

<https://drive.google.com/file/d/1CC7udUgx43o69sQ441cbKQ438qGdspEj/view?usp=sharing>

# **Curation Workflow Diagram steps with script execution:**

**Note: Before running the scripts described in this document, please make sure that you have the following environments set up:**

1. **Set up the coding environment following the curator instructions to run VTDR curation workflow scripts using ScriptsSetupAndExecution\_CurationWorkflow\_Windows document for windows installation or ScriptsSetupAndExecution\_CurationWorkflow\_Mac document for Mac installation**
2. **Download DART app and set up the workflow for bag creation and APTrust deposit on DART + Download WebDrive/Cyberduck for VT libraries S3 bucket deposit using DARTSetup\_APTrust\_VTLibS3Setup\_CurationWorkflow document**

In the steps below, Python (.py) scripts are referenced from their relative path on your computer, e.g. C:\Users\username\anaconda3\envs\curation

1. **Request to publish dataset received from Client:**

Request to publish dataset received from Client:

When a client requests publication, it is typically received either through

a) an email notification from our figshare for institutions instance with the title “Item to review” or

b) through an email sent by the client to the [Virginia Tech Data Repository](https://data.lib.vt.edu/) (VTDR) with the dataset as an attachment requesting [VTDR](https://data.lib.vt.edu/) to publish it.

When the dataset is received as an attachment in an email, the curator can request the client to access their [VTDR](https://data.lib.vt.edu/) account and deposit the dataset. This creates a transfer of the data from the client to the curator with file integrity (e.g. checksums).

1. **‘Ingest record created’ and**
2. **‘Ingest dataset bagged and deposited’:**

**Creation, bagging and deposition for articles in review:**

**Part 1: Get the article ID:**

Go to [VTDR](https://data.lib.vt.edu/), click on the name initials on the top right corner displayed in a circle. Click “Review requests”, select the article whose ingest record needs to be created. Click on the red “Cite” button, copy the number at the end of the doi hyperlink and paste it in generate\_config.py under the folder ‘curation/VTechDataRepo’ in Visual Studio Code . For example: if the ‘DataCite’ appears as: Test (2022): Testing dataset. <https://doi.org/10.7294/20222222>, then the figshare article ID is 20222222.

**Part 2: Fill in the information in the spreadsheet:**

Open the 20211214\_VTDR\_PublishedDatasets\_Log\_V7 spreadsheet, go to the “Ingest” sheet. Enter the information for the article in review by assigning it the next consecutive number under column A Ingest#. For e.g. if the last entry in column A is I00122 then the current review item entry will be I00123. Other columns include:

**Requestor:** Requestor is the person requesting publication of the dataset, this is the client associated with the article in review’s [VTDR](https://data.lib.vt.edu/) account. This can be found under “Request details” on the right-hand corner for the item in review. For e.g. “by xyz ([xyz@vt.edu](mailto:xyz@vt.edu)) ->access this account”.

**Corresponding Author/Corresponding Author Email address**: metadata fields found in the [VTDR](https://data.lib.vt.edu/) item review page.

*Note:* When the Corresponding Author Email address field is empty (e.g. because a graduate student does not want to submit this information), it can be inferred for input into the spreadsheet. In most publication requests, the requestor and the corresponding author are the same clients, but in some cases this is not true. Where the first name in the “Authors” list field in the [VTDR](https://data.lib.vt.edu/) review page of the dataset does not match the requestor’s name, for example, the requestor is the client requesting publication, and the corresponding author is the first name in the “Authors” list field.

**Version #:** Since ingest record is created before any changes are made to the article, this is almost always a first version (01).

**Date Ingested:** The date at which ingest record is submitted in YYYYMMDD format.

**Dataset Title:** The title that appears on the article in review.

**Comments**: Comment’s cell must be filled and cannot be left empty in order for the scripts to run. While running the scripts to ingest and upload to APTrust please enter some comment in the ‘Comments’ column. Changes can be made to these comments after review. Examples of comments: ‘folders/files open up using excel’ or ‘text files open up fine’ or ‘large files open up with notepad++’ or ‘scripts open up with visual studio code’ or ‘netcdf files open up with idl and look file’ or ‘license is attached to scripts’ or ‘author received DOI, publishing process delayed until paper is accepted’ or ‘author embargoed the item until YYYYMMDD’ etc. Please go through the comments for previous articles in this column to get a better idea of the comments.

