Presentation is available at

https://downloads.openmicroscopy.org/presentations/2018/Users-Meeting/Workshops/Metadata-Import/slides/index.html

This is the first part of the three parts Metadata workshop held at the Annual Users meeting:

- Part 2 https://downloads.openmicroscopy.org/presentations/2018/Users-Meeting/Workshops/Metadata-Analysis/analysis.pdf
- Part 3 https://downloads.openmicroscopy.org/presentations/2018/Users-Meeting/Workshops/Metadata-Handling/handling.pdf

Import metadata

Description

In this first part, we show how to import data for another user into OMERO using various import strategies. The user importing the data needs to have some admin privileges. More information about restricted privileges can be found at https://docs.openmicroscopy.org/latest/omero/sysadmins/restricted-admins.html

For this workshop, we will mainly use the Command Line Interface (CLI) to import and manipulate data.

The import will done by a facility manager ***importer1.***

Setup

- The Desktop client OMERO.insight has been installed on the local machine. The installation instructions can be found at http://help.openmicroscopy.org/getting-started-5.html#installing**.**
- OMERO.server has to be installed to import images using CLI. The installation instructions can be found at https://docs.openmicroscopy.org/latest/omero/users/cli/installation.html.
- To use the cli-render plugin, OMERO.py has to be installed. This is included with OMERO.server but if you are not planning to import using CLI, you can **only** install OMERO.py.
- We assume that you have `pip` already installed.
- To install the cli-render plugin, run:
    a. `$ pip install omero-cli-render`

Desktop client Import

In this example, we show how to import data for another user. A facility manager *importer1* with restricted privileges imports the data for *user-1*.

1. Launch OMERO.insight.
2. In the OMERO login dialog, click the wrench icon and then add the server address in the dialog i.e. **outreach.openmicroscopy.org**. Click *Apply*.
3. Enter your *Username* and *Password* and click *Login*.
4. Click on the Import Icon in the toolbar.
5. Use the *File Chooser* to locate the data to be imported. We will import the DICOM files listed in https://github.com/ome/training-repos/blob/master/data.md ("Brain").
6. Select the image data, click on the right Add arrow .
    a. Optional: Create a Project **medical.**
7. *(Optional)* Go to the *Options* tab
    a. Click on Add tag to images ⊞ : to bring the Tag selection dialog.
    b. Select the tag(s) on the left-hand side.
    c. Click to move the tag(s) to the right-hand side.
    d. Click *Save.*
8. In the Import Location dialog, select:
    a. the Group
    b. the User to import data for
    c. Optionally, select the target Project and/or Dataset
9. Click *Add to the Queue.*
10. Click *Import.*
11. Check that the images have been imported for the specified user.

Advantages:

- Users can validate that import worked.
- Failed imports can be repeated and/or reported to QA etc..
- Users do not have to wait for import to be scheduled.

Limitations

- Installation of a Desktop application to import data.
- The number of files that can be imported at the same time is limited. The default is 2000. The option is configurable client-side.
- Import can be slow due to the data transfer to OMERO via the client.

- The person doing the import **must** be in the same group as the user. This is a limitation of the Desktop client.

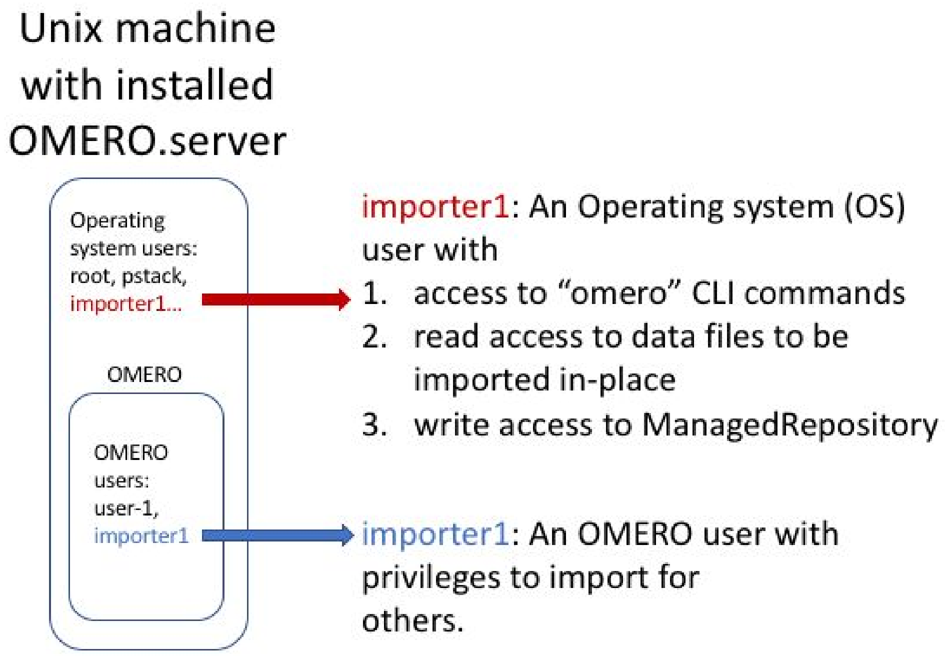
More information about import options using the Desktop client can be found at http://help.openmicroscopy.org/importing-data-5.html.

