***AIP Creation Workflow for Hargrett and Russell Library Digital Assets***

***Last updated: 2018-08-01***

\*The Linux tape server should have everything needed for this workflow.

\*The ARCHive is UGA Libraries digital preservation storage system.

1. **Determine that a collection is high priority for ingest into the ARCHive.**

Digital assets must be appraised, have a deed of gift, and be processed before they are ingested into the ARCHive. Lower priority collections can be stored on the Linux RAID disk, SAN (G: Drive), and/or LTO tape.. In most cases, collection assets we are responsible for preserving long-term will be ingested into the ARCHive. Exceptions to this may include digital surrogates of analog materials (created by DLG or in-house).

1. **Decide at what level AIPs will be created (e.g., collection, series, subseries, or folder level).**

Aspects to consider when choosing an appropriate level include the potential size of the AIP, diversity of file formats, and potential need for future format migration or other preservation actions that will require a new version of the AIP to be ingested into the ARCHive.

1. **Confirm fixity of files to be included in AIP has not changed.**

For files that were bagged during accessioning and/or processing:

* Validate bag using “bagit.py” Python script and make note of any errors returned in terminal window.

$ bagit.py --validate /path/to/accession/folder\_bag

For individual files that have not been bagged:

* Use md5sum (Linux), [MD5summer](http://www.md5summer.org/) (Windows), or similar tool to create new checksum values for files.
* Import the new values into a spreadsheet along with the original values stored during accessioning. Note that some accessions may only have [Data Accessioner](http://dataaccessioner.org/) outputs in XML format, while others might have a complete list of checksum values stored in CSV format. [Data Accessioner Metadata Transformer](http://dataaccessioner.org/da-mt.htm) can be used to convert Data Accessioner XML to CSV so the values can be easily imported into a spreadsheet.
* Compare the newly created checksum values with those that were created and stored when the files were originally accessioned to the SAN.
* To compare checksum values in MS Excel, use the “match” formula in a three column spreadsheet. Column A: new checksum, Column B: formula [match (cell, a:a, 0)], Column C: original checksum. If there is a match between the two checksum values, the formula in Column B will tell you which row the match is in; if there is no match, the formula will return “N/A”. Ignore mismatched checksum values for Thumbs.db, DS\_Store, and other temporary system files.

1. **Copy AIP folders to a single directory on the processing workstation, removing temporary files.**

AIP folder titles should include the AIP unique identifier, followed by an underscore, and then the AIP title (e.g. “rbrl-157-er-000008\_Press Releases 2001”). The “aip-id\_AIP Title” naming convention will allow both the AIP ID and the AIP title to be automatically extracted and added to the master.xml file. The AIP folder will then be renamed using a second script to remove the AIP title so that the AIP complies with the naming convention required for ingest into the ARCHive (e.g. “rbrl-157-er-000008\_Press Releases 2001” becomes “rbrl-157-er-000008”).

Use [rsync](https://ss64.com/bash/rsync.html) to copy files to the LTO tape server prior to running AIP scripts. Rsync is command line tool for Mac/Linux that does what Teracopy does for windows machines. Instructions for how to use rsync are available [here](https://docs.google.com/document/d/171nJDt3IAbZOgWXYTqz0XIaT3aGNgcZpKK8zCptXe5s/edit?usp=sharing). Use the “--exclude-from” option and path to “exclude-list.txt” file with rsync command to remove temporary files (e.g. .DS\_Store, Thumbs.db) from the AIP before it is packaged for ingest.

$ rsync -rlptog --exclude-from '/path/to/exclude-list.txt' 'directory/with/files' 'destination/directory'

1. **Determine rights information and apply appropriate statement(s)**

The default rights statement in the the stylesheet used to produce the master.xml file is “in copyright” (<http://rightsstatements.org/vocab/InC/1.0/>). This will need to be manually changed in the stylesheet if another rights statement is needed.

Access restrictions (e.g. needs redaction, attorney-client privilege, etc.) should also be indicated by applying a local rights statement in the master.xml or by using the ARCHive interface during or after ingest. Consider using the interface to apply temporary restrictions since these can be modified at a later date, whereas the rights information recorded in the master.xml file cannot be modified within the file itself after ingest.