**Figshare Article ID:** 8 digit number found at the end of the citation, found by clicking “Cite” on the item review/published page.

**DOI suffix:** This number is the same number as the figshare article ID for all new data publications. VT Libraries used a different repository before moving to figshare, so some of the older publications in the 20211214\_VTDR\_PublishedDatasets\_Log\_V7 spreadsheet have a DOI suffix that is not the same as the article IDs that figshare assigned to them. The older publications not initially hosted by [VTDR](https://data.lib.vt.edu/) but migrated to [VTDR](https://data.lib.vt.edu/) have combinations of letters and numbers in their DOI suffix as opposed to numbers only .

**Part 3: Enter configurations:**

Open generate\_config.py located at your curation environment directory, e.g. “C:\Users\padma\anaconda3\envs\curation\VTechDataRepo”. If this script cannot be found then open save generate\_config\_example.py and save it as generate\_config.py. Fill in the “FigshareArticleID” tag. Also fill in all the other tags: PubVerNum, VTDRToken, CurName, NonDissContentDir, DartExePath, ReadmeDir including DART executable and directory paths under different sections in this script. For more details on configurations, refer to “Setting up configurations on your local computer (section 5) of the document: ScriptsSetupAndExecution\_CurationWorkflow\_Windows/ScriptsSetupAndExecution\_CurationWorkflow\_Mac. Run generate\_config.py to create configurations.ini in the curation folder.

**Part 4: Run the script/scripts:**

Open IngFolder\_Download\_TransferBagAPTrust.py located at your Figshare-APTrust directory e.g. “C:\Users\padma\anaconda3\envs\curation\VTechDataRepo\Figshare-APTrust” and run this script.

For testing purposes, pick “1” to deposit the bag to APTrust-Demo: Depositing bags to APTrust-Demo is for testing purposes only, bags deposited here will be deleted after a certain time period.

For archiving purposes: enter “2” after running this script, in order to deposit the bag to APTrust-Repo and VT library S3. This creates the ingest bag for the article ID under review and deposits the bag created to APTrust-Repo. Other repository upload options can be found in part 2 of the section “Running the VTDR Workflow scripts” (section 8-2) in the ScriptsSetupAndExecution\_CurationWorkflow\_Windows/ScriptsSetupAndExecution\_CurationWorkflow\_Macdocument. The bag deposited can be also extracted and verified following the instructions in “Checking the bags created by DART and bag validation” (section 9) of ScriptsSetupAndExecution\_CurationWorkflow\_Windows/ScriptsSetupAndExecution\_CurationWorkflow\_Macdocument.

Some possible errors that can be encountered are also discussed in the section “Possible errors” (section 12) in the document above.

1. **: Provenance log and client record created:**

**Filling in the provenance log:**

Open [Provenance Log Template](https://docs.google.com/document/d/1AOQsr0GP00mp3ZfpeW9oJE-wLY48oKWYZ_NR2cLz8wI/edit) and start filling in the initial interaction details and save it as a new file on your local computer named ‘Provenance.rtf’. The curator is expected to update the interaction details with the client in the provenance log from time to time until the publication process is complete. When completed this provenance log should describe all the steps taken to enhance and modify the dataset between its ingest and publication. Examples of filled in provenance logs can be found in the folder “Non Disseminated Content” under “Bags” in the VT curation services google drive archive.

**Creating client record:**

Go to [LibCRM](https://vt.libapps.com/libapps/login.php?site_id=17108&target=/), a VT library portal to record interactions with the client. Click the blue “People” tag found under “At a glance” box, this will open the “People Profiles” page, under “Keyword” type the client’s name. If the client appears under “Active” then follow instructions under 1 below, otherwise follow instructions under 2. If the client is a ‘library employee’ then follow instructions under 3 :

**1. When client profile already exists in** [**LibCRM**](https://vt.libapps.com/libapps/login.php?site_id=17108&target=/)**:**

