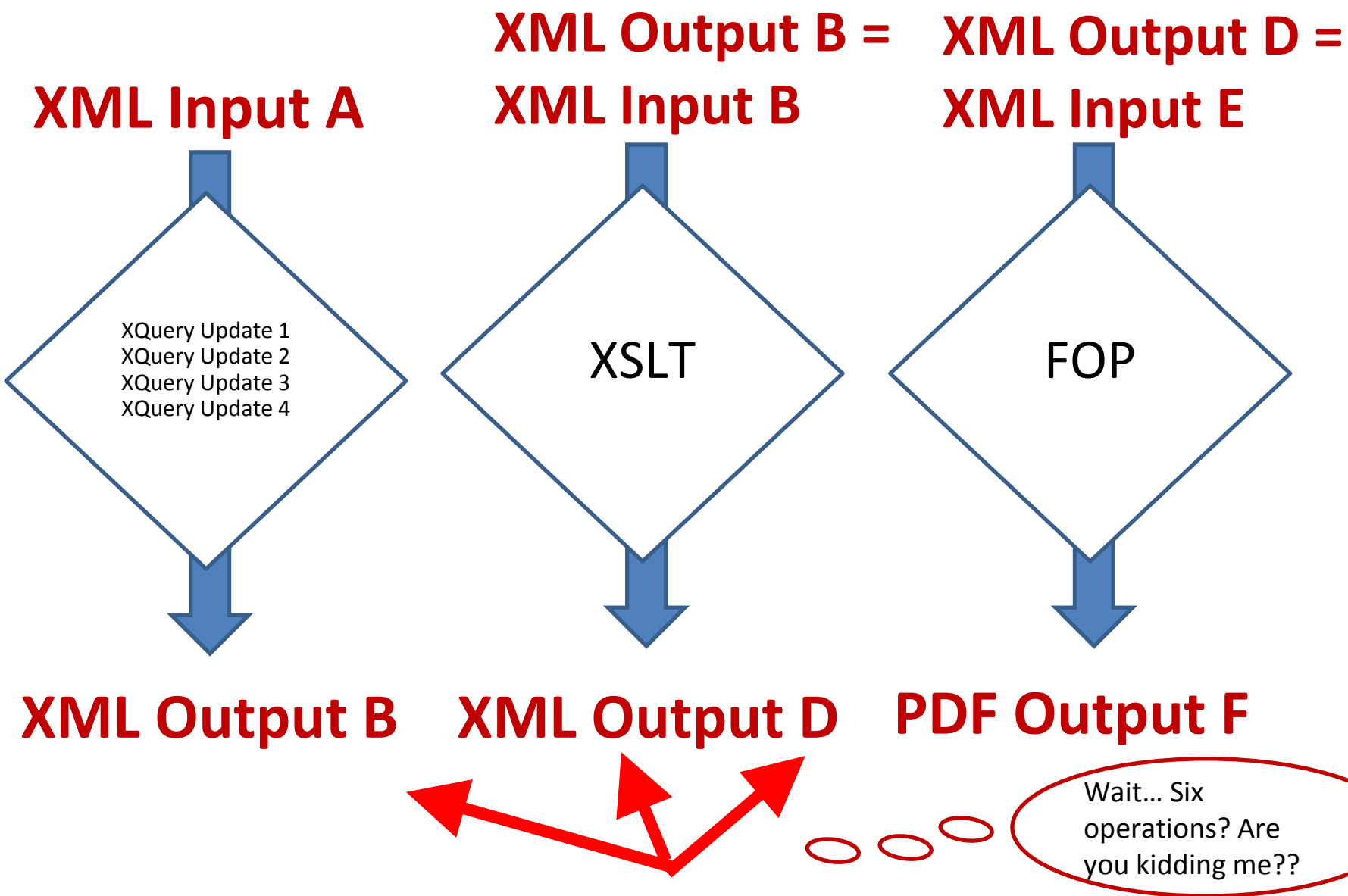
6. Think Like a Machine



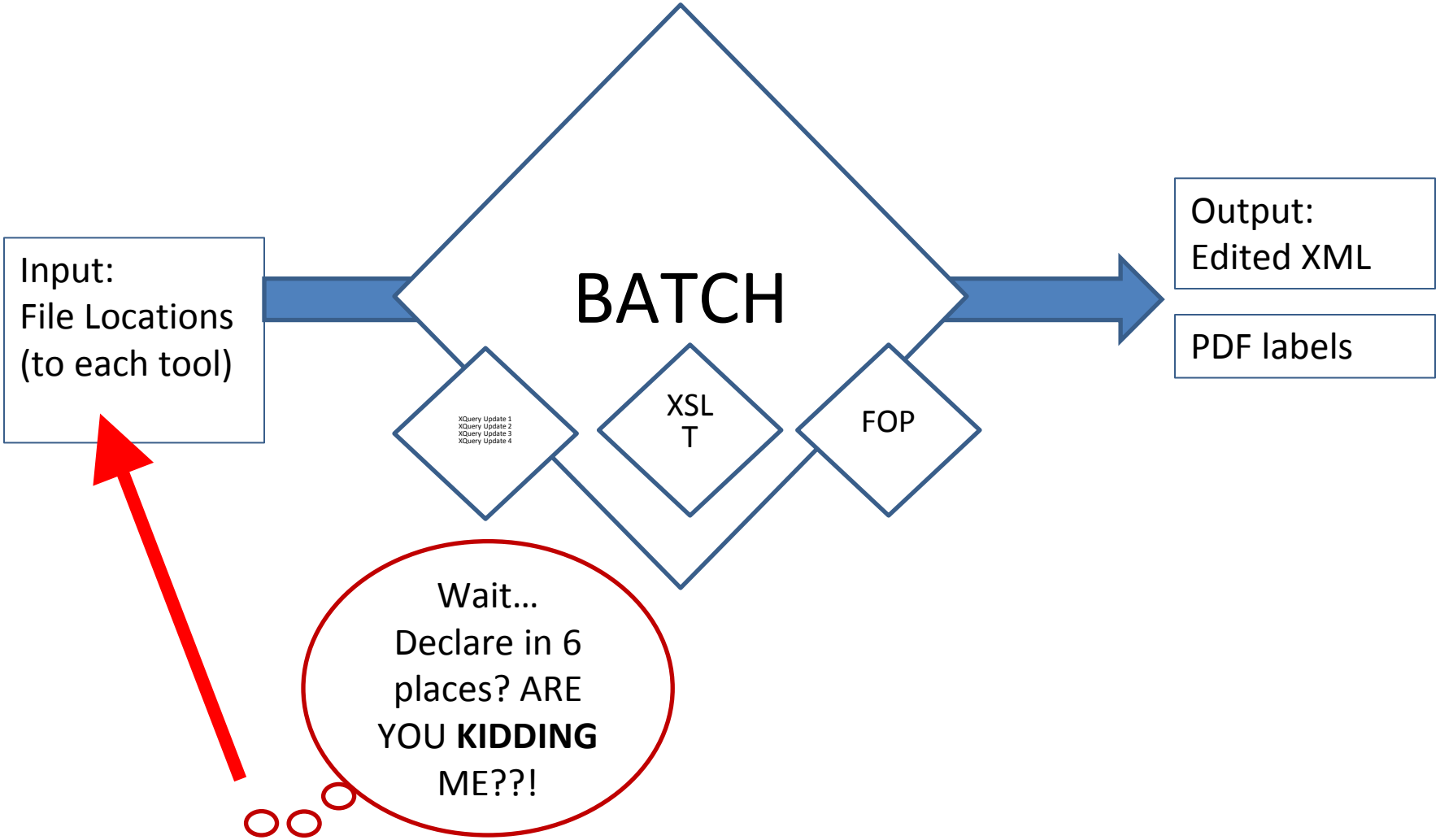
