**Aarhus University Science Museums herbarium sheet workflow**

**Imaging**

At the beginning of a digitisation session, the digitiser goes to the E drive on the workstation computer (located at Aarhus University Science Museums herbarium) and opens the Dassco\_Digitisation folder. They then make a subfolder there with the following naming convention: date in form of YYYYMMDD\_Institution\_Workstation\_Family\_Digitiser initials. See examples (below):

A group of black text

Description automatically generated

There should be one folder per day, per digitiser. The date on the folder should always represent the day the imaging took place.

Open CaptureOne.

Direct all images to be imaged to the folder you have just created.

The first image you take at the beginning of the day should always be a folder image. After that, image the contained herbarium sheets one be one (after adding a barcode to all specimens contained on each sheet). If you move to another physical herbarium folder, continue imaging in the same fashion and using the same folder on the E Drive.

At end of the day’s imaging, select all images and export tifs to the same folder on the E Drive.

**Processing of images**

At 11 pm each evening a program will run automatically.

[From Khaled’s understanding from the code, run\_scheduled.py (which automatically runs all 7 scripts) selects only one subfolder from the DaSSCo\_Digitisation that contains .tif files and processes it. It might still go through and process a subfolder that has already been processed depending on the configuration of the program. Aksel has included a variable called ALLOW\_PREVIOUS\_SESSION, and if this variable is set to True, it might process an already processed subfolder (if it has not been deleted).  
The program also includes a variable called DELETE\_IMAGE\_FOLDER (only used in run\_scheduled.py). If this variable is set to True, the selected subfolder will be deleted once it has been processed.]

The program goes through all of the images in the first folder and looks for a barcode. If no barcode is detected, it assumes it is a physical folder.

The program only checks the top part of the sheet – it does not check the whole sheet, in order to save time. Hence, if a barcode was placed outside of the area checked, it will not be detected. The workstation is also having issues with reflectivity of barcodes, resulting in them not being detected. Occasionally, the digitiser will also forget to add a barcode to the sheet. Hence, there are multiple scenarios where the program will incorrectly identify a specimen sheet as a folder.

The program then subdivides the images into sessions. A session [please let us change this name in future] is defined as one physical folder image followed by images of all herbarium sheets physically contained in that folder. Hence, if multiple physical herbarium folders were imaged, the program separates these out into separate sessions. If the program incorrectly identifies a specimen image as a folder, it will separate that image out (and not ingest it) and associate subsequent images with it as a session.

[Where are the sessions physically stored at this point? On a server? Is anything added to the local drive on the workstation as the program runs?]

The sessions are named based on earliest detected date\_image\_taken from the exif data in the original folder on the E drive. For example, if a folder was created on the E Drive on 13/08/2024 called 20240813\_AU\_Herb02\_Fabaceae\_CVH, but the first images taken and saved into that folder were from the 15/08/2024 and the last ones taken in that folder were from 20/08/2024, they would all be in session folders called 2024-08-15. It is this latter date which is visible in the UI.

Metadata is added to the specimen table in the dassco-au database for all specimen images (this is data that will go into the DaSSCo Asset Registry System and storage and from there to Specify and is derived from the Ingestion Client code). The dassco-au database is separate to the Species-OCR and Species-Web and runs on a MySQL server in a Docker container. Only Species-Web communicates with the database to retrieve or store information. The metadata here contains project data and image data. It does not include data extracted by reading the session folder images in Species-OCR or Species-Web. Note that the DaSSCo Ingestion Client and the Species-OCR are two different entities. They both use a package called DaSSCoUtils which contains utilities to create GUIDs, metadata, checksums and to upload assets to the N-drive.

The species table in the database comprises the following fields:

* id: The primary key. This is unique for each specimen image. Note that folder images are not listed in this table.
* guid: This is the GUID generated by the Ingestion Client and what the specimen images are renamed as.
* digitiser: This is generated automatically and this needs to be untangled so it is specific to each imaging session. Perhaps this should be added to the exif data in future and read from the exif data at this early stage.
* date\_asset\_taken: This is read from the image exif data and is the date that the image was taken with the camera at the workstation. If this is wrong, it is likely that the date is set up incorrectly on the camera.
* image\_file: This gives the original name of the image on the E drive of the workstation.
* checksum: This is the checksum value used to check for errors when transmitting data
* folder\_id: This is the foreign key for the folders table in the database. This number corresponds to the number of the folder displayed in the Species-Web UI.

