

# import.py

```
usage: import.py [-h] [-v] [-d] [-a AUTHOR] [-t TITLE] [-i ISO] [-l DSHANDLE]
                [-m META] [-u UID] [-e EMAIL] [-c]
                document [parent]
```

import documents into application

positional arguments:

document	document folder
parent	UUID of parent collection

optional arguments:

-h, --help	show this help message and exit
-v, --verbose	show informational messages
-d, --debug	show debugging messages
-a AUTHOR, --author AUTHOR	set author
-t TITLE, --title TITLE	set title
-i ISO, --iso ISO	set language (iso code, 3 characters)
-l DSHANDLE, --dshandle DSHANDLE	set DSpace handle link
-m META, --meta META	set metadata fields (field=value)
-u UID, --uid UID	uid of importing user
-e EMAIL, --email EMAIL	email address of importing user
-c, --continue	continue even if there are errors (dangerous!)

Only one of UID or email address is needed to specify the importing user.

The import tool adds a document to the editor. The importer creates a UUID for the new document; then the folders are added to the application's file storage and revision and page entries inserted into the database.

The UUID of the parent collection should be provided on the command line if you want the document to show up in the interface.

You can provide the UID of the user doing the importing by providing the --uid parameter.

You can add metadata by providing --title, --author, --iso (language) and --dshandle (DSpace link) options, or add random metadata by providing --meta tag=value pairs.

The document's directory structure needs to be in the correct format so the importer can find ALTO XML and pictures files. If you have to import a lot of documents, you'd do well to make a trivial shell script to automate the process. Several working examples of previous imports can be found in the application's path.

```
/document  
  
  /pages/1/5eb63bbbe01eed093cb22bb8f5acdc3  
  
  /pages/2/2fbb46d103e517b49f2acffd9d51b4b6  
  
  /thumbs/00001.jpg  
  
  /thumbs/00002.jpg  
  
  /images/00001.jpg  
  
  /images/00002.jpg  
  
document.pdf
```

For the sake of completeness you can include the PDF in the import path, but as of now there is no direct link in the interface to download this PDF. Use the DSpace link field.

You can update the metadata of any inserted documents after the fact by making sure the document has a handle link to DSpace – you can add this link from the web interface if you did not do so at the time of import – and running the `dscacher` tool, which will retrieve the JSON metadata and cache it locally in the database.

See also:

dscacher.py: get/sync Dublin Core metadata from DSpace