
Command line Tools for Ingest in DigiTool

Overview

The ingest process via the command line consists several stages. Firstly a script is run to create a directory structure for the ingest in the file system and data base. The files to be ingested will need to be placed in the ingest directory tree before running scripts to trigger the actual ingest of the files to the repository.

Command line scripts

1 - FSUTIL script

This will be the first stage in any ingest

The fsutil.sh script allows you to create an ingest database entry and a specific load directory (highlighted below) for use in loading elements into the DigiTool repository. Ultimately, the load will be performed either from the Web Ingest interface, or by using command line tools (ingest.sh or tasker.sh). The fsutil script is run on the server with specific parameters which determine the appropriate location and ownership of the load directories that will be created.

The "profile" directory is located by typing `>j_home` at the terminal. In most cases this changes to directory `/exlibris/dt1/j3_1/digitool/home/`, but this may differ in some installations.



The fsutil.sh script is found in the j_bin (alias j_bin) directory of the DigiTool version.

Following are the parameters that can be defined using the fsutil.sh script:

```
fsutil.sh <AdminUnit> <UserName> -desc="<SomeDescription>" -note="<SomeNote>" -asap
```

fsutil.sh - The script name

AdminUnit - The code of the Administrative Unit in which the specific load directory will be created and which will be available from the webingest module (DTL01 in the example above).

UserName - The name of the staff user that will be "assigned to" the ingest load.

-desc - Allows you to define an ingest_description for the ingest activity that will ensue based on the created load directory.

-note - Allows you to define an ingest_note for the ingest activity that will ensue based on the created load directory.

asap - When -asap appears as a command line parameter, the resulting ingest activity will be enabled for immediate activation, after all server-side files and definitions have been defined.

E.g. `fsutil.sh DTL01 USER -desc="The loneliness of the long distance runner"
-note="For internal use only" -asap`

Two additional ingest command line tools exist: `tasker.sh` and `ingest.sh`, also available in the `j_conf` directory.

2 - Tasker Script

The `tasker.sh` script transforms materials found in the transform directory, utilizing the specified transformer, and runs any specified tasks. The newly transformed materials are copied to the ingest directory and, by default, loaded automatically into the repository.

Following are the parameters that can be defined using the `tasker.sh` script (`j_bin`):

```
./tasker.sh <AdminUnit> <UserName> <user:pw> <directoryName>
```

`tasker.sh` - The name of the script

AdminUnit - The name of the Administrative Unit in which the transform directory is located.

UserName - A staff user

user:pw - The username and password for the repository. (Currently not active).

directoryName - The name of the specific load directory within the Administrative Unit's general load directory. The name of the specific load directory will always be of format: `load_ing<nnnn>`. If the specific load directory is named `load_ing1234`, for example, the script parameter would be simply `ing1234`.

E.g. `./tasker.sh DTL01 USER user:user ing1234`

3 - Ingest Script

The ingest.sh script loads already transformed digital entities and file streams, stored in the ingest directory of the specific load directory, into the repository (see: Notes about the ingest directory structure... below).

Following are the parameters that can be defined using the ingest.sh script (j_bin):

```
./ingest.sh <AdminUnit> <UserName> <user:pw> <directoryName> INGEST
```

ingest.sh - The name of the script

AdminUnit - The name of the Administrative Unit in which the ingest load directory is located.

UserName - A staff user

user:pw - The username and password for the repository. (Currently not active).

directoryName - The name of the specific load directory within the Administrative Unit's general load directory. The name of the specific load directory will always be of format: load_ing<nnnn>. If the specific load directory is named load_ing1234, for example, the script parameter would be simply ing1234.

INGEST - Command for loading the materials into the repository.

E.g. `./ingest.sh DTL01 USER user:user ing1234 INGEST`

Notes about the ingest directory structure

Within a specific load_ing<nnnn> directory, for example in:

```
/exlibris/dtl/j3_1/digitool/home/profile/units/DTL01/load/load_ing1234
```

An ingest_settings.xml file will need to be placed in the root load directory - a configuration file that defines the ingest settings including the appropriate transformer and task chain/parameters to use for the ingest activity as well as the ingest_name which is both descriptive, i.e. the name of the activity, as well as functional, i.e. when the ingest_name is defined, tasker.sh will actually ingest the materials after finishing the transformation and any tasks defined. Removing the ingest_name from the ingest_settings.xml file will cause tasker to not actually ingest the materials to the system (similar to a dry run).

transform - a directory under the root ingest load directory that contains file streams and a single digital entity template (see the file structure diagram above). The materials in the transform directory need to be transformed, and any specified tasks run upon them, in order to prepare them for load/ingest into the repository. When ready, the "transformed" digital entities are copied to the ingest directory.:

digital_entities - a sub-directory of transform storing the DigitalEntityTemplate.xml to be applied to the load.

streams - a sub-directory of transform that stores the file streams to be transformed/loaded.

Once the transformer and any tasks are run, the digital entities and file streams are copied into their same-named sub-directories in the ingest directory.

ingest - a directory under the root ingest load directory that will/should contain the actual digital entities and file streams that will be loaded into the repository. After any transformation or tasks have been performed, the digital entities are in the proper format and ready to load from this directory.

digital_entities - a sub-directory of ingest that stores the actual Digital Entities (post-transformed) to be loaded/ingested into the repository.

streams - a sub-directory of ingest that contains the actual (post-transformed) file streams that will be loaded/ingested into the repository.

Practice creating a few ingest activities using the Web Ingest interface. Examine the resulting definitions created in the associated server-stored load directory, as described in this document. This will allow for a better understanding of appropriate ingest_settings.xml, digital_entities, etc., which will aid in the correct usage of command line tools.

You can see working examples of ingest load directories already existing on your server from ingests previously performed via the web ingest interface.