SWIFT

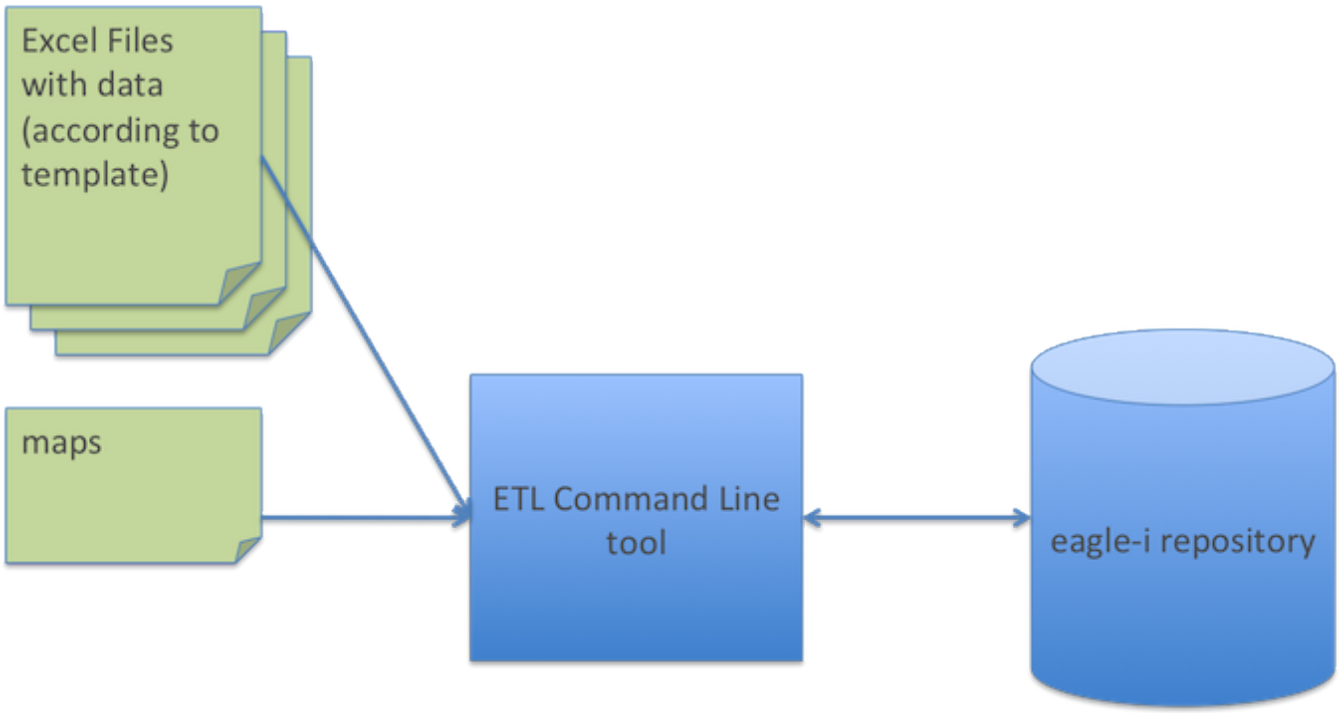
The ETL toolkit recently underwent a major re-architecture that will be part of our upcoming 2.0MS4 release. As a result, templates generated with versions prior to 2.0MS3 will no longer supported. Grab a snapshot of the 2.0MS3 code if you'd like to start experimenting.

The toolkit described herein is currently not user-friendly (though it works well – we use it routinely to bulk-upload data for the Consortium members). If you encounter issues, please do not hesitate to contact us.

SWIFT (*Semantic Web Ingest from Tables*) is a toolkit that allows experienced users to bulk-upload data into an eagle-i repository, via ETL (Extract, Transform and Load). Currently the toolkit supports only Excel spreadsheets as input files.

This guide provides an overview of system administrator tasks pertaining to ETL and the usage of the SWIFT toolkit. The ETL workflow requires a person with domain knowledge and understanding of the eagle-i resource ontology to prepare the input files for optimal upload. This topic is outside the scope of this guide.

The figure below is a high level depiction of the ETL process for spreadsheets.



The SWIFT toolkit is comprised of:

- an **ETLer** - command line program that executes a bulk upload
- a **deETLer** - command line program that deletes a previous ETL upload
- A **bulk workflow** - command line program that executes workflow transitions on groups of records, e.g. Publish, Return to curation, Withdraw
- an **ETL Input Generator** - command line program that allows system administrators to generate spreadsheet templates and map files for the various resource types of the eagle-i resource ontology (e.g. a template/map for antibodies, for instruments, etc.)