6. Think Like a Machine



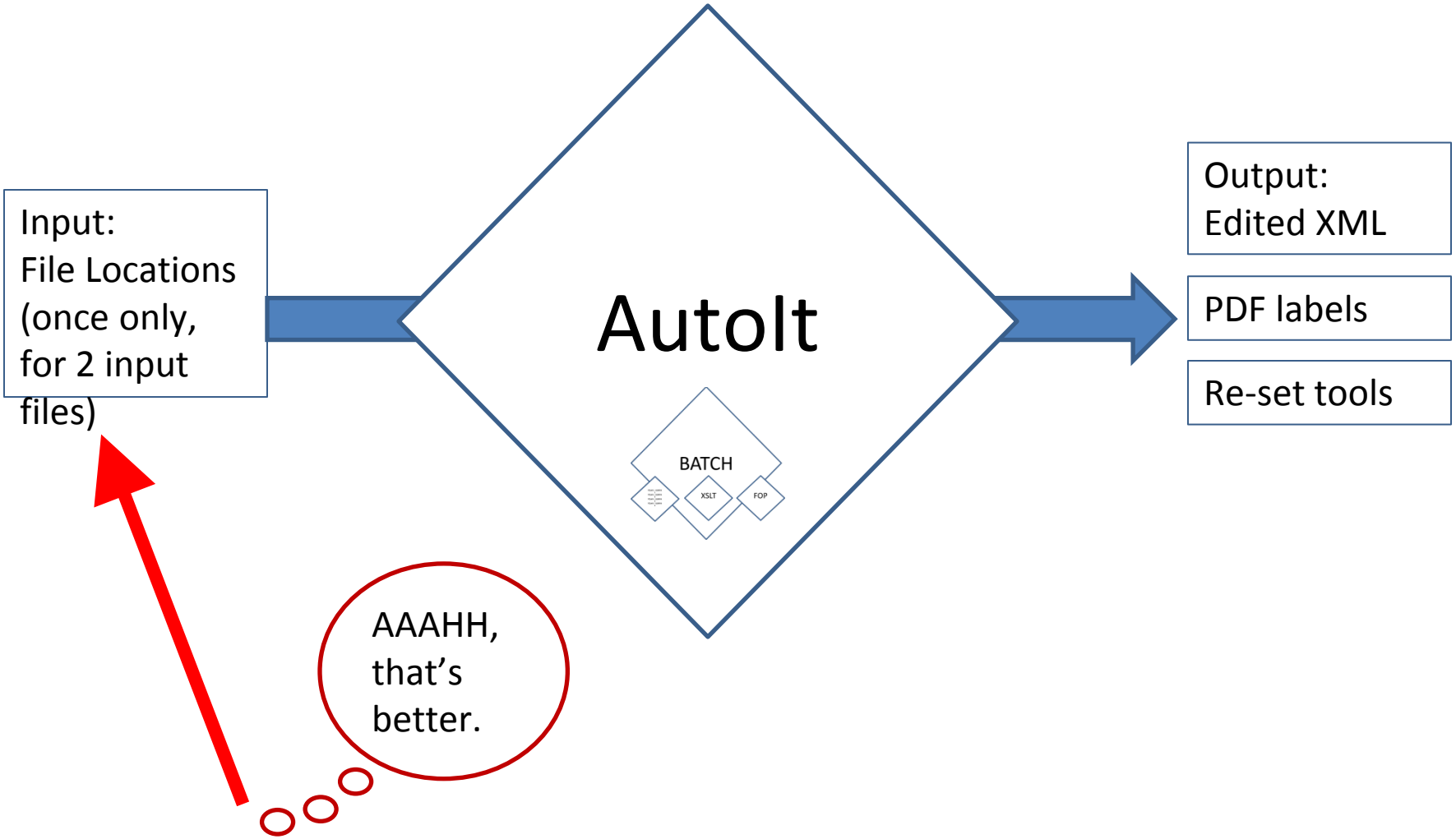
6. Think Like a Machine