Finally, the tif images of specimen sheets and json files with the metadata are uploaded to NHMD N drive. No folder images are uploaded to the N Drive. If a specimen image has been incorrectly identified as a folder, it will not have been added to the specimen table and will not have been uploaded to the N Drive.

**NHMD N drive**

Images are only temporarily being stored on the N drive, until we have our storage up and running.

The AU specimen images are stored on the NHMD N Drive in the following location: N:\SCI-SNM-DigitalCollections\DaSSCo\MASTER\_IMAGE\_STORE\Main\_image\_store\WORKHERB0002

Each time specimen images are uploaded to the N Drive, they are placed within their own folder with the name of the folder being the date they were ingested to the N Drive.

Each image will have an accompanying .json file which provides additional data on the image as provided by the DaSSCo ingestion client.

The images (and .json files) have been renamed with a globally unique identifier (see value in asset\_guid below).

This is an example of the contents of an image .json file from folder 2024-9-10:

{

"asset\_created\_by": "",

"asset\_deleted\_by": "",

"asset\_guid": "7e8-9-0a-0a-3a-05-1-001-01-000-0be2d8-00000",

"asset\_pid": "",

"asset\_subject": "",

"date\_asset\_taken": "2024-09-10T10:58:05+02:00",

"asset\_updated\_by": "",

"audited": false,

"audited\_by": "",

"audited\_date": null,

"barcode": [],

"collection": "Vascular plants",

"date\_asset\_created": null,

"date\_asset\_deleted": null,

"date\_asset\_finalised": null,

"date\_asset\_updated": null,

"date\_metadata\_created": null,

"date\_metadata\_updated": "",

"date\_metadata\_uploaded": "",

"digitiser": "Birgitte\_Bergmann",

"external\_publisher": [],

"file\_format": "tif",

"funding": "DaSSCo Tranche 1",

"institution": "AU",

"metadata\_created\_by": "",

"metadata\_updated\_by": "",

"metadata\_uploaded\_by": "",

"multispecimen": false,

"parent\_guid": "",

"payload\_type": [

"image"

],

"pipeline\_name": "PIPEHERB0002",

"preparation\_type": "sheet",

"pushed\_to\_specify\_date": null,

"restricted\_access": [],

"specimen\_pid": "",

"status": "",

"tags": {

"metadataTemplate": "v2\_1\_0"

},

"workstation\_name": "WORKHERB0002"

}

The date\_asset\_taken is the date the image was actually taken and is extracted from the exif data.

[Note that the collection\_name is wrong in the .json file – this is the value for NHMD. We need to see how Specify is set up for AU to see what this is for AU]

As it is difficult to quickly check through tif files, a jpeg generator is running at NHMD. Downsampled jpeg images are available in the following location: N:\SCI-SNM-DigitalCollections\DaSSCo\MASTER\_IMAGE\_STORE\JPEGs\WORKHERB0002

The jpeg folders and image names are exactly the same as the tif images. However, the json files are not reproduced in the jpeg folders, so you need to look in the tif folders for them.

Once the image processing pipeline is up and running, the jpegs will be automatically generated and available in the storage.

To facilitate checking of images, a script is also run which populates a database of image GUIDs and their barcodes and folder names, as well as the date the asset was taken. This is currently backlogged and will be available soon.

[Allison to insert details of where to access these here]

**Species-OCR**

For the folder images, an algorithm is run which detects and crops the label from the image of a physical folder, producing a .png. These images are sent to Species-Web.

The cropped .pngs are stored inside a Docker container called nhmaspeciesweb-web where Species-Web is running. See <https://github.com/NHMDenmark/DaSSCo-Tranche-1-work/issues/40> regarding storing folder images including the pngs.

The cropped .pngs have the same name as the tif files on the E Drive. The .png name for a specific folder is added to the Folders table in the database under “label”.