Prerequistes

The SWIFT Toolkit requires:

- A Unix-like environment including a terminal for executing commands
  - MacOS and Linux users don't need to install anything extra. For MacOS, use the Terminal app under Applications/Utilities
  - cygwin is recommended for Windows users
- A **Java 1.7** runtime environment
  - execute the command {{ java -- version}} to find out what version you have.
  - If necessary, you may download the JRE directly from Oracle and follow the installation instructions.

Download

The SWIFT toolkit is packaged as a zip file, and can be downloaded from our software repository.

Download the SWIFT toolkit distribution `eagle-i-datatools-swift-[version]-dist.zip`, unzip it into a dedicated directory, and navigate to it. For example

```
mkdir ~/eagle-i
unzip -d ~/eagle-i eagle-i-datatools-swift-2.0MS3.01-dist.zip
cd ~/eagle-i/swift-2.0MS3.01
```

Input generation instructions

To generate etl templates and maps, navigate to the dedicated directory (above) and run the script:

```
./generate-inputs.sh -t typeURI
```

*You may obtain the type URI from the eagle-i ontology browser . Use the left bar to find the most specific type you need, select it and grab its URI from the browser's address bar, e.g. http://purl.obolibrary.org/obo/ERO_0000229 for Monoclonal Antibodies.

Innocuous warnings are produced when generating the templates; these may safely be ignored. If you encounter errors or issues, please do not hesitate to contact us.

This script will create/use two directories with obvious meanings: `./maps` and `./templates`

Transformation maps will be contained in a subdirectory of ./maps named after the type and ontology version, e.g:

```
./maps/instrument_ont_v1.1.0
```

ETL instructions

The ETLer expects data to be entered into one of the generated templates, and a few conventions to be respected (see Appendix A) . A data curator usually makes sure that the template is correctly filled. In particular, the location of the resources to be ETLd (e.g. Lab or Core facility name) must be provided in every row of data.

1. Place your input files (i.e. the completed templates) in a directory of your choice, e.g. `dataDirectory`. All files contained in this directory will be processed by the ETLer.
2. To run an ETL, execute the following command (note that all records will be uploaded in the requested workflow state):

   ```
   ./ETLer.sh -d dataDirectory [-p DRAFT|CURATION|PUBLISH] -c username:password -r repositoryURL
   ```

   If you are practicing the ETL process, you may wish to upload your data to the common eagle-i training node. In this case, if your directory is named dataDirectory, the script would be executed as follows (default workflow state is DRAFT):

   ```
   ./ETLer.sh -d dataDirectory  -c L4:Level4 -r https://training.eagle-i.net
   ```

   Note that the data that is uploaded to the training node CAN be viewed and modified by others even in a draft state (even if you subsequently lock the records). Note also that the information in the training node is not persistent as the node is refreshed periodically.

3. To verify the data upload, log on to the SWEET application and select the lab to which the ETLd resources belong.

De-ETL instructions

Resources that are uploaded to an eagle-i repository via ETL are tagged with the name of the file from which they were extracted. It is therefore relatively simple to de-ETL an entire file. To do so, execute the following command:

```
./deETLer -f filename -c username:password -r repositoryURL
```

Appendix A. Input file odds and ends

- When ETLing a primary type, there are usually resources of other types that are related to it (e.g People, Organizations, Publications). It is best to enter information for these related types in a template of their own. For example, when ETLing a Monoclonal Antibody, it is best to have separate files for related Hybridoma Cell Lines, People and Publications. The primary file (Monoclonal Antibody) will contain references to instances from these other secondary files **- references in the primary file need to use the exact name (ignoring case) entered in the secondary file for the correct linkage to occur**. ETL the secondary files first and then the primary file.
- It is best to use the Sweet to add the Organization (e.g. lab) to which these resources are associated, and then reference this name in the files.
- If there is more than one value for a given column, enter values separated by ; (semicolon). Conversely, check your input file for the presence of ; in values that are not meant to be split and substitute for a different character.
- The first two columns (hidden) of a template are reserved for metadata. Please do not modify them or the name of the Tabs.
- Every resource needs to have a name and a type as a minimum. If the template has a type column, you must enter a value even if it is superfluous (e.g. in a template for Journal Article, you still need to enter Journal Article as a type) => this should be fixed in the near future.
- You must always enter the Organization to which the resource is associated.