7. If a Machine Can Do It...



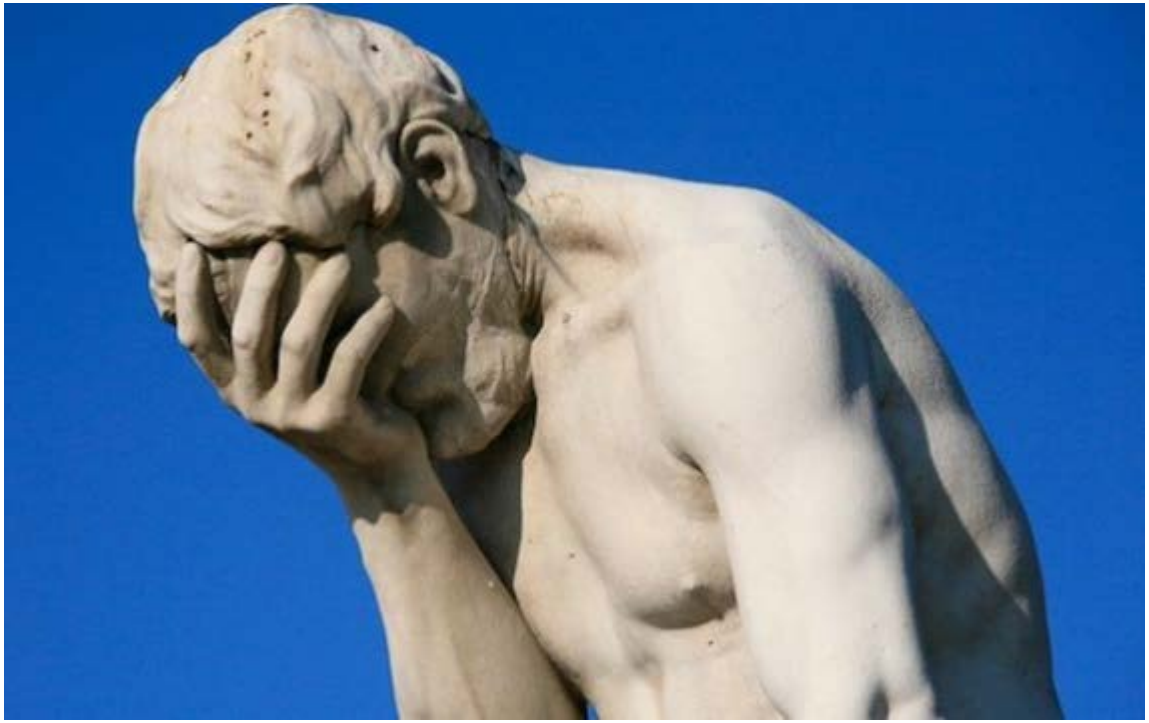
…Let It.