According to Aksel’s documentation, <https://github.com/NHMDenmark/Species-OCR/tree/main>, the specimen ID is extracted from the data matrix [BUT this does not seem to exist in the database]

Next, a script creates a threshold image from each session folder label, reads the full text, and adds it to the output file. This text is then categorised using a rule-based approach, so it is parsed into “Area”, “Family”, “Genus”, “Species”, “Variety” and “Subspecies”. This is stored in the folder table in the database.

The folders table in the database comprises the following fields:

* id: The primary key. This is the folder id number that can be seen in the Species-Web UI. This id links the specimen and folder-version tables with this table.
* image: This gives the original name of the folder image as saved on the E drive.
* label: This gives the name of the cropped .png image of the label that is saved in the docker container called nhmaspeciesweb-web.
* ocr\_read\_json: This is the OCR read of the folder data, including the confidence levels for each piece of text, structured as a JSON format.
* flagged: The folder record is flagged if it is not possible for the OCR read to find either a genus or a species.
* approved\_by: This the person who is logged into the Species-Web UI who approves the record.
* approved\_at: This is the date and time that the record was approved (presumably when the save button is pressed) in the Species-Web UI.
* session\_started\_at: This is the date the processed subfolder was created by Species-OCR.

Next a search of the GBIF database is performed, looking for a match of the highest classification level found in the previous script (actually this is incorrectly labelled and it is actually looking for the lowest classification level). A confidence rating is associated with the GBIF match. The data is then uploaded to Species-Web. If the confidence level of the match to GBIF is below ??

This is stored in the folder-version table in the database.

The folder-version table in the database comprises the following fields:

* id: The primary key for this table. There will be multiple lines for the same folder (each with its own id) if the detected values were overwritten in the UI.
* area: Area as detected by the Species-OCR. This is matched against an internal list.
* family: This is the family as detected by Species-OCR. If this field was overwritten and a new record created in the table, this field will contain the new values in the new record. The original record (line in the table) will retain the original ones for comparison. This allows the user to revert back in the UI, if necessary.
* genus: This is the genus as detected by the Species-OCR. If this field was overwritten and a new record created in the table, this field will contain the new values in the new record. The original record (line in the table) will retain the original ones for comparison. This allows the user to revert back in the UI, if necessary.
* species: This is the species as detected by the Species-OCR. If this field was overwritten and a new record created in the table, this field will contain the new values in the new record. The original record (line in the table) will retain the original ones for comparison. This allows the user to revert back in the UI, if necessary.
* variety: This is the variety as detected by the Species-OCR. If this field was overwritten and a new record created in the table, this field will contain the new values in the new record. The original record (line in the table) will retain the original ones for comparison. This allows the user to revert back in the UI, if necessary.
* subsp: This is the subspecies as detected by the Species-OCR. If this field was overwritten and a new record created in the table, this field will contain the new values in the new record. The original record (line in the table) will retain the original ones for comparison. This allows the user to revert back in the UI, if necessary.
* gbif\_match\_json: This is the data associated with the GBIF match against the name (and presumably is the full GBIF data for that taxon). It includes details such as basionyms, synonyms, authorship and sometimes even bibliographic references. If Species-OCR was unable to recognise genus or species, the value in the record will be NULL. If the “Detected” values are overwritten in the UI and a new record in the folder-version table created, Species-Web will again match to GBIF and should show a successful match in the new record. If the match is a synonym, this is flagged in the UI and you can see details of the synonym. Ideally, we should capture the synonyms and the name from the folder (the latter tagged as “Shelved as” when importing to Specify).
* highest\_classification: This is the *lowest* classification level detected. If it is a species folder, the value in this column will be species, if it is variety it will be variety etc.
* created\_at: This refers to the date the record was created in the database. The database generates this field. When data is edited in the UI, a new record is created in the folder-version table and the created\_at date in that case refers to the date that new record was added.
* folder\_id: This is the foreign key for the folders table in the database. It is the number of the folder record seen in the UK.
* created\_by: For the records created by Species-OCR, this value is NULL. For records created by editing the data in the UI, this is the user who edited the data and created a new record in the folder-version table in the database.

