The University of Chicago Library Digital Repository Staging Structure Specifications

Introduction:

A staging structure is the containing directory for an accession while it is being prepared for accessioning into the LDR long term storage location. It is arranged specifically to facilitate two things:

1. The automatic ingestion the files into long term storage
2. Establish each file as an archived object.

Its structure allows for easily organizing files in order to establish provenance and documenting steps taken in order to stabilize and document the files.

Every staging structure should be laid out as such:

- *Staging Root/*
  - *Staging ID/*
    - admin/
      - notes/
      - legal/
      - accession_records/
        - accession_record.json
      - *prefix*1/
        - manifest.txt
        - premis/
          - *originating_data_dir/*file.xls.premis.xml
          - *originating_data_dir/*file.xls.presform.csv.premis.xml
          - *originating_data_dir.xls*.presform.pdf.premis.xml
          - *originating_data_dir/file.txt*.premis.xml
      - *prefix*2/
        - ...
      - *another_prefix*1/
        - ...
      - *another_prefix*2/
        - ...
    - data/
      - *prefix*1/
        - *originating_data_dir/*
          - *originating_data_file.xls*
          - *originating_data_file.xls*.fits.xml
          - *originating_data_file.xls.presform.csv*
          - *originating_data_file.xls.presform.csv*.fits.xml
          - *originating_data_file.xls.presform.pdf*
          - *originating_data_file.xls.presform.pdf*.fits.xml
        - *originating_data_file.txt*
      - *prefix*2/
        - ...
      - *another_prefix*1/
        - ...
      - *another_prefix*2/
        - ...

Note: Portions italicized are relative, and therefore may change. What they are relative to is explained below.

Explanation:

Staging Root: This is the path component up to the location on disk where we will be storing all of our files. This location should be backed up regularly and capable of storing as many accessions as will be being accessioned at a single time while they are being operated on. This component is mandatory.

Staging ID: The temporary ID of a given accession. This string will be replaced with the ARK when the files are moved to the long term storage location. This differentiates one accession from another while files are in the staging root. It is often useful to make this string the accession number, if one exists, or an easily recognizable human readable name for the materials if they do not. This component is mandatory

admin: The subdirectory of the Staging ID directory which will contain all of our administrative metadata. This component is mandatory.

data: The subdirectory of the Staging ID directory which will contain the data moved into the accession from multiple sources. This component is mandatory.

prefix#: These subdirectories of admin and data differentiate different sources of data within the same accession. For example if an accession is composed of 3 CD's and 2 floppy disks each may warrant a prefix, and therefore the contents of admin and data both would be: cd1, cd2, cd3, floppy1, floppy2. One such folder in the data directory is mandatory for each separate location addressing that happens during the staging process. The admin directory must contain a prefix folder named with the same name for every prefix folder in the data directory.

premis: These subdirectories in each of the admin/prefix* directories serve as the root point for the PREMIS metadata files that will be constructed for every file in the data/prefix* directory. This component is mandatory.

accession_records: This subdirectory of admin holds all accession level, rather than file level, administrative metadata. This component is mandatory

accession_record.json: This file contains an accession record which abides by the specifications laid out in (link to accession_record specification here).

originating_data*: This entry is meant to convey that at this point in the file structure of the staging area the directory structure of the origin is preserved, and the containing files copied.

originating_data*.**xml**: These entries represent the PREMIS metadata files which are generated to mirror the contents of the origin directory directory structure in the premis subdirectory. Creating the original PREMIS metadata files is handled by by (link to premis builder here). Tools which operate on the staging directory should amend these files when appropriate utilizing (link to premis library here).

notes: this directory is meant to store short textual notes and support materials. Its presence is mandatory, but its contents is arbitrary.

legal: this directory is meant to store any supporting legal documentation for the accession. Its presence is mandatory, but its contents is arbitrary.

Tools:

Stager: The stager utility automates the creation and population of the staging directory with original data files and corresponding manifest.txt files in their corresponding prefix folders.

Pruner: The pruner utility automates the remove of certain kinds of files which are automatically generated by certain operating systems and common programs. These files are of little archival value, and are therefore removed from the accessions. For more information on the pruner: (link to pruner policy)

Premis Creator: The premis creator automates the creation of PREMIS xml records for each file in the staging environment. For further information on the PREMIS record policy: (link to PREMIS policy)

File Converter: The file converter automates the process for the creation of preservation stable file formats from originating data when possible by wrapping many different tools. For further information on file conversion policies and specifications for naming converted files in the staging directory: (link to File Converter docs)

Technical Metadata Creator: The technical metadata creator automates the process of creating technical metadata for each file in the staging environment. For further information on the process of technical metadata creation and the specifications for technical metadata in the staging environment: (link to tech. metadata project/spec)

Record Creator: The record creator automates the creation and editing of accession_record.json files according to the (link to accession_record.json specification).

Archiver: The archiver automates ingestion of a completed staging directory into the (link to long term storage specification).

Workflow:

1. Ingest materials in reasonable delineations through the use of the Stager utility.
2. Prune the resulting staging environment to remove files which are not of interest via the Pruner utility.
3. Create PREMIS records for all origin files using the premis creator utility
4. Create preservation stable file formats from the origin files when able. Writing events into origin file premis records and spawning new premis records for the newly created preservation stable files. The file converter utility automates this process.
5. Generate technical metadata via the technical metadata creation utility.
6. Create/finalize the accession_record.json file which describes the accession and place it in the staging directory via the AccessionRecorder utility.
7. Ingest the staging directory into long term storage via the Archiver utility.