# Focus on Your Content, Not on Ingesting Your Content

Terrence Brady, Applications Programmer Analyst, Georgetown University, twb27@georgetown.edu

Preparing a large collection for ingest into a repository should conceptually mirror the process for adding an individual item into a repository just at a larger scale. In reality, the process demands a unique workflow.

Web-based item submission processes are excellent for a single item or a couple of times, however it becomes laborious if you were to ingest more than 20 items. So what happens if you wanted to ingest 500 items or even 1000 items?

When processing a large collection at one time, there can be a significant amount of metadata that is reusable from item to item; or there can be a significant amount of metadata that is reusable after minor modification. Rather than being an exercise in metadata authoring, the process quickly becomes tedious and demands some form of automation.

Repository bulk ingest processes exist to alleviate some of the tedious steps in such a process. These processes require the assembly of a collection of items into some form of an ingest folder structure that contains item content to load and metadata associated with each item. Creating an ingest folder manually with tools such as a text editor is likely to either fail or to require significant re-work. Fortunately, the creation of an ingest folder structure lends itself well to automation.

One must possess a unique skill set to automate the creation of an ingest folder structure successfully. In order to create a collection of ingest folders on a file system, one must have a good knowledge of working on a file server and working data transformation tools. These processes can become unwieldy or in cases even impossible for librarians and other staff without having appropriate server level training or tools to aid in the process. As a result, this approach leads one to focus on *how* the content will be processed rather than on the quality of the metadata that needs to be authored.

At Georgetown University, when we first started with DSpace, our librarians and staff used the Bulk Item Import Tool provided by DSpace which requires interaction at the server level and a disproportionate amount of energy on managing and planning for the ingestion process. Our goal was to channel this energy towards improving metadata quality and marketing the use of the DigitalGeorgetown repository, which led us to develop and deploy successfully a collection of “Automated Ingest Tools”. These tools have been adopted by 3 of our libraries and has greatly reduced the time, energy and troubleshooting that the traditional bulk ingest process used to require.

This presentation will describe a series of simple applications that were combined to eliminate the tedious and error-prone steps within the bulk ingestion process for DigitalGeorgetown. It will also describe the challenges that were encountered with the bulk ingestion process and the tools that we created to address these issues. There will also be a segment dedicated to the training that was provided and the lessons learned in deploying these applications. Lastly, this presentation will also provide an overview of the applications that the Georgetown University Libraries have made available for use by other institutions.

## Challenges

### Assembling item ingest packages on the file system

In order to prepare an item ingest package, a number of metadata files must be assembled according to precise rules. The names of digital files associated with a set of metadata must precisely match the name of a file on the file system. When executing a bulk ingestion task, item specific failures will occur if some mismatches occur between files and metadata. The user is forced to decide whether it is easier to re-ingest an entire batch or to selectively ingest individual items that failed.

Within an item ingest package, metadata fragments are dispersed across many files. Each of these files is very simple in structure and easy to understand, but the process of applying simple changes to a large number of files is particularly error prone. It is particularly frustrating to iteratively discover small errors that needed to be re-applied to each individual file (i.e. simple well-formedness errors in files).

Our solution was to create a desktop application that a user to could interactively use to generate all of the necessary metadata components. This application also automated a number of validation steps to catch common errors that would trigger failures during the ingestion process.

This presentation will discuss the DSpace-specific components of an item ingest folder. While some of the details would differ in another repository, the problems that we overcame would translate to other repository ingest processes.

#### Contents file creation

The “contents file” of an ingest package requires an exact, case-sensitive match on file names in order to properly identify original media. While mistyped file names were fairly easy to troubleshoot, errors of this type occurred with some frequency. The contents file is a tab-separated file. Users manually creating contents files in a text editor often used spaces in place of tab characters. These errors were often a bit more confusing to identify.

The first tool that we introduced is an  **Ingest Inventory** tool. This application runs on the user’s desktop and scans their file system for candidate files to ingest. The tool creates an **Ingest Inventory Spreadsheet** containing the exact file name for each item to be ingested. The tool also identified the presence of thumbnail files conforming to DSpace naming standards. The file names in this spreadsheet serve as an index key to the subsequent tools that will be run. The user is able to indicate the specific DSpace metadata field names that will be populated for each item and to identify item-specific license files to be uploaded.

