

SWIFT

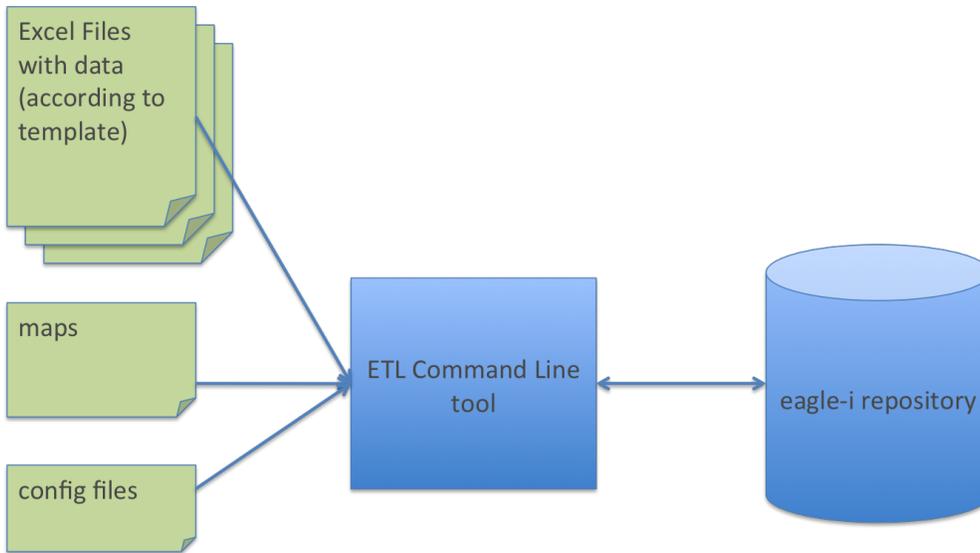


The toolkit described herein is currently not user-friendly (though it works well – we use it routinely to bulk-upload data for the Consortium members). If you encounter issues, please do not hesitate to [contact us](#).

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 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 **De-ETLer** - command line program that reverses a previous ETL upload
- A **bulk workflow** command - 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.)

The SWIFT toolkit is packaged as an executable jar, and can be downloaded from our [software repository](#). We have also manually packaged a zip file that contains shell scripts to wrap the main commands and a few pre-generated templates and maps (only for the most recently released eagle-i software and ontology versions); the following is a temporary location, while we integrate the generation of this package into our build processes:

<https://open.med.harvard.edu/svn/eagle-i-dev/apps/trunk/dev-resources>

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

```
mkdir ~/eagle-i
unzip -d ~/eagle-i eagle-i-etl-dist-1.7MS2.01.zip
cd eagle-i/eagle-i-etl-dist-1.7M2.01
```

Input generation instructions

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

```
./generate-inputs.sh --type INSTRUMENT|SERVICE|PERSON|ORGANIZATION|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. However in this case you will need to add `mapfileinfo.properties` (see below).

 Innocuous warnings are produced when generating and uploading 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
```

At the moment a third input, `mapfileinfo.properties`, is not generated. The ETLer looks for it under the type and version subdirectory, e.g. `./maps/instrument_ont_v1.1.0/mapfileinfo.properties`

The zip file contains pre-generated maps and templates for a few types, and includes these property files. If you need to generate inputs for a different type, please make sure to copy this property file to the appropriate subdirectory and edit it. We are working on automatically generating this file, this step will go away soon ;-)

ETL instructions

 The ETLer expects data to be entered into one of the generated templates, and a few conventions to be respected (outside the scope of this guide) . 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 and must correspond to a location already entered in the eagle-i repository via SWEET.

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 script:

```
./etl.sh --dir dataDirectory --workflow DRAFT|CURATION|PUBLISH username:password 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:

```
./etl.sh --dir dataDirectory --workflow DRAFT L4:Level4 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; it is periodically reset programmatically.

All the data will be uploaded to the requested workflow state.

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