Input:
File Locations
(once only,
for 2 input
files)
AutoIt
BATCH
XSLT
FOP
Output:
Edited XML
PDF labels
Re-set tools
AAAHH,
that's
better.

8. Thou Shalt Steal

- don’t reinvent the wheel
- learn from others
- search the forums
- don’t be afraid to ask for help
- <u>do</u> credit others’ creative/extensive work

9. Dare to Make Mistakes



- that is, as long as you know how to undo / revert / roll back!
- view mistakes as an opportunity
- mistakes can teach you as much about your data as about your tool
- share your mistakes so others may benefit
- realize that everybody makes them

Image: http://lifehacker.com/5863490/how-to-learn-from-your-mistakes

THINGS THAT MIGHT KEEP YOU FROM GETTING STARTED

Maureen Callahan

I. Demonstrations of setting up your environment and using the proper safety equipment

A. If you don’t have admin access…

1. Setting up a virtual environment
2. Making the argument that, really, you should have admin access

Playing It Safe

Maureen Callahan
Regine Heberlein
Dallas Pillen

Using a Local Instance of a Database

Maureen Callahan

a) Local instance of database, e.g. AT or ASpace

Using a Local Versioning System

Regine Heberlein

https://drive.google.com/open?id=0B7vNlf4EPQFeVzFNUFZSbDNDdTA

Using Distributed Version Control

Dallas Pillen

GitHub

- Distributed version control system
- Multiple people can work on the same project
  - Does not totally remove the opportunity for conflicts, but provides a way to identify and resolve them
- Detailed record of what changes were made
- Ability to revert back to a previous state

GitHub at the Bentley Historical Library

- http://archival-integration.blogspot.com/2015/07/git-flow-for-archival-workflows.html
- GitHub is used for version controlling our legacy EAD clean-up project
- Four people working on the project in different ways
  - Manual clean up
  - Automated clean up
- GitHub gives us the ability to work on our own fork of the Bentley's main repository
  - Changes are made to individual forks
  - Changes are then pushed to the main repository where they can be reviewed and merged
- Conflicts still happen -- sometimes two people happen to modify the same file in different ways and submit conflicting pull requests to the Bentley's main repository
  - GitHub helps us identify and resolve these issues

bentley-historical-library / **vandura**

Unwatch 4 | Star 0 | Fork 4

For this project: http://archival-integration.blogspot.com/2015/04/legacy-ead-import-into-archivesspace.html — Edit

1,946 commits | **1** branch | **0** releases | **4** contributors

Branch: **master** | **vandura** / +

Merge pull request **#178** from devonproudfoot/master ...

eckardm authored an hour ago — latest commit 4d25e33933

Real_Masters_all	Corrected container listing	21 hours ago
marc_xml-split	deleted marcxml that had an EAD; moved all webarch marcxml to sep. fo…	2 months ago
web_archives	deleted marcxml that had an EAD; moved all webarch marcxml to sep. fo…	2 months ago
README.md	Update README.md	2 months ago