Click the client’s name, under the client’s page click “Interactions” tab, select “+INTERACTION”, fill in the fields from the information in 20211214\_VTDR\_PublishedDatasets\_Log\_V7 spreadsheet and their [VTDR](https://data.lib.vt.edu/) account review page. Following is an example of a test client creation who is **not** a library employee, for library employees requesting publication follow the instructions under 3. Fields might be filled in as follows:

**Section1:** **General Info:**

**Title:** I00189: Susanna Wright’s publication request

(Note: I00189 is the Ingest # obtained from the “Ingest” sheet of 20211214\_VTDR\_PublishedDatasets\_Log\_V7 spreadsheet, after publication the title will be changed to P00123: Susanna Wright’s publication request where P00123 is the publication accession# obtained from the ‘Published’ sheet of the spreadsheet, this can be done in step 16)

**Source:** Email

**Profile:** Susanna Wright, Andrew Jones, lib-Data Services (Note: Susanna Wright is the requestor and Andrew Jones is the corresponding author, often however these two names are the same, the “lib-Data services” option should be picked for all non-library employees to connect the interaction to Data Services)

**Record Owner:** Select curators involved in the review from the drop down menu

**Type:** 2. Consultation

**Date:** Current date

**Section 2:** **Fields for all types:**

**Number of participants:** enter the number of curators exchanging interactions with the client/involved in the article review process, this could be any number.

**Section 3: Additional Info:**

**Location:** Off-campus

**Duration:** 180 (this is an estimated time that might be spent reviewing the article, this time will be updated in step 16 after publication process is complete)

**Section 4: Attachments:**

Drag and drop the provenance log and email interactions from their saved locations. The email interactions should be saved as a pdf file, go to the vt gmail interaction with the client and click the print icon in the top right corner, select the option “Save as pdf” from the drop-down print menu options, and save it as Email\_Correspondence or Email\_CorrespondenceX where X is 1,2, 3 etc. if there are multiple client-curator email threads exchanged with the client. Updated provenance log and email interactions will replace the initial attachments upon completion of the publishing process in step 16.

**2. When client profile is not found in LibCRM:**

Click “+Person” on the top right corner of the page under the “Records” tab. Enter the fields here, following is an example for a test profile:

**First Name:** Susan

**Last Name**: Werth

**Email**: [xyz@vt.edu](mailto:xyz@vt.edu)

**Record Owner:** Padma

**Email Subscriptions:** Unsubscribed

**Type:** Student

Click “Create”, after the client’s profile is created, go to the “Interactions” tab and click “+ INTERACTION”, fill in the fields for sections 1, 2 and 3 as described in 1 above.

**3. Client interaction for Library employees:**

Library employees should not be on the profile side but they should be recorded on the record owner side. So, if a library employee requests a publication with VTDR, click “Interactions” (instead of profile) under “Dashboard”, click “+Interaction” and fill in the fields, following is an example for a test library employee Matthew Dier who works in Data Management and Curation Services:

**Section 1: General Info:**

**Title:** I00144/P00123: Matthews publication request

**Source:** Email

**Profile:** lib- Data Management and Curation Services (pick this option if library person is the only client/person requesting publication and the library person belongs to this group, note that the library employee name should not be added under profile)

**Record Owner:** Matthew Dier (Note that this is the library employee requesting publication, and thereby should appear on the record owner side)

**Type:** Consultation

**Date:** 09/15/22

**Section 2: Fields for all types:**

**Check if internal to library:** Check this box if client is a library personnel

Fill in the other fields in section 2 and 3 as described under 1.

1. **Dataset metadata on repository platform evaluated for quality/completeness:**

After the article in review is created, bagged and uploaded to APTrust, begin the review process. Go to [VTDR](https://data.lib.vt.edu/), click on the name initials on the top right corner displayed in a circle. Click “Review requests”, select the article that needs to be reviewed. In the right column, under “Review”, click “Assign request”, select reviewer name and click “Save”. This will make the “Edit” tab accessible to the assigned curator. Click the “Edit” tab and review the metadata. In addition to reviewing the metadata, the dataset downloaded also needs to be reviewed. Ingested article can be reviewed by opening the ingest folder created in the “curation” folder (steps 2 and 3: part 4), and look for the ingest folder e.g. VTDR\_I00238\_RamseyerC\_RamseyerC\_v01\_20220907. Reviewing the dataset could include: checking to see if the files submitted open up properly, contents in the worksheets submitted are scanned, naming in the spreadsheet reflects the information in it, codes/websites in the dataset are checked to see if they open up, checking to see if license is provided with the scripts (either copied at the top of the script or provided as a license.txt file) etc. Talk with other Curation Services personnel about their actions towards this step.