As detailed above, if the parsed OCR data in the “Detected” column in the UI is overwritten, this adds a new record to the folder-version table. Hence, when preparing the data for import into Specify, this needs to be taken into account (only the record with the latest created\_date should be kept in instances when there are two records with the same folder\_id in the folder-version table in the database.

**Species Web UI**

Following successful processing by Species-OCR, a user can open the Species-Web UI and a new folder will have been added for checking. The folders are named based on the date the first image in the batch being processed was taken (as extracted from the exif data using the ExifRead package). This is not always the same as the date that the image was taken, or when it was ingested to the N Drive, or when it was uploaded to Species-Web. To help easily locate images, batches (i.e., individual folders on the E Drive of the workstation) should only relate to a single digitisation day (the day all of the images in the folder were taken). All of the images in one folder on the E Drive should also only relate to a single digitiser.

To access Species-Web, type the following URL in your browser: <https://dassco-au.duckdns.org/>

Log in with the detail provided to you.

To add new users to Species-Web, contact an admin use who will add you.

Once logged in, the UI looks like this (below)

A screenshot of a computer

Description automatically generated

The UI will show all folders/sessions ever digitised. They are currently not being deleted from the interface, even after being fully approved. We need to change this

If any session folders contain images which have not been approved, they are highlighted in yellow. If all images in a session have been approved, the session is highlighted in green.

You can click on the arrow at the right of the field showing the folders for approving, to scroll through a list of all folder sessions (see below).

A computer screen shot of a computer

Description automatically generated

Alternatively, you can switch between seeing session folders that have been completely verified in the top folder field and having these removed, by selecting the clipboard icon to the far right of the field.

A screenshot of a computer

Description automatically generated

At the moment, it is necessary to approve all images in a session by clicking on the red “Not Approved” button at the top right of the image. You are then able to edit the data and a green “Save changes” button located at the bottom right of each image (see screen shot below) which you can use to save your changes.

In the future, the intention is to only manually approve the ones for which the program has identified an issue (i.e., it is not a clear match to GBIF data), and the user would not need to go through all images. However, as the program is still being tested, this has not yet been implemented.

A screenshot of a computer

Description automatically generated

Once an image has been approved, you can see who approved it, and when, in the UI below the image. The image will also have an “Approved” message in green at the top right of the session folder details in the UI. Two fields in the database are also updated: approved\_by and approved\_at. In the UI, these folders will now be highlighted in green.

If you made a mistake in editing or approving the data, simply click on the red undo button at the top right of the session folder info box.

A screenshot of a computer

Description automatically generated

The area data is matched against an internal list of the Broad Geographic Areas used by AU. If a match is not found, a red line appears on the right side of the screen underneath the Area. If a match is found, the line is green. Similarly, when GBIF makes an exact taxon match to the OCR read of the session folder image, a green line appears in the right pane of the UI for that session folder below species. If no match is made, the line is red. The screen shot below shows an example with two green lines i.e., matches were made for both the area and the taxon.

A screenshot of a computer

Description automatically generated

For the example below, matches were not found for both the area and the taxonomy resulting in two red lines in the UI.

A screenshot of a computer

Description automatically generated

The program also recognises the existence of synonyms. These can be seen as a yellow box to the right of the word “GBIF” in the UI (see image below). Clicking on the word “Synonym” shows you the details. It is not possible to do anything further with this in the UI.

A computer screen shot of a computer screen

Description automatically generated

The program also recognises the existence of homonyms (identified by the same taxonomic name, but different author names). Homonyms appear as part of a drop-down box to the left of the word “GBIF” in the UI, allowing you to select the correct homonym.

[Add image of homonym drop down box in UI]

[note that on the E drive of the local computer, there are thumbnails and .cop and .cof files in a subfolder called CaptureOne]

[Are we able to get any stats from the database such as how many images needed information adding by the verifier?]

The program does not delete any files from the E drive after it has finished processing the session.

[We need to think about whether we want to save images of folders – the tiffs, or pngs, and how.]

**What to do if the program incorrectly identifies an image as a folder? Example is when a herbarium sheet was not allocated a barcode**

Make a note of the folder number in the UI.