Code

Issues 0

Pull requests 1

Wiki

Pulse

Graphs

Settings

Commits

Commits on Aug 18, 2015

Merge pull request #178 from devonproudfoot/master ···

eckardm authored an hour ago

4d25e33

Commits on Aug 17, 2015

Corrected container listing

devonproudfoot authored 21 hours ago

c4cede5

Spell check

devonproudfoot authored 21 hours ago

a8b32ae

Corrected container listing

devonproudfoot authored 22 hours ago

a848080

Corrected dates and extent

devonproudfoot authored 22 hours ago

45164fe

Corrected extent, call numbers, spell check

devonproudfoot authored 22 hours ago

caf6afd

Spell check

devonproudfoot authored 22 hours ago

6ba25f2

Compare Changes

```
59  -       <unittitle encodinganalog="245">A. Alfred Taubman College of Architecture + Urban Planning (University of Michigan), records <unitdate type="inclusive" encodinganalog="245$f" normal="1878/2010">1878-2010</unitdate></unittitle>
60          <physdesc>
61  -         <extent encodinganalog="300">80 linear feet, 2 oversized boxes, and 1 flat file drawer and digital files.</extent>
```

```
63  +       <unittitle encodinganalog="245">A. Alfred Taubman College of Architecture + Urban Planning (University of Michigan), records <unitdate type="inclusive" encodinganalog="245$f" normal="1876/2010">1876-2010</unitdate></unittitle>
64          <physdesc>
65  +         <extent encodinganalog="300">80 linear feet, 2 oversized boxes, and 1 flat file drawer and 137.2 MB (online).</extent>
```

Demos

Maureen Callahan
Regine Heberlein
Dallas Pillen

Database Querying

Maureen Callahan

XML Transformations

Regine Heberlein

https://drive.google.com/open?id=0B7vNlf4EPQFebEpRUy1xYXBxd28

Python and Open Refine

Dallas Pillen

The Problem

- <unitdate> tags without normal attributes
  - <unitdate>August 20, 2015</unitdate>
- Dates are only human-, not machine-, readable
  - Many different forms of dates
    - Month DD, YYYY
    - MM/DD/YYYY
    - YYYY Month DD
    - etc., etc.
- Normal attributes allow for the inclusion of a machine-readable form of a date
  - Standardized form for dates: YYYY-MM-DD
  - <unitdate normal="2015-08-20">August 20, 2015</unitdate>

The Solution

- Identify the scope of the problem
- Find a way to normalize a large portion of the dates programmatically
- For the remaining non-normalized dates…
  - Extract <unitdate> text with Python
  - Normalize the dates in OpenRefine
  - Insert normal attributes using Python

https://github.com/djpillen/saa15_metadata_power_tools

djpillen / **saa15_metadata_power_tools**

Unwatch 1

Scripts used in SAA 2015 Pop-Up 5: Metadata Power Tools for the Curious Beginner — Edit

17 commits | **1** branch | **0** releases | **1** contributor

Branch: **master** | **saa15_metadata_power_tools** / +

fixing xpath csv column

djpillen authored 3 hours ago — latest commit `ffeafe448d`

demo_eads	deleting file	4 hours ago
demo_eads_normalized	adding normalized folder	4 hours ago
OpenRefine_Expressions.txt	finalizing files	3 hours ago
README.md	Update README.md	16 hours ago
all_unitdates.csv	fixing xpath csv column	3 hours ago
insert_normalized_dates.py	fixing xpath csv column	3 hours ago
non-normalized_unitdate_count.py	few more modifications	3 hours ago
non-normalized_unitdates.csv	fixing xpath csv column	3 hours ago
normalize_unitdates.py	few more modifications	3 hours ago
print_unitdates.py	file renaming	16 hours ago
write_all_unitdates.py	file renaming	16 hours ago
write_non-normalized_unitdates.py	finalizing files	3 hours ago

print_unitdates.py - script

- Test your xpaths, loops, and other bits of logic

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

path = 'demo_eads'

for filename in os.listdir(path):
        tree = etree.parse(join(path,filename))
        unitdates = tree.xpath('//unitdate')
        for unitdate in unitdates:
                if unitdate.text is not None:
                        print unitdate.text
```

print_unitdates.py - output

```
1944
1943-1944
1943-1945
1943-1944
1944
1943-1944
1944
1943
1943-1945
1943-1945
July-August 1943
1943
```

non-normalized_unitdate_count.py - script

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

path = 'demo_eads'

total_dates = 0
normalized_dates = 0
non_normalized_dates = 0

for filename in os.listdir(path):
        tree = etree.parse(join(path,filename))
        unitdates = tree.xpath('//unitdate')
        for unitdate in unitdates:
                if unitdate.text is not None:
                        total_dates += 1
                        if not 'normal' in unitdate.attrib:
                                non_normalized_dates += 1
                        else:
                                normalized_dates += 1


print "Total dates:", total_dates
print "Normalized dates:", normalized_dates
print "Non-normalized dates:", non_normalized_dates
```