**Step 5a: Consider potential sensitivity of dataset, communicate with client about this as needed:**

There are several kinds of data that can be considered sensitive and in need of additional scrutiny before publishing. For example, If the dataset submitted involves human subjects or survey results then it could be considered as potentially sensitive data. Such a dataset might require checking for possible re-identification risks of subjects involved in the dataset (for e.g. if survey results are included in the dataset). Issues related to research conducted with human subjects also need to be considered before moving forward with the publication process of such a dataset. Please interact with the appropriate personnel regarding the publishing of a sensitive dataset, starting with colleagues in Curation Services. After clearance of re-identification risks or other risks that might be involved, interact with the client either to modify the dataset to reduce the risks associated with it or to improve the dataset as needed.

Other datasets with sensitivity concerns are those originating from proprietary or national security research, data involving endangered species or animal subjects, and research involving biohazards. Work with other Curation Services personnel in assessing and addressing these concerns.

1. **Does metadata meet Publishing Requirements? Dataset of sufficient quality to publish?**

**Coming from Step 5**: If the dataset contains metadata that meaningfully meets the VTDR Publishing Requirements, and the dataset evaluation in step 5 does not lead to recommendations for data improvement that necessitate contact with the data depositor, then proceed to step 11 and begin publication of the dataset. See <https://guides.lib.vt.edu/VirginiaTechDataRepository/PreparingDataForDeposit> under “Documenting Your Data” for the VTDR Publication Requirements.

However, if further publication requirements need to be met by the client, e.g. the client needs to modify the dataset or update the sheets etc. then decline the dataset and follow the steps 7-10. If the curator is implementing these changes then the article need not be declined.

**Declining the dataset:** Go to the review tab on the right column of the article in review and click “Decline and return”, this will prompt a window asking for a reason for declining the dataset, here the curator can enter the reason for declining the dataset for e.g. “client agreed to combine individual items and submit them under one item” or “client agreed to remove column B in the spreadsheet to reduce risks involved in re-identification and submit the dataset again”.

**Coming from Step 11**: If the author approves the suggestions/improvements made after the curator review process and the dataset meets the minimum publishing requirements, then proceed to step 12 to implement these suggestions followed by the publication of the dataset.

1. **Communicate with client to get minimum metadata and suggest other dataset sharing improvements:**

After reviewing the metadata under “Edit” tab on VTDR review page, and the ingest dataset under ‘curation’ folder, contact the client to check/enhance the metadata as needed. Recommendations for enhancement follow from the evaluation done in Steps 5 and 5a.   
  
Here is an example of a client interaction:

“*Dear xyz,*

*Thanks for submitting your dataset to VTDR. We reviewed your dataset titled “test1” and have the following suggestions for your figshare metadata:*

*-Please change Item type to “dataset” instead of “software”*

*-Please add a location to your metadata since your measurements were made at location x*

*Following are the suggestions for the* ***files in your dataset****:*

*-Please include an open source license in your scripts. We recommend* [*open source licenses like MIT*](https://choosealicense.com/licenses/mit/)*. You could copy the license text and paste it to the very beginning of your codes after replacing [year] with the current year and [fullname] with your name or include the text file in your “scripts" folder.*

*-Please label the different sheets in your worksheet with meaningful names instead of “sheet 1”, “sheet 2” etc.*

*Please let us know if you approve these suggestions or provide modifications. We will create a README.rtf based on your metadata and upload it to your account after we hear back from you. Please let us know if you have any questions” or*

*“Attached is a README.rtf we created based on your metadata, please let us know if this looks good or suggest modifications. Please let us know if you have any questions.”*

To get a better understanding of suggestions in the review process and the interaction with the client, please read the provenance logs and email interactions associated with the published articles. These can be found at: CurationServicesGoogleDriveArchive->BAGS->NonDisseminatedContent, each folder here contains a provenance log and email interaction associated with a published article whose DOI is displayed in the folder name.