Before proceeding, consider organizing the AIPs into subfolders organized by rights situation to make it easier to apply local rights statements for restrictions, redactions, etc. during ingest. Each subfolder can then be processed as a batch when running the scripts below.

1. **Create log file before running scripts (optional)**

For more efficient error tracking and general auditing purposes, you can create a log file that records everything that appears in the terminal window during the AIP creation process. If created, log files should be retained after the ingest process is complete.

* Before running the scripts save the terminal output to a log file by typing the following into the terminal:

$ script /path/to/log.txt

You should see the following message in the terminal:

$ script started, file is /path/to/log.txt

* Run the scripts and anything else you need to do in the terminal.
* When you are finished with the processes you want recorded in the log file, type "exit" into the terminal and hit the “Enter” key. You can continue working in the terminal but the results will not be saved in the log.txt file.

1. **Run “aip-structure\_script.sh" to create the directory structure for the AIP bag.**

For each AIP folder in the working directory, this script will:

* Make an "objects" subfolder
* Move AIP contents to the objects folder
* Make a "metadata" subfolder

aip-structure\_script.sh /path/to/aipss/directory

1. **Add additional files to the metadata folders, other than FITS and master.xml, which are created by the script and automatically added to the metadata folder (optional)**

You can use the command line to add the same file to multiple AIPs (e.g., to add a deed of gift document to each metadata folder for all the AIPs for a certain collection). From the directory with the AIPs where “coll-number\_dg.txt” is the file to copy and metadata is the folder to copy it to :

$ find . -maxdepth 2 -type d -name "metadata" -exec cp coll-number\_dg.txt {} \;

All metadata files should have a code in the filename to reflect what kind of metadata it is (e.g. “\_da” for DataAccessioner output; “\_dg” for Deed of Gift). Any file type can be added, but only DOC/DOCX, PDF, TXT, and XML metadata files will be keyword searchable in the system. The controlled vocabulary for metadata codes is available [here](https://drive.google.com/open?id=1vC7lJcoJKTRIRao24A4tbX__I2IgcxYmdBQuXg6Twvk).

1. **Run “aip-finish\_script.sh” to run** [**File Information Tool Set (FITS)**](https://projects.iq.harvard.edu/fits/fits-xml)**, create the master.xml file, bag and tar/zip the AIP, and generate md5 manifest.**

For each AIP folder in the working directory, this script will:

* Make a subdirectory for the FITS XML in the AIP folder
* Run FITS and save the FITS XML output for each file to that subdirectory
* Run stylesheets against the XML files in the FITS subdirectory to combine the FITS XML files into a single FITS XML file and make the master.xml file from the combined FITS XML file
  + Dependencies: open.xml, close.xml, saxon9he, fits-cleanup.xsl, fits-to-master\_singlefile.xsl, fits-to-master\_multifile.xsl
* Validate the master.xml against custom schemas
  + Dependencies: xmllint, master.xsd, premis.xsd, dc.xsd
* Save the master.xml in the metadata folder AND outside of the working directory
* Move the contents of the FITS subdirectory to the metadata folder
* Move the combined FITS XML file outside the AIP folder to store with collection documentation
* Rename the AIP folder from “aip-id\_AIP title” to “aip-id”
* Bag the AIP and validate bag (using bagit.py)
* Rename the AIP folder to “aip-id\_bag”
* Tar and zip the AIP (using “prepare\_bag” Perl script, [tar](https://ss64.com/bash/tar.html), and [bzip2](https://ss64.com/bash/bzip2.html))
* Generate a manifest (using md5sum) containing md5 checksum values for each tar/zip folder. This manifest will be uploaded to the ARCHive along with the AIPs.

When running the script, be sure to include as an argument the name of the department the AIPs belong to (e.g. “hargrett” or “russell”)

$ aip-finish\_script.sh /path/to/aips/directory department

1. **Perform quality control on AIP batch as needed**

Given the variety of content processed using the AIP workflow, there will sometimes be files and metadata with characteristics that were not accounted for in the scripts or stylesheets. For this reason, a sample of AIPs should be checked prior to ingest into the ARCHive. The AIP structure and master.xml files are validated upon ingest to the ARCHive. Staff perform this quality control process to catch errors prior to ingest for two reasons: ingest can be time consuming and, more importantly, there are some errors that could result in valid but inaccurate AIPs. These must be caught with a manual quality control process to avoid ingesting lower-quality AIPs into the ARCHive.