non-normalized_unitdate_count.py - output

- Get a sense of the scope of the problem

```
Total dates: 14833
Normalized dates: 1
Non-normalized dates: 14832
```

write_all_unitdates.py - script

- Output text from EADs to a slightly more readable format

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

path = 'demo_eads'

for filename in os.listdir(path):
        tree = etree.parse(join(path,filename))
        unitdates = tree.xpath('//unitdate')
        for unitdate in unitdates:
                if unitdate.text is not None:
                        date = unitdate.text
                        date_path = tree.getpath(unitdate)
                        with open('all_unitdates.csv','ab') as csvfile:
                                writer = csv.writer(csvfile)
                                writer.writerow([filename, date, date_path])
        print filename
```

write_all_unitdates.py - output

engstrom.xml	/ead/archdesc/did/unittitle/unitdate[1]	1952-1970
engstrom.xml	/ead/archdesc/did/unittitle/unitdate[2]	
engstrom.xml	/ead/archdesc/dsc/c01[1]/c02/did/unittitle/unitdate	1955-1970
engstrom.xml	/ead/archdesc/dsc/c01[2]/did/unittitle/unitdate	1955-1970
engstrom.xml	/ead/archdesc/dsc/c01[2]/c02[1]/c03[3]/did/unittitle/unitdate	1952-1962
engstrom.xml	/ead/archdesc/dsc/c01[2]/c02[1]/c03[4]/did/unittitle/unitdate	1964
engstrom.xml	/ead/archdesc/dsc/c01[2]/c02[1]/c03[5]/did/unittitle/unitdate	1966
engstrom.xml	/ead/archdesc/dsc/c01[2]/c02[1]/c03[8]/did/unittitle/unitdate	1964
engstrom.xml	/ead/archdesc/dsc/c01[2]/c02[1]/c03[15]/did/unittitle/unitdate	May 1, 1965
mosaicjk.xml	/ead/archdesc/did/unittitle/unitdate	
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[1]/did/unittitle/unitdate	1886-1889
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[2]/did/unittitle/unitdate	1890-1892
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[3]/did/unittitle/unitdate	March 1892-May 1892
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[4]/did/unittitle/unitdate	1899-1905
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[5]/did/unittitle/unitdate	1907-1910
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[6]/did/unittitle/unitdate	1910-1914
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[7]/did/unittitle/unitdate	1914-1917
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[8]/did/unittitle/unitdate	1931-1937
mosaicjk.xml	/ead/archdesc/dsc/c01[1]/c02[9]/did/unittitle/unitdate	1945-1957
mosaicjk.xml	/ead/archdesc/dsc/c01[2]/c02[1]/did/unittitle/unitdate	1986-1962

normalize_unitdates.py

- https://github.com/djpillen/saa15_metadata_power_tools/blob/master/normalize_unitdates.py
- Adds normal attribute for all <unitdate> elements that contain either a year or a range of years
- Normalized about 75% of our unitdates

```
Normalization attempted on 14833 dates
Number of dates normalized: 10919
Number of dates not normalized: 3914
```

write_non-normalized_unitdates.py - script

- Export all non-normalized dates to a csv for manipulation in OpenRefine

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

path = 'demo_eads_normalized'

undated = re.compile(r'^[Uu]ndated$')

for filename in os.listdir(path):
        tree = etree.parse(join(path,filename))
        unitdates = tree.xpath('//unitdate')
        for unitdate in unitdates:
                if unitdate.text is not None:
                        if not 'normal' in unitdate.attrib and not undated.match(unitdate.text):
                                date = unitdate.text
                                date_path = tree.getpath(unitdate)
                                with open('non-normalized_unitdates.csv','ab') as csvfile:
                                        writer = csv.writer(csvfile)
                                        writer.writerow([filename, date_path, date])
        print filename
```

OpenRefine

- http://openrefine.org/
- Powerful tool for working with messy data -- especially spreadsheets