1. **Record communications in client record:**

Update the provenance log started in step 4, along with the date and interaction details, to the client record management system (currently [**LibCRM**](https://vt.libapps.com/libapps/login.php?site_id=17108&target=/)). The interaction details could include information from the email interactions, zoom, slack or phone interactions, in-person interactions etc. Upload the updated provenance log and any additional email interactions (replace them with the original attachments uploaded in step 4) under “section 4: Attachments” (in the interaction page) as described in step 4.

1. **Modify metadata and files in repository platform with agreement of client:**

After further interaction with the client, modify/implement agreed upon suggestions to the metadata (e.g. adding certain keywords to client’s metadata or changing the group selected etc.) and/or the dataset (e.g. unzipping client’s dataset to append license.txt and zipping it back up again, uploading an updated file sent by the client etc.) on the repository platform. This can be done by impersonating the client’s record described below:

**Impersonating client’s record**: One way to access/impersonate a client’s account is by clicking “access this account” under “Request details” in the article review page (on the right-hand column). Another way to do this is by clicking the name initials icon on the top right corner of [VTDR](https://data.lib.vt.edu/) and selecting “Administration” from the drop-down menu. In the “Admin” page, select “Users” and type the user name, click on the gear icon->access this account. When you are in the client’s account, a red box appears on the top of the page displaying “You are currently viewing the account of xyz”. Scroll on the item that needs modifications and click on the little pen icon that appears at the end of the review article listed here. This opens up the item page, here the dataset can be modified by clicking “manage” and/or metadata can be changed upon author’s approval agreement. After modifications are done, click “Save changes”, followed by “No, save privately” if the article was not declined in step 6 or “Yes, publish” if the article was declined in step 6.

1. **Record modifications in Provenance log:**

Update the provenance log with changes to the content to publish (i.e. metadata and data files). These changes should be derived from email exchanges,zoom/phone interactions, or slack interactions with the depositor and/or corresponding author. Be as explicit as possible in describing these changes in the provenance log; the intent is to have a clear record of changes from ingest of the dataset to its publication.

1. **Add metadata to dataset on repository platform and Step(11a): Record metadata changes in provenance log:**

Go to the item review page in [VTDR](https://data.lib.vt.edu/), and under the “Edit” tab, implement any additional metadata suggestions and click save, metadata changes can also be implemented by impersonating the client’s profile as described in step 9. Create a README file based (README.rtf) based on the updated metadata in the repository system as follows:

**Creating README.rtf:**

-Enter the article id whose README.rtf needs to be created in generate\_config.py (this was already entered in step 2 part 1). The “ReadmeDir” tag in generate\_config corresponds to the path where the current README.rtf file will be created.

-Run the script generate\_config.py. Open and run the script: AutomatedREADMErtf.py located at C:/Users/username/anaconda3/envs/curation/VTechDataRepo/Figshare-APTrust under the current generate\_config\_example.py settings cloned from github repo.

-A README.rtf will be created at the “ReadmeDir” path provided above

(More details on creating a README file can also be found in the section “Running the VTDR Workflow scripts-1. Creating README.rtf” (section 8-1) of the document: ScriptsSetupAndExecution\_CurationWorkflow\_Windows/ ScriptsSetupAndExecution\_CurationWorkflow\_Mac

Note: If the corresponding author leaves the email address option as blank in the metadata field then leave it as blank in the automated README.rtf file

Impersonate the client’s account as described in step 9. In the client’s profile, click “Manage” and upload the README.rtf to the client’s account by dragging it or click browse and add it from the saved location. Add/delete other files as agreed upon with the client.

Record any changes to the dataset, including the addition of the README file, in the provenance log.

1. **Publish dataset on repository platform and step (12a) Record dataset publication in provenance log:**

After all the agreed upon changes are implemented, publish the dataset by going to the “Review” tab on [VTDR](https://data.lib.vt.edu/) article review page and picking “Approve and publish”. Record publication of the dataset to the provenance log.