#### Associating metadata with a specific item

Inevitably, in each of our bulk ingest processes, users started with a spreadsheet of metadata that needed to be associated with each item on ingest. While a spreadsheet is a great tool for authoring this information, the process of transferring this metadata to the ingest folder structure was a tedious process. By utilizing a standard, Ingest Inventory Spreadsheet we were able to streamline the subsequent phases of creating the ingest folder structure.

#### Generating the Ingest Folder structure

The DSpace Item Import tool makes use of an ingest folder structure that is simple to understand. When processing a large number of files, the creation of ingest folders is tedious and error prone. The next tool that we introduced was an **Ingest Folder Creation Tool**. Using the Ingest Inventory Spreadsheet as input, this application built a flat hierarchy of ingest folders -- one per item. The tool provides the optional ability for a user to designate a boilerplate license file and thumbnail file to associate with each item. This tool assembles the original file, license file and thumbnail file into the ingest folder structure. Each of these items is cataloged within the contents file. A Dublin Core metadata file is generated for each item using the metadata from the Ingest Inventory Spreadsheet. A summary report is generated to provide an overview of the ingest folder structure that was created.

#### Validating the Ingest Folder Structure

After creating the ingest folder structure, a user might need to tweak the metadata for specific items or gather custom resources such as license files and custom thumbnail files. The **Ingest Folder Validation** tool checks the contents of each ingest folder and validates the metadata to be ingested. We created custom validation rules to enforce our institution’s minimum metadata requirements. This permits the user to use the bulk editing process to fix metadata within a collection rather than fixing metadata item by item after ingest.

#### Invoking the Bulk Ingest Process

Before creating our bulk ingestion tools, users needed file transfer access and shell access to our DSpace servers in order to invoke the item import process. The DSpace Item Import tool is a powerful tool with many command line operations.

We developed a web-based **Bulk Ingest Invocation Tool** that ingests content from a staging file system. Users of this process have been given restricted access to our staging area rather than shell access to our entire server. This application invokes the Item Import tool on behalf of the user prompting the user only for the location of the staging folder and the name of the collection into which the item will be imported. Once an item has been successfully ingested, we automatically trigger the DSpace media filter process on the containing collection so that derivative media (such as thumbnail images) are available to the user as soon as possible. The user is also presented with a link that will allow them to undo a bulk ingestion if an error has occurred.

One byproduct of this work was the creation of a web-based Filter Media Invocation Tool. Our users have the ability to trigger the generation of derivative media within a collection of their choosing after manually submitting items via the web-based item submission process.

### User Training

We developed a narrative collection of screenshots to walk a user through the bulk ingest process. In addition, we provided one-on-one training to each user the first time that they made use of the process. In addition, we have provided content owners with access to a test server that allows them to ingest and preview their collections before deploying the collections to our live instance. This small amount of training has empowered users to take ownership of their ingestion processes.

After introducing these tools, a number of new collections were introduced across 3 of our libraries. These tools enabled a backlog of existing collections to be quickly processed.

## Lessons Learned

These simple automation tools have provided a number of benefits for our librarians and staff working with digital collections. Working with the repository has become much more interesting and engaging for our librarians since the most tedious steps associated with content ingestion have been automated. Librarians are able to perform thorough quality control checks much more quickly than they could have before. Our librarians are also more comfortable and confident with the repository and have been freed to more aggressively market the services of the repository to the constituents.

The conversations between our collection owners and developers have switched from *how* do I ingest this collection to a more detailed conversation about *what* we can do with the individual metadata fields within the repository.

## Our Next Steps

We intend to extend our Ingest Folder Creation Tool to automate the creation of ingestion packages from other services. We would like to automatically prepare the Georgetown electronic theses and dissertations that we receive from ProQuest into DSpace ingestion folders. We intend to extend our tools to generate MARC records from DSpace export packages so that we can add information about our repository collections within our library catalog.

The presentation will provide an update on these initiatives.

## Try it Yourself!

The Georgetown University Libraries have made the desktop-based applications described in the presentation available on [GitHub](https://github.com/organizations/Georgetown-University-Libraries) for use by other institutions.

*Note: the desktop tools extend the “File Analyzer and Metadata Harvester” application that was originally developed by National Archives and Records Administration staff. This application can serve many other purposes in a digitization workflow.*