**Import images into OMERO**

This document covers the processes for importing images into OMERO, detailed design of software solution/app for doing that as well as some common mistake when doing that.

I will use the term ***import*** to indicate the process of getting images into OMERO.

Process

To make the *import* happen, Phenotypers/technicians would need to put the images they took and collected into folder, then drop the folder to a specific location. For KOMP team, all images are dropped into [*\\jax.org\jax\phenotype\OMERO\KOMP\ImagesToBeImportedIntoOmero*](file://jax.org/jax/phenotype/OMERO/KOMP/ImagesToBeImportedIntoOmero).

After the folder is dropped, ***App*** will detect the creation of new files, create a submission form that contains metadata of images within each folder (see .xlsx file I attached in this folder) and then send it to the OMERO dropbox *(\\jax.org\jax\omero-drop\dropbox*) along with the submission form. OMERO importer then will read the submission form and import images.

OMERO Importer will create a .log file in corresponding folder that contains the importing-status of images within that folder. Images that are successfully imported will be flagged “*Success*” in the log file. Similarly, images failed to be imported will be flagged as “Failed” and the importer will log the reason why importing failed. For all successful imported images, the app will send messages to a Teams channel (name is “KOMP OMERO import Team”) to notify phenotypers/technicians. Then comes the next step of the process: we need to generate a .csv file that the app will read the log file and extract useful info of images(import status, image name etc) and create a .csv file named “logfilename\_log.csv” that contains omero link and test code of images; then send the generated file to target directory.

* For Eye Morphology: [\\jax.org\jax\phenotype\EYE-MORPHOLOGY\KOMP\images-omero](file://jax.org/jax/phenotype/EYE-MORPHOLOGY/KOMP/images-omero)
* For Gross Pathology: [\\jax.org\jax\phenotype\GrossPathology\KOMP\images-omero](file://jax.org/jax/phenotype/GrossPathology/KOMP/images-omero)
* For ERG:

\\jax.org\jax\phenotype\ERG-V2\KOMP\images-omero

Design

GitHub repository:

*https://github.com/TheJacksonLaboratory/Images\_Import\_Scripts.git*

As you can see from above, ***Import*** is made up by 3 steps:

1. Drop images to the folder.
2. Send image folders to DropBox.
3. Read the log file.
4. Create JaxLims import file

Except for step 1), everything is done automatically. This is achieved via a powerful library of Python called ***WatchDog*** (official document: <https://python-watchdog.readthedocs.io/en/stable/>). The project is composed by the following parts:

* *transfer\_to\_omero.py* : Script that detect creation of new image folder; grab images to create submission form; send them to OMERO Dropbox.
* *transfer\_to\_lims.py:* Script detect creation of log file in OMERO dropbox; read log messages and generate .csv file that used to import image info to JaxLims.
* *read\_config.py:* Script to read configuration file user provided.
* *az\_devops.py:* Script to create PBI on Azure DevOps Board if any runtime error detected.
* *cleaner.*py: Deleted all imported folders.

Algorithms(*transfer\_to\_omero.py*):

1. Read the configuration file you provided, extract useful information to setup everything.
2. Start monitor the image import folder.
3. If new file is created, check if it is a directory or not, if not, ignore it and continue. Otherwise, iterating files in the directory.
4. For each file in the directory, strip space of its name, then split it by “\_” to get the animal id and eye information (if it’s eye morphology image).
5. Use the animal id to query the database to get information needed to create submission form (for details, see the sample submission form I attached in the folder/directory) in same directory.
6. Send the folder to DropBox.

Algorithms(*transfer\_to\_lims.py*):

1. Read the configuration file you provided, extract useful information to setup everything.
2. Start monitor the DropBox.
3. If new file is created, check if it is a .log file or not, if not, ignore it and continue. Otherwise, read each line of the file.
4. For each line in the file, check if the line contains word “success”, get image filename out from the line, record it.
5. Query the database using filename and testcode, then query omero for the image url.
6. Send the generated file to corresponding target folder (see above).

Things worth noticing

1. Be sure you are using the right OMERO username when importing, for example, if you want to import an image into group a), you need to use the username of the owner of group a). That means, in the *config.yml* file, set the attribute ‘group\_owner’ under the schema ‘*transfer\_to\_omero*’ to be the OMERO/JAX account username of the group owner.

A screen shot of a computer code

Description automatically generated

1. If you change your JAX account password (i.e., OMERO password), be sure to update it in the configuration file. The app will still be running but won’t do anything meaningful if you did not update.
2. If you want to make change to the code, be sure to follow these steps: 1. switch to dev branch (or you can create your own featured branch); 2. make sure everything works as expected; 3. commit the change; 4. Merge to master branch.

Current Status

The app is currently hosted and actively running on server bhwin0236.jax.org (location: C:\Program Files\KOMP\SoftwareUtilities\Images\_Import\_Scripts). To run it on the server, open your command prompt, cd to “C:\Program Files\KOMP\SoftwareUtilities\Scripts”, there are several batchfile scripts in the directory, run “export\_to\_omero.bat” to start importing images to omero; run “export\_to\_lims.bat” to start importing images info to jaxlims. Or you can create tasks using windows task scheduler to automate it. If you would like to use it somewhere else, follow the steps in the README file in the repository to set it up.