In-place Import CLI

In this example, we show how to import data for another user and how to avoid data duplication by doing
a so-called *in-place* import. A facility manager *importer1* with restricted privileges imports the data for *user-1*. The facility manager has been given the ability to import for others.
For this training, the path to the OMERO.server is `/opt/omero/server.`
The facility manager is also a UNIX user.

Important:
Someone wanting to perform an in-place import **MUST** have:

- a regular OMERO account
- an OS-account with access to `bin/omero`
- read access to the location of the data
- **write** access to the ManagedRepository or one of its subdirectories. More information about the ManagedRepository can be found at https://docs.openmicroscopy.org/latest/omero/developers/Server/FS.html


Unix machine with installed OMERO.server
Operating system users: root, pstack, importer1...
OMERO
OMERO users: user-1, importer1
importer1: An Operating system (OS) user with
1. access to "omero" CLI commands
2. read access to data files to be imported in-place
3. write access to ManagedRepository
importer1: An OMERO user with privileges to import for others.

1. Open a terminal and connect to the server as ***importer1*** using ssh.
2. Go to /OMERO and look at the subdirectories.
3. Go to /OMERO/in-place-import i.e. `cd /OMERO/in-place-import.`
4. The ***importer1*** user logins as *user-1*:
    a. `$ /opt/omero/server/OMERO.server/bin/omero --sudo importer1 -u user-1 login`
5. Create a Dataset **import_one** as *user-1*:
    a. `DID=$(/opt/omero/server/OMERO.server/bin/omero obj new Dataset name=import_one)`
6. Import the data *"in-place"* in the newly created Dataset. The option for performing an in-place transfer is `--transfer:`
    a. `$ /opt/omero/server/OMERO.server/bin/omero import -d $DID --transfer=ln_s /OMERO/in-place-import/svs/77917.svs`
7. Check that the image is successfully imported and open it in OMERO.figure.

More information about import options using the CLI import can be found at https://docs.openmicroscopy.org/latest/omero/sysadmins/in-place-import.html and https://docs.openmicroscopy.org/latest/omero/users/cli/import.html

Advantages:

- All in-place import scenarios provide non-copying benefit. Data that is too large to exist in multiple places, or which is accessed too frequently in its original form to be renamed, remains where it was originally acquired.
- The person doing the import **does not need to** be in the same group as the user.

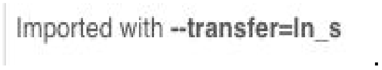
Limitations:

- Only available on the OMERO server system itself
- Do not move the files after an in-place import. OMERO may no longer be able to access them if you do.

Bulk Import CLI

In this example, we show how to combine several import strategies using a configuration file. This is a strategy heavily used to import data to https://idr.openmicroscopy.org/.
We import one study (set of images) named *idr0021.* For this training, the path to the OMERO.server is `/opt/omero/server.`

1. Open a terminal and connect to the server as *importer1* using ssh.
2. Description of the files used to set up the import, the files are in the directory idr0021-scripts under /OMERO/in-place-import:
    a. **idr0021.tsv**: this file has at least two columns where the first column is the name of the target Dataset and the second one is the path to the file to import. We will import the whole study, which consists of ~400 files in 10 Datasets.

   b. **bulk.yml**: this file defines the various import options: transfer option, checksum algorithm, format of the *.tsv* file, etc. Note that setting the *dry_run* option to true allows to first run an import in *dry_run* mode and copy the output to an external file. This is useful when running an import in parallel.
3. Go to `/OMERO/in-place-import` i.e. `cd /OMERO/in-place-import.`
4. The ***importer1*** (Facility Manager with ability to import for others) user logs in as *user-1*:
   a. `$ /opt/omero/server/OMERO.server/bin/omero --sudo importer1 -u user-1 login`
5. Import the data using the `--bulk` command:
   a. `$ /opt/omero/server/OMERO.server/bin/omero import --bulk idr0021-scripts/bulk.yml`
6. Go to the webclient during the import process to show the newly created dataset.
7. Select an image.
8. In the right-hand panel, select the *General* tab to validate:
   a. Click on to show the import details.
   b. Validate that In-place import is indicated 

Imported with **--transfer=ln_s**

.

