Accessioning Born-Digital Records

Instructions for accessioning transfers born-digital records

Create a new ASpace Accession

- Basic use will accession records in a transfer folder (see above)
- Will assume transfer is at: \Lincoln\Library\<Collection Folder>

1. Create a new Accession record in ArchivesSpace
   a. Required:
      i. Content Description
      ii. Dates
      iii. Related Resource

2. Get the accession identifier
   a. If there are two fields, connect them with a dash (-) to get the full ID
   i. "2019" and "036" would be: "2019-036"

   ii. just "2003-17" would just be "2003-17" (for pre-ArchivesSpace accessions)

   b. These are the ArchivesSpace accession id_0 and id_1 fields

Simple Accession

Use when there is no need to create logs for the records creator/donor

This will ingest files while updating the ArchivesSpace accession record.

Accession from the ingest directory

Will assume files are in \Romeo\SPE\ingest\<collectionID>

1. Log on to railsdev server with railsdev
2. Run ingest using an -a flag with the ASpace accession ID (see above):
   ingest apap101 -a 2019-015
3. Script will log to \Romeo\SPE\ingest
Accession Records from another directory

1. Log on to railsdev server with railsdev
2. Run ingest
   a. use an -a flag with the ASpace accession ID (see above)
   b. use a -p flag with the path

   ingest apap101 -p /media/SPE/folder -a 2019-005

   • Must use path accessible to the railsdev server
   • Must convert to Linux path:
     • \Romeo\SPE\folder1\folder2 is /media/SPE/folder1/folder2
     • \Lincoln\Masters\Special Collections\Electronic_Records_Library is /media/Masters/Special Collections/Electronic_Records_Library
   • Script will log to \Romeo\SPE\ingest

Using a Transfer Folder or Accession Template

Use for creating logs in standard transfer folders or template accessions. For other accessions, create the accession manually and use ingest with an -a flag instead.

Overview of Transfer Folder

Requirements:

• Folder starts with collection ID in uppercase, ("UA200", "UA710_Events", "UA608_LACS")
• Located at \Lincoln\Library\<Collection Folder>
• Has three subfolders, "transfer," "log," "request"

Examples:

• UA615_History
  • log
  • request
  • transfer

• UA200
  • log
  • request
  • transfer

Accessioning records using a transfer folder

1. Log on to the railsdev server with: railsdev
2. Run: transfer <collection ID> -a <accession ID>

   Examples
   i. transfer ua200 -a 2019-045
   ii. transfer ua710 -a 2020-003

3. Script will log to \Romeo\SPE\ingest\log

Transfer Records from another directory

• Use the additional -p flag to use another transfer directory
• Must use path accessible to the rails2/processing server
• Must convert to Linux path:
  • \Romeo\SPE\folder1\folder2 is /media/SPE/folder1/folder2
  • \Lincoln\Masters\Special Collections\Electronic_Records_Library is /media/Masters/Special Collections/Electronic_Records_Library
  • May barfs on paths with spaces
• -p flag must point to "transfer" directory in a transfer folder (see above)

• Run transfer using -p flag:

  transfer ua500 -p "/media/Masters/Special Collections/ua500/transfer"

  • Will still log to \Romeo\SPE\ingest

Results of Transfer

1. Files will be packaged unto a SIP bag here: \Lincoln\Masters\Archive\SIP\<collection ID>\<package ID>
a. SIP and AIP packages are here: https://github.com/UAlbanyArchives/packages

2. Processing folder for package is created in \Romeo\SPE\processing\<collection ID>\<package ID>

3. Master files are places in masters subfolder

4. Example Processing package:
   - ua200_d4f5RMWRX9dx2aVkV7erL
   - derivatives
   - masters
   - metadata

Using an Accession template

Accession templates are for scheduled transfers, so the process can be automated

1. Create a spreadsheet accession template in \Lincoln\Library\ESPYderivatives\accessionProfiles
2. Must be named with the collection ID in uppercase
   - UA200.xlsx
   - UA395.xlsx

What is happening

- **transfer** is a function defined in: /etc/profile.d/processingFunctions.sh
- It runs a python script: /opt/lib/ingest-processing-workflow/transferAccession.py
  - relies on /opt/lib/ingest-processing-workflow/ingest.py
  - relies on packages: /opt/lib/ingest-processing-workflow/packages/SIP/...

usage: transferAccession.py [-h] [-p PATH] [-a ACCESSION] ID

positional arguments:
ID                Collection ID for the files you are packaging.

optional arguments:
-h, --help         show this help message and exit
-p PATH, --path PATH  Path of files to ingest.
-a ACCESSION, --accession ACCESSION
                    Optional ArchivesSpace Accession ID for new acquisitions.

- Github repository for ingest-processing-workflow is here: https://github.com/UAlbanyArchives/ingest-processing-workflow
- The script is run under nohup, so it runs in the background

    nohup /opt/lib/ingest-processing-workflow/transferAccessions.py ua200 -a 2019-043 >> /media/SPE/ingest/ua200-transfer.log 2>&1

Uses a pyenv called "ingest" by running: pyenv activate ingest

- List all pyenv with: pyenv versions

Related articles

- Ingesting Digitized Materials
- Updating the website
- Subjects for bulk ingest
- Processing Ingested Digital Files
- asinventory Location Codes