BUSINESS IN THE FRONT, PARTY IN THE BACK

Revising Metadata Processes Up-Front to Benefit Back-End Workflows

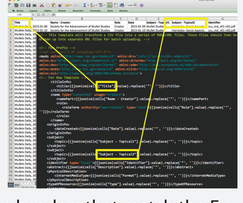
2017 ALCTS Exchange

Scott D. Bacon, Coastal Carolina University,
Kimbel Library - sbacon@coastal.edu

BACKGROUND

When faced with the prospect of manually uploading thousands of collection objects into our digital repository, I knew I needed to create a workflow to automate batch uploading processes. This resulted in a workflow that allows me to take a metadata spreadsheet containing thousands of rows and transform it into a series of MODS XML files contained in one master file, using OpenRefine's templating tool. The csplit command can be used to split the master file up into thousands of fully-formed MODS XML files. Using a Perl script, the files can be batch renamed to match their corresponding digital object files. These matched files can then be uploaded as a zip file into Islandora for easy batch uploading. Each one of the tools used in this process can be modified to enhance the existing workflows of any institution.

EXCEL / CSV



Before beginning the batch transformation process, metadata spreadsheets will need to contain column headers that match the Exporting Template to be used in OpenRefine.

Each column header within the spreadsheet also needs to have a unique name, so if a column's cell contains multiple items they each need to be broken out into separate columns and given unique column header names. As an example, subject headings are most likely going to need to be broken out: Subject - Topical1, Subject - Topical2, etc.

When your columns are revised, revise the exporting template to match your metadata schema, and save your metadata spreadsheet as a CSV (Comma Separated Values) file.

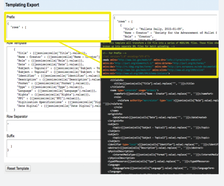
OPENREFINE

OpenRefine is an open source application that enables users to clean up messy data and perform several helpful transformations across multiple metadata spreadsheets.

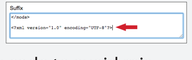
Open the OpenRefine application, which can be downloaded from http://www.openrefine.org. After downloading, click on the OpenRefine icon in your Applications folder, which will open the app using a localhost (127.0.0.1) connection.

Click on the Create Project tab. Import your CSV using the Choose Files button. Click Next. Click on the top right Create Project button to continue. Now you can use facets and filters to clean up your metadata as needed (fix misspellings, change all "&" symbols to "and", format dates as yyyy-mm-dd, fix broken symbols like quotation marks, etc.).

MODS TEMPLATING



Click the "Templating…" option in the Export drop-down at the top right of the OpenRefine interface. Copy and paste the Exporting Template (ccu_md.xml) code from Prefix, Row Template, Row Separator, and Suffix, then click Export.

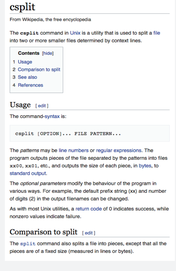


Be sure to delete all lines and white space after the Suffix code to avoid misalignment of the files.

Open the resulting file (ccu_md-csv.txt) and save it with an .xml extension. This file should contain a list of all of your spreadsheet metadata in one file, with fully formed MODS XML files corresponding to each CSV row.

CSPLIT

The csplit command will break the large XML file up into separate files.

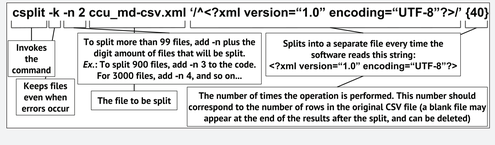


Open Terminal, navigate to the directory where you want the files to be split, and make sure the file you're splitting is also in that directory.

Type this command, which should split the ccu_md-csv.xml file into separate fully formed files:

csplit -k -n 2 ccu_md-csv.xml '/^<?xml version="1.0" encoding="UTF-8"?>/' {40}

Diagram of this csplit command:



Alternatives to csplit include gawk, awk, python, perl, and others.

FILE RENAMING

Batch rename your files using Perl.

```
#!/usr/bin/perl
use XML::LibXML;
use Cwd;
use strict;

my $pwd = getcwd();
# define app directory
my $loc = "/path to file/perl_output";
mkdir $loc unless (-d $loc);
opendir my $dh, $pwd or die "Can't open directory: $!";
while (my $filename = readdir $dh) {
    # skip
    next if $filename eq /^\.{1,2}$/ or $filename !~ /.xml$/;
    open my $in, '<', "$pwd/$filename";
    my $doc = XML::LibXML->load_xml(IO => $in) or die;
    my $identifier;
$doc->findnodes('//*[local-name()=\'identifier\'][@type=\'local\']')->sort(sub{$a->textContent
    [0]->textContent
    $identifier = $1 if /(.*)$/;
    print "$pwd/$local_identifier";
    rename "$pwd/$filename", "$loc/$local_identifier.mds";
}
closedir $dh;
exit;
```

This Perl script renames files by scraping the value from each file's <identifier> element, then loops that command through all of the files in the current working directory.

The renamed files are moved into a new folder.

Create a folder named perl_input containing all of the xml files you want renamed, and put the rename_loop.pl file in that directory too.

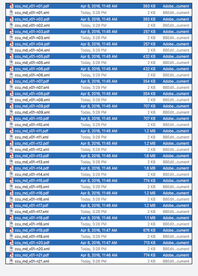
In Terminal, I made sure I was in the perl_input directory, then typed this command:

perl /Users/sbacon/Desktop/perl_input/rename_loop.pl

Hit return, and now the files should be renamed and should appear in a new folder in Desktop named perl_output. Finish any necessary custom batch renaming using Finder, Automator, Windows File Explorer, etc.

ZIP FILE IMPORTER

Zip or compress your metadata files and digital object files using the compression software on your machine or using a third party tool.



The files are now ready to be uploaded to the Islandora repository using the Zip File Importer module.

Github Project Page:
https://gist.github.com/scottdbacon/c81226d20b7e71c6e39fd6d44c95fabe

List of files to be used in this project:
1. ccu_md.xlsx: Test spreadsheet
2. ccu_md.xml: Exporting Template
3. ccu_md-csv.xml: Example of Resulting OpenRefine Transformation (all xml files in one large file)
4. rename_loop.pl: Perl renaming script

Shout outs to Sara Allain, whose MODS template I used as the basis for this project, and Justin Beerley, Library Systems Administrator extraordinaire, who helped me create the Perl renaming script.