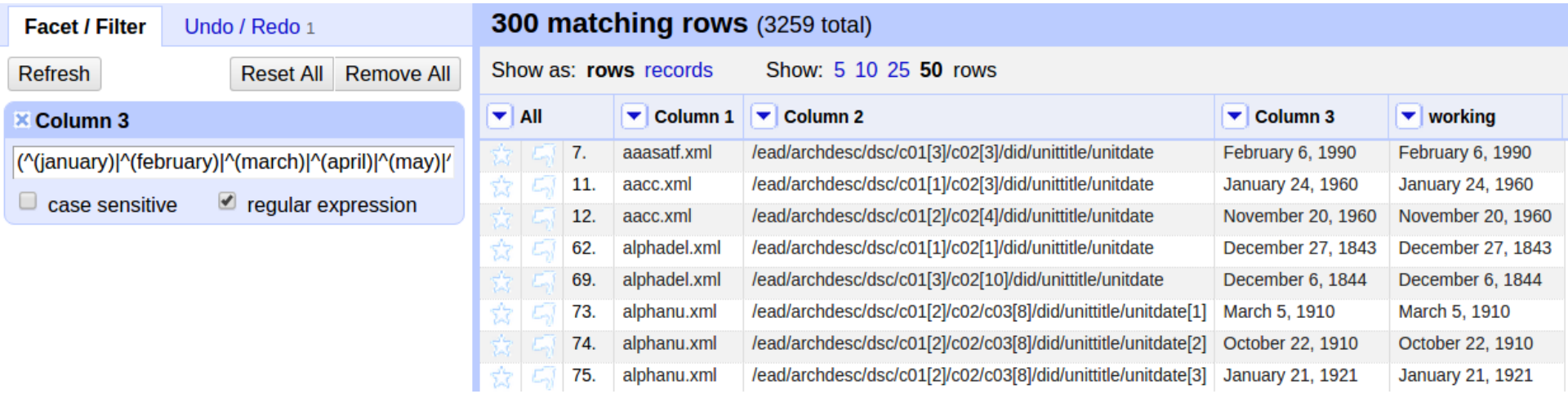
3259 rows

Show as: **rows** records Show: 5 10 25 **50** rows

All		Column 1	Column 2	Column 3
1.	flaherty.xml	/ead/archdesc/did/unittitle/unitdate	1918-1919.
2.	aaasatf.xml	/ead/archdesc/did/unittitle/unitdate	1987-1992
3.	aaasatf.xml	/ead/archdesc/dsc/c01[1]/c02[3]/did/unittitle/unitdate	January-June, 1989
4.	aaasatf.xml	/ead/archdesc/dsc/c01[1]/c02[4]/did/unittitle/unitdate	July-December, 1989
5.	aaasatf.xml	/ead/archdesc/dsc/c01[1]/c02[5]/did/unittitle/unitdate	January-June, 1990
6.	aaasatf.xml	/ead/archdesc/dsc/c01[1]/c02[6]/did/unittitle/unitdate	July-December, 1990
7.	aaasatf.xml	/ead/archdesc/dsc/c01[3]/c02[3]/did/unittitle/unitdate	February 6, 1990
8.	aaasatf.xml	/ead/archdesc/dsc/c01[5]/c02[12]/did/unittitle/unitdate	June 1989
9.	aaasj.xml	/ead/archdesc/did/unittitle/unitdate	1997-2000
10.	aabookfs.xml	/ead/archdesc/did/unittitle/unitdate	2003-2010
11.	aacc.xml	/ead/archdesc/dsc/c01[1]/c02[3]/did/unittitle/unitdate	January 24, 1960
12.	aacc.xml	/ead/archdesc/dsc/c01[2]/c02[4]/did/unittitle/unitdate	November 20, 1960
13.	aachartr.xml	/ead/archdesc/did/unittitle/unitdate	1938-1955
14.	aachartr.xml	/ead/archdesc/dsc/c01[2]/did/unittitle/unitdate[1]	19381954

OpenRefine

- The GREL (Google Refine Expression Language) text for all of the OpenRefine operations demonstrated in this session can be found here
- Step 1: Isolate all dates of the form “Month DD, YYYY”



OpenRefine

- Step 2: Remove all commas
- Step 3: Split into several columns on spaces
- Step 4: Rename resulting columns “mm,” “dd,” “yyyy”

300 matching rows (3259 total)

Show as: **rows** records Show: 5 10 25 **50** rows

All		Column 1	Column 2	Column 3	mm	dd	yyyy
7.	aaasatf.xml	/ead/archdesc/dsc/c01[3]/c02[3]/did/unittitle/unitdate	February 6, 1990	February	6	1990
11.	aacc.xml	/ead/archdesc/dsc/c01[1]/c02[3]/did/unittitle/unitdate	January 24, 1960	January	24	1960
12.	aacc.xml	/ead/archdesc/dsc/c01[2]/c02[4]/did/unittitle/unitdate	November 20, 1960	November	20	1960
62.	alphadel.xml	/ead/archdesc/dsc/c01[1]/c02[1]/did/unittitle/unitdate	December 27, 1843	December	27	1843
69.	alphadel.xml	/ead/archdesc/dsc/c01[3]/c02[10]/did/unittitle/unitdate	December 6, 1844	December	6	1844
73.	alphanu.xml	/ead/archdesc/dsc/c01[2]/c02/c03[8]/did/unittitle/unitdate[1]	March 5, 1910	March	5	1910
74.	alphanu.xml	/ead/archdesc/dsc/c01[2]/c02/c03[8]/did/unittitle/unitdate[2]	October 22, 1910	October	22	1910
75.	alphanu.xml	/ead/archdesc/dsc/c01[2]/c02/c03[8]/did/unittitle/unitdate[3]	January 21, 1921	January	21	1921