Look at the details of the previous folder image in the UI. This should be the folder in which this specimen is in. Make a note of this number as well.

Go to the collection and find the herbarium sheet using the correct folder image details. Add the barcode and make a note of the number.

In Species-Web enter details for the previous folder image into the incorrectly identified folder image and save.

Send the new barcode number, the number of incorrect folder image and the number of the correct folder image number to Allison at [info@dassco.dk](mailto:info@dassco.dk).

Allison will look at the data and tell you the original name of the image of the E Drive. That will then need ingesting separately using the ingestion client. Let Allison know when that has been done.

Later after the data has been ingested, the details for the incorrectly identified label image will be added manually to Specify (it can be added temporarily to “Barcodes and Assoc. not in App” in the issues log, similarly to how we do it at NHMD).

**Specimen data destined for Specify**

Need to extract data from Species DB and send daily to Allison at [info@dassco.dk](mailto:info@dassco.dk). Can this be automated? Is there a way that we can easily see what is in the DB (e.g., via viewing in a DB browser?)

There also needs to be a process of deleting from Species DB after everything has gone into Specify.

We need to decide on mapping to Specify. This is the current situation:

|  |  |  |
| --- | --- | --- |
| **Table** | **Field** | **Map to Specify?** |
| specimen | id | No |
| specimen | guid | No |
| specimen | digitiser | No |
| specimen | date\_asset\_taken | No |
| specimen | image\_file | No |
| specimen | checksum | No |
| specimen | folder\_id | [might be worth keeping this or not?] |
| folders | id | No |
| folders | image | No |
| folders | label | No |
| folders | ocr\_read\_json | No |
| folders | flagged | No |
| folders | approved\_by | [is there a field?] |
| folders | approved\_at | [is there a field?] |
| folders | session\_started\_at | No |
| folder-versions | id | No |
| folder-versions | area | Yes: Match to BGR in localities and include source |
| folder-versions | family | No? |
| folder-versions | genus | No? |
| folder-versions | species | No? |
| folder-versions | variety | No? |
| folder-versions | subsp | No? |
| folder-versions | gbif\_match\_json | Yes: Need to extract additional data from here e.g., synonyms |
| folder-versions | highest\_classification | Yes |
| folder-versions | created\_at | Yes? |
| folder-versions | folder\_id | No: Use to identify when two records with same folder\_id |
| folder-versions | created\_by | No |
| folder-versions | Column 13 | No |
| folder-versions | Column 14 | No |
| (Our DB) | Barcode | Yes |

[Allison to work on post processing script and to put some basic non-technical details here]

[Details of then imported to Specify]

**Quality Checking**

[Insert details of quality checking of images and data and protocols, frequency etc]

**Changes Requested**

Requests for changes to workflows, programs etc are managed via Github boards and evaluated on a quarterly basis (to ensure we have the resources and allowing us to prioritise). Bugs and issues are also managed via Github boards, but it is best to let someone know if something is urgent so they can go straight to the ticket and assess it.

If you are unsure about Github, or don’t have time to make a ticket during digitisation, we have an issues log for recording issues which is checked daily and a solution provided in the log. If the issue needs escalating, it will be added to the relevant Github board (and a link to the Github ticket added to the issues log). The issues log is on OneDrive and has been shared with you. Links to the Github boards are below.

Requested changes to herbarium workflow or workstation: <https://github.com/orgs/NHMDenmark/projects/8>

Requested changes to Species-OCR and SpeciesWeb and associated data ingests: <https://github.com/orgs/NHMDenmark/projects/41/views/1>

**Additional Documentation**

SpeciesOCR repo is here: <https://github.com/NHMDenmark/Species-OCR/tree/main>

SpeciesWeb repo is here: <https://github.com/NHMDenmark/Species-Web>

Currently there is very little documentation associated with Species OCR or SpeciesWeb or the entire workflow, but it will be added to this repo soon.

Herbarium workstation and workflow documentation is here: <https://github.com/NHMDenmark/Herbarium-Sheets-workstation/tree/main>

At the moment most workstation and workflow documentation is based on NHMD. Sara Stenz is working on making documentation more institution agnostic and Pip Brewer is working on current AU workflow.