

Directory Ingest Service

The Directory Ingest Service (diringest) is a servlet that ingests a zipfile of data. Capabilities include:

- automatic creation of objects and datastreams based on METS file and directory structure
- automatic creation of RELS-EXT to mimic directory structure (can be customized to interpret this hierarchy in many ways)

The [basic documentation](#) in the Fedora users guide is not very detailed.

A [helpful tutorial](#) on this tool provides more information than the base Fedora documentation.

A Java applet called SIPCreator is distributed with Dirlngest. We have not been able to make it work yet (the dialog for loading files doesn't actually display files to insert – security problem?). Even when it is fully functional, it will probably not help with many of the problems we have.

The [batch build and ingest tools](#) are related to the directory ingest service, and provide some complimentary features, but they are very simplistic. The build tool simply combines a (FOXML or METS) template with xml descriptions of the object datastreams to make simple (non-hierarchical) objects in FOXML format. The ingest tool simply ingests a set of FOXML files into the repository.

While this tool comes close to meeting our ingest needs, it still has drawbacks:

- Users must build a METS file containing all the metadata for all items. (Note that the bulk of our [Ingest Tool](#) is dedicated to building a METS file that is somewhat similar to the one required by Dirlngest.)
- It assumes that all items go into the "general collection", which we don't want. It is possible to specify crules.xml in such a way as to connect to a collection object, but we would need a separate crules.xml for each collection unless the collection object is being created with each ingest.
- It doesn't specify anything about disseminators.
- Certain types of RELS-EXT data (like sequential page numbers) cannot be created with this method.
- It doesn't allow modification of existing objects.

Basic steps in using Dirlngest

1. Create images and metadata files as normal
2. Create MIX metadata by running JHOVE manually
3. Distribute images and metadata files to the proper hierarchical directory structure.
4. Create a METS document that describes the relationships between the files (using something like our ingest tool!)
5. Zip all the files
6. If necessary, modify the crules.xml file to give datastreams the correct labels and specify proper relationships between the objects.
7. Run Dirlngest

Installation

- Download the diringest from www.fedora.info.
- Deploy it to Tomcat
- Change the configuration file and change the username and pass

Test Ingests

- **A photo item from the slocum coll**
 - Prepare a METS file using the examples
 - Put file locations and metadata into the METS
 - Ingest the METS using the diringest servlet
 - Not sure how to connect the new item to a collection object. Supposedly this is done using the crules.xml. There is probably not a way unless the parent is ingested with the child objects.
 - See the ingested item here: <http://bl-ldlp-mz.ads.iu.edu:8080/fedora/get/demo:11>
- **A photo item from the hoh coll**
 - Similar to above:
 - Create a collection under the Lilly Community: Hohenberger Photographs
 - Follow the procedure above
 - make sure the image file name does not contain more than 1 dot
- **A paged object**
 - Fez doesn't seem to have a hierarchical model for item storage except Community-Collection.
 - Fez uses a complex and customizable method for defining new content models. Each content model is defined in two XSDs. One is for storing the metadata and one for displaying the metadata fields to the user on the UI.
 - It might be possible to define a new content model for paged documents.
 - It looks like the only way for ingesting page objects is defining a paged object content model and attaching pages individually to items of the paged content model. So, it's more like a compound object model with multiple datastreams where each datastream corresponds to a page.