* See [batch-qc-documentation](https://docs.google.com/document/d/1WacAmuiiz8lst92E3qy0xFAWF__Hml5kajfHJiKlulQ/edit?usp=sharing) for detailed instructions for how to perform quality control on AIPs and master.xml files using spot checking and stylesheets.

1. **Copy AIPs to be ingested the ingest server folder**

* Connect to the base-uri/hargrett (for Hargrett Library) or the base-uri/russell (for Russell Library) server.
* Enter your the login credentials for the ingest server.
* Copy the tar/zipped bags to the appropriate folder on the ingest server.

$ rsync -r --progress 'directory/with/files' 'destination/directory'

Note that it is not necessary to use rsync (as opposed to drag-and-drop using the file manager gui) when copying the bags to the server because the fixity of the files will be verified during the ingest process using the md5 values recorded in the manifest.

* After the files finish copying to the ingest server, they are ready to be accessed using the ARCHive web interface. Uploading onto the ingest server is different than finishing an ingest into ARCHive. The server is the processing space from which the ARCHive system accesses the files as it validates them and prepares them for writing onto the disk server and tape storage. After successful ingest, files are deleted from the ingest server.

1. **Schedule job and ingest AIPS into the ARCHive using the web interface**

Log into the website using your individual credentials.

Note that in order for the master.xml files to validate, there must already be corresponding Collection and Rights objects in the ARCHive. New collections and rights statements can be added using the web interface prior to ingest.

To add a new collection to the ARCHive (if necessary):

* Click on the “Colls” tab in the bar at the top of the page.
* Click “New Collection” in the upper right hand corner of the page.
  + Groups: The repository (e.g. “Russell” or “Hargrett”) is automatically selected based on user’s login credentials.
  + Object identifier type: This is automatically filled in as “base-uri/hargrett” or “base-uri/russell”
  + Object identifier value: Enter the collection ID using lowercase (e.g. rbrl-057).
  + Title: Enter the full name of the collection (e.g. Democratic Party of Georgia Records).
  + Repositories: Do not select anything.This would only apply to materials that belong to one library but are being processed by another.
  + Rights statements: Do not apply rights statements to collections. This will only be done at the AIP level.

To add a new local rights statement to the ARCHive (if necessary):

* Click on the “Rights” tab in the bar at the top of the page.
* Click on “New Rights Statement” above the list of current rights statements.
  + Groups: The repository (e.g. “Russell” or “Hargrett”) is automatically selected based on user’s login credentials.
  + Uri: Append appropriate terms to the URI prefix to represent the rights statement (e.g. “base-uri/russell/attorneyclient”).
  + Title: Enter the full name of the local rights statement (e.g. “Attorney-Client Privilege Restriction”).
  + Text: Enter a brief description of the rights statement (e.g. “Contains attorney-client privileged records that are restricted for 75 years after the date of creation.”).

Check master.xml files for errors

To reduce the chance of receiving error messages during ingest, you can use the ARCHive interface to check your master.xml for errors before starting the ingest process.

* Click on the “Check master.xml for errors” link in the top left part of the homepage, above “Finished jobs.”
* Click “Choose file” and select the master.xml file you want to test from the master-xml folder in the AIP directory.
* “Your master.xml is **valid**” will display if there are no errors.
* “Your master.xml is **invalid**” will display if there are errors, with a description of the errors listed below it in red.
* It is recommended to test at least one master.xml file per batch, so as to screen for errors before beginning the long ingest process. If there are no errors, you are ready to begin a new ingest job.

1. **Save a copy of all master.xml and combined FITS xml files on the SAN for ongoing collection management purposes**

File formats and other information recorded in the master.xml and FITS xml files will be used to inform future digital curation activities such as format migration for preservation.

**Authors**

Steve Armour, University Archivist, Hargrett Rare Book and Manuscript Library

Brandon Pieczko, Processing and Digital Archivist, Richard B. Russell Library for Political Research and Studies

**Acknowledgements**

This workflow was developed and tested with the assistance of Adriane Hanson, Head of Digital Stewardship.

The aip-finish script incorporates a script to tar and zip the files which was developed by Shawn Kiewel, UGA Libraries Application Analyst