1. **Inform client dataset is published, send citation and DOI:**

Click “Browse” in the [VTDR](https://data.lib.vt.edu/) page and select the recently published item (click refresh if the item published does not appear or impersonate the client's account and select the published article if it can’t be seen under “browse”). Click “Cite”, click “Copy citation” and “Copy DOI”, and paste it in the template email below. Below is a template email for sending citation and DOI to client (author instructions in parentheses):

“*Dear xyz,*

*We published your dataset titled “test 1 2 and 3”.*

*Here is the citation:*

*(Paste citation after removing .v1 or .v2 or .vx at the end)*

*And, here is the DOI:*

*(Paste DOI after removing .v1 or .v2 or .vx at the end)*

*Please let us know your resource title and DOI after your manuscript is accepted(Note: this option only applies to manuscripts under review). Please let us know if you have any questions.*

*Thanks,*

*Curator signature “*

**Note:** It is important to remove the version (for e.g. .v1 or .v2 etc.) at the end of the citation and DOI before sending it to the client. This minimizes the potential of the inclusion of a previous not updated version as reflected by the DOI being linked from other digital objects. The base DOI (DOI without .v1 or .v2 etc. ) always points to the latest version. If updates were to be made on a published dataset and the dataset is published again as the next updated version, then the base DOI points to the latest version.

1. **Publication record created**
2. **Publication dataset aggregated bagged, and deposited:**

**Part 1: Get the article ID:**

Go to [VTDR](https://data.lib.vt.edu/), click on “Browse” on the top left corner next to the search box. Click on the article that was published. Click on the red “Cite” button, copy the number at the end of the doi hyperlink before the version (e.g. before v1 or v2 etc.). For example: if the ‘DataCite’ appears as: Test (2022): Testing dataset. <https://doi.org/10.7294/20222222>.v1, then the figshare article ID is 20222222 and version is 01

**Part 2: Fill in the information in the spreadsheet:**

Open the 20211214\_VTDR\_PublishedDatasets\_Log\_V7 spreadsheet, go to the “Published” sheet. Create a log for the published article by assigning it the next consecutive number under column B Accession#(P). For e.g. if the last entry in column B is P00122 then the current published item entry will be P00123.

**Ingest #:** copy the ingest number that this published item corresponds to from the “Ingest” sheet.

**Requestor:** Requestor corresponds to the person requesting to publish the dataset, this is the person who is associated with the figshare account used to send the item for review. This can be found under “Request details” on the right-hand corner for the item in review. For e.g. “by xyz ([xyz@vt.edu](mailto:xyz@vt.edu)) ->access this account”.

**Corresponding Author/Corresponding Author Email address:** these are found in the metadata fields. If the corresponding author leaves the email address option as blank in the metadata field then obtain the email address from the email interactions. This email address will be stored in the ArchivalPackageREADME.rtf file for administrative use only; if it was left blank in the repository system this indicates the e-mail address is not to be publicly shared.

**Version #:** This is found at the end of “Cite”

**Date Published:** Published date has to match the date published on figshare, but in YYYYMMDD format. The published date is not to be confused with the date that the published article was created and bagged since this could be different from the date of the published article. The published date can be found under the red cite button on the published article, here it appears as “Dataset posted on day.month.year”

**DOI:** This is found under “Cite”, please enter this in the same format as it appears in other rows for this column.

**Title:** This appears in the published article on figshare

**College, Department**: These could be found by doing a google search, college/department usually appear on the virginia tech faculty/staff page of the requestor on the top left corner of their VT webpage, or with the use of [Virginia Tech Person Search](https://search.vt.edu/search/people.html).

**Date of most recent comment:** This corresponds to the date when the last interaction was made with the author regarding their published dataset, this date is usually set to the published date.

**Most recent comment:** The comment here is usually “Publishing workflow has been completed”, any additions can also be entered, for e.g.: “waiting on author to submit resource title and doi” or “waiting on author to update his dataset periodically” etc.

**Figshare Article ID:** Found under “Cite” button

**Updated Email interaction to** [**LibCRM**](https://vt.libapps.com/libapps/login.php?site_id=17108&target=/)**:** After creating/updating existing interaction on the client’s profile in [LibCRM](https://vt.libapps.com/libapps/login.php?site_id=17108&target=/), enter “done by xyz” where xyz is the curator’s name. However, this update will be done in step 16 below, so this field will be left blank until then.

**Part 3: Enter configurations:**

Open generate\_config.py and fill in the “FigshareArticleID” tag. Run generate\_config.py to create configurations.ini in the curation folder.