Advantages:

- Large amount of data imported using one import command.
- Reproducible import.

Limitations:

- Preparation of the .tsv file.

More information about import options using the CLI import can be found at https://docs.openmicroscopy.org/latest/omero/users/cli/import.html.

Render data using CLI

In this section, we will now change the rendering settings of the images imported in OMERO in two different Datasets in a bulk manner. The cli-render plugin is available from https://pypi.org/project/omero-cli-render/.
This section assumes that the plugin has already been installed.

1. On your local machine, open a terminal.
2. Go to the place where you can run `bin/omero` commands e.g. in the `OMERO.py-xxx/` directory.

3. Description of the files used to set up the rendering. The files can be found under https://github.com/ome/training-scripts/blob/v0.1.0/maintenance/scripts/:
    a. *renderingdef.yml* and *renderingdef2.yml* are files defining the various rendering parameters, such as the channel color, channel name and minimum and maximum values.
    b. *renderingMapping.tsv:* this file has two columns, in the left-hand one there are the images to be rendered and in the right-hand side column points to the appropriate *renderingdef.yml* for that image. This file is consumed by the bash script (see below) and thus is not used in the workshop itself.
4. Change the rendering of images in one Dataset:
    a. Run `bin/omero render set Dataset:$ID local_path/to/renderingdef.yml` where the `$ID` is the ID of the **CDK5RAP2-C** Dataset.
    b. This will modify the rendering settings of each of the Image included in **CDK5RAP2-C** Dataset.
5. Verify the change in the browser.
6. Change the rendering of images in two Datasets:
    a. Run `bin/omero render set Dataset:$ID1 Dataset:$ID2 renderingdef2.yml` where the `$ID1` and `$ID2` are the IDs of **CDK5RAP2-C** Dataset and **CENT2** Dataset respectively.
    b. This will again change the rendering settings of Images in **CDK5RAP2-C** Dataset as well as in **CENT2** Dataset.
7. It is also possible to modify the rendering settings in batch using, for example, a shell script such as https://github.com/ome/training-scripts/blob/v0.1.0/maintenance/scripts/apply_rnd_settings_as.sh
8. which uses `HQL` to find the Images IDs in OMERO and deliver them to the `omero-cli-render` plugin. The script reads the Datasets in which the Images are located in OMERO are listed from a *renderingMapping.tsv* file, such as https://github.com/ome/training-scripts/blob/v0.1.0/maintenance/preparation/renderingMapping.tsv
9. **(not covered today)** Go to the folder when the script is located.
10. Run the bash script with the default parameters:
    a. $ `sh apply_rnd_settings.sh`
    b. The script could be run by a facility manager on behalf of other users.

Metadata Import CLI (not covered today)

In this example, we show how to add Map Annotations to the idr0021 dataset. This plugin is not yet available on PyPI but it can be used as part of the default installation. We are currently migrating it to its own repository, see https://github.com/ome/omero-cli-metadata.

1. Log in to webclient on the target server and create a Project: **in-place-idr0021*.***
2. Drag and Drop the ten Datasets of idr0021 you just imported, into the Project: **in-place-idr0021*.*** Also, note the ID of the Project: **in-place-idr0021*.***
3. On your local machine, open a terminal.
4. Go to the place where you can run `bin/omero` commands e.g. in the `OMERO.py-xxx/` directory.
5. There is a CSV file which is a table with one row per image describing what metadata will be added to each Image in a given Project. The CSV table is converted into an OMERO.table (an HDF5 table) using the command 8b below. The OMERO.table has an additional column (when compared to the original CSV file), which lists the ID of the Image in OMERO.
6. In the second step, a **bulkmap-config.yml** file specifies the categories of the Map Annotations (Key-Value pairs) such as Gene or Phenotype. In the command 8c below the OMERO.table on the Project: **in-place-idr0021** is read and the values are written into Key-Value pairs on each Image. The Key-Value pairs are categorized according to the bulkmap-config.yml file.
7. Below is the description of the files used to set up the metadata population:
    a. **Idr0021-annotation.csv** describes the setup of the future OMERO.table created in the command 8b listed below.
    b. **Idr0021-bulkmap-config.yml** specifies the categories of the Map Annotations (Key-Value pairs).
8. Run:
    a. `$ export OMERO_DEV_PLUGINS=1`
    b.
    ```
    $ bin/omero metadata populate --report --batch 1000 --file
    idr0021-subset-annotation.csv Project:<project_id> (enter value)
    ```
    c.
    ```
    $ bin/omero metadata populate --context bulkmap --cfg
    idr0021-bulkmap-config.yml --batch 100 Project:<project_id>
    ```