OpenRefine

- Step 5: Replace spelled out months with numeric counterparts
- Step 6: Add leading 0s to single digit days

Column 3	mm	dd	yyyy
February 6, 1990	02	06	1990
January 24, 1960	01	24	1960
November 20, 1960	11	20	1960
December 27, 1843	12	27	1843
December 6, 1844	12	06	1844
March 5, 1910	03	05	1910

OpenRefine

- Step 7: Join columns in the proper order -- YYYY-MM-DD
- Step 8: Delete working “mm,” “dd,” and “yyyy” columns
- Step 9: Export the normalized dates as a csv

300 matching rows (3259 total)

Show as: **rows** records Show: 5 10 25 **50** rows

All		Column 1	Column 2	Column 3	normalized
7.	aaasatf.xml	/ead/archdesc/dsc/c01[3]/c02[3]/did/unittitle/unitdate	February 6, 1990	1990-02-06
11.	aacc.xml	/ead/archdesc/dsc/c01[1]/c02[3]/did/unittitle/unitdate	January 24, 1960	1960-01-24
12.	aacc.xml	/ead/archdesc/dsc/c01[2]/c02[4]/did/unittitle/unitdate	November 20, 1960	1960-11-20
62.	alphadel.xml	/ead/archdesc/dsc/c01[1]/c02[1]/did/unittitle/unitdate	December 27, 1843	1843-12-27
69.	alphadel.xml	/ead/archdesc/dsc/c01[3]/c02[10]/did/unittitle/unitdate	December 6, 1844	1844-12-06
73.	alphanu.xml	/ead/archdesc/dsc/c01[2]/c02/c03[8]/did/unittitle/unitdate[1]	March 5, 1910	1910-03-05
74.	alphanu.xml	/ead/archdesc/dsc/c01[2]/c02/c03[8]/did/unittitle/unitdate[2]	October 22, 1910	1910-10-22
75.	alphanu.xml	/ead/archdesc/dsc/c01[2]/c02/c03[8]/did/unittitle/unitdate[3]	January 21, 1921	1921-01-21
236.	amosearl.xml	/ead/archdesc/dsc/c01[1]/c02[2]/did/unittitle/unitdate	June 30, 1919	1919-06-30
312.	beheejhn.xml	/ead/archdesc/dsc/c01/c02[1]/c03[2]/did/unittitle/unitdate	March 23, 1971	1971-03-23

insert_normalized_dates.py - script

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

path = 'demo_eads_normalized'
normalized_csv = 'normalized_dates.csv'
normalized_count = 0

with open(normalized_csv, 'rb') as csvfile:
    reader = csv.reader(csvfile)
    next(reader, None)
    for row in reader:
        filename = row[0]
        print filename
        xpath = row[1]
        normalized = row[3]
        ead_file = open(join(path, filename))
        tree = etree.parse(ead_file)
        unitdate = tree.xpath(xpath)
        unitdate[0].attrib['normal'] = normalized
        outfile = open(join(path, filename), 'w')
        outfile.write(etree.tostring(tree, encoding="utf-8", xml_declaration=True))
        outfile.close()
        normalized_count += 1

print "Normalization based on contents of " + normalized_csv + " complete"
print str(normalized_count) + " dates normalized"
```

insert_normalized_dates.py - output

- Add normal attributes for dates normalized in OpenRefine

```
y2kweb.xml
Normalization based on contents of normalized_dates.csv complete
300 dates normalized
dallas@dallas-ubuntu:saa15_metadata_power_tools$ python non-normalized_unitdate_count.py
Total dates: 14833
Normalized dates: 11220
Non-normalized dates: 3613
dallas@dallas-ubuntu:saa15_metadata_power_tools$
```

Next Steps

- Generate a new csv of non-normalized dates
- Identify a new subset of dates to normalize in OpenRefine
- Continue to chip away at the problem in small, manageable steps