**Part 4: Run the script/scripts:**

* Open and run the script PubFolder\_Download.py located in ‘C:\Users\username\anaconda3\envs\curation\VTechDataRepo/Figshare-APTrust’ (as per the github setup instructions), this script downloads the published dataset and creates a publication folder in the “Curation” folder(or the path entered in “CurationDir” in generate\_config)
* Open the publication folder created, this folder will have the following naming convention : VTDR\_P00XYZ\_I00XYZ\_DOI\_XYZ\_lastnamefirstinitial\_v0X\_YYYYMMDD.
* Add the completed provenance log and all the email interactions to the “VTCurationServicesActions” folder. Name these as ProvenanceLog.rtf and Email Correspondence (or Email\_Correspondence1, Email\_Correspondence2 etc. in case of multiple email threads). (VTCurationServicesActions folder found at the path: C:\Users\username\anaconda3\envs\curation\VTDR\_P00XYZ\_I00XYZ\_DOI\_XYZ\_lastnamefirstinitial\_v0X\_YYYYMMDD\VTCurationServicesActions)
* Open PubBagDART\_TransferBagAPTrust.py and run this script
* When prompted, pick a workflow for depositing bag to APTrust-Demo/APTrust-Repo/VT-Library S3:

Option 1: enter “1” (without the quotations) to deposit the ingest bag to APTrust-Demo storage, this option is for depositing test bags to APTrust-Demo storage system. Bags deposited here will be deleted after a certain amount of time

**Option 2**: enter “2” to deposit the ingest bag to APTrust-Repo and VT library S3 storage systems (this is the standard option). APTrust-Repository does periodic integrity checks on the bags deposited. Depositing bags to the VT library S3 storage system will allow us to access the bags if needed. Bags deposited to VT library S3 storage will be deleted every few months.

Option 3: enter “3” to deposit the ingest bag to VT library S3 storage system only

Option 4: enter “4” to deposit the bag to APTrust-Repo only

This will create a publication bag (publication folder in tar format and with tag values in it) at “Username/.dart/bags” (or a different path if the default folder setting in the DART app was changed) and has the same naming convention but with a .tar at the end. This bag will be uploaded to APTrust-Demo/APTrust-Repo/VT-Library s3 bucket based on the option selected in 5. The bags on demo or repo can be checked for successful upload at demo.aptrust.org or repo.aptrust.org

* The bag created can be extracted and the contents can be verified following the instructions in the section “Checking the bags created by DART and bag validation” (section 9) of the document: ScriptsSetupAndExecution\_CurationWorkflow\_Windows/ ScriptsSetupAndExecution\_CurationWorkflow\_Mac

1. **Complete client record for data publication:**

Go to [LibCRM](https://vt.libapps.com/libapps/login.php?site_id=17108&target=/), open the interaction created in step 4, under section 2: “fields for all types” update number of participants if more people were involved in the publication process, update duration in section 3 “Additional Info” and under section 4: “Attachments” update the provenance log and email interaction with the ones saved in the publication folder created in “CurationDir”(path provided in generate\_config) or “Curation” folder. Save the interaction. Data publication process is now complete.

***Other: Updating Client record, creating future versions?***

Clients can update/make changes to the published record and submit their publication for review again. Examples of when such an update is required include: when the client needs to update research doi/title after their manuscript is accepted, update the dataset after manuscript review, update README file upon acceptance/rejection of their manuscript etc.

Published datasets can be updated by accessing the published datasets in the client’s [VTDR](https://data.lib.vt.edu/) account under "My data" either by the client or by the curator impersonating the client’s record. Scrolling over the published article brings up a pen icon next to the size of the published dataset. Clicking this pen icon allows the client/curator to access the published item, where updates can be made to the metadata fields, clicking “Manage” on the item page allows for updating the files. After updating the metadata fields/files, the client/curator will click "Save changes", followed by "Publish", this will bring the dataset to the account review page, the curator can then review and publish it again as an updated version (e.g. ‘v0x’ where x is the new version number).

After publishing the dataset as an updated version, the curator will follow steps 11-16 to finish completing the client record. Note that the newer versions of the article will have the same base DOI since this DOI does not have an extension of '.vX' at the end where x is the updated version number.