Importing and Exporting Items via Simple Archive Format

Item Importer and Exporter

DSpace has a set of command line tools for importing and exporting items in batches, using the DSpace Simple Archive Format. Apart from the offered functionality, these tools serve as an example for users who aim to implement their own item importer.

DSpace Simple Archive Format

The basic concept behind the DSpace Simple Archive Format is to create an archive, which is a directory containing one subdirectory per item. Each item directory contains a file for the item's descriptive metadata, and the files that make up the item.

```
archive_directory/
    item_000/
        dublin_core.xml         -- qualified Dublin Core metadata for
metadata fields belonging to the dc schema
        metadata_[prefix].xml   -- metadata in another schema, the prefix
is the name of the schema as registered with the metadata registry
        contents                -- text file containing one line per
filename
                collections                             -- text file
that contains the handles of the collections the item will belong two.
Optional. Each handle in a row.
                                                             --
Collection in first line will be the owning collection
        file_1.doc              -- files to be added as bitstreams to the
item
        file_2.pdf
    item_001/
        dublin_core.xml
        contents
        file_1.png
        ...
```

The `dublin_core.xml` or `metadata_[prefix].xml` file has the following format, where each metadata element has it's own entry within a `<dcvalue>` tagset. There are currently three tag attributes available in the `<dcvalue>` tagset:

- `<element>` - the Dublin Core element
- `<qualifier>` - the element's qualifier
- `<language>`- (optional)ISO language code for element

```
<dublin_core>
    <dcvalue element="title" qualifier="none">A Tale of Two Cities<
/dcvalue>
    <dcvalue element="date" qualifier="issued">1990</dcvalue>
    <dcvalue element="title" qualifier="alternative" language="fr"
>J'aime les Printemps</dcvalue>
</dublin_core>
```

(Note the optional language tag attribute which notifies the system that the optional title is in French.)

Every metadata field used, must be registered via the metadata registry of the DSpace instance first, see Metadata and Bitstream Format Registries.

Recommended Metadata

It is recommended to minimally provide "dc.title" and, where applicable, "dc.date.issued". Obviously you can (and should) provide much more detailed metadata about the Item. For more information see: Metadata Recommendations.

The `contents` file simply enumerates, one file per line, the bitstream file names. See the following example:

```
file_1.doc
file_2.pdf
license
```

Please notice that the `license` is optional, and if you wish to have one included, you can place the file in the .../item_001/ directory, for example.

The bitstream name may optionally be followed by any of the following:

- `\tbundle:BUNDLENAME`
- `\tpermissions:PERMISSIONS`
- `\tdescription:DESCRIPTION`
- `\tprimary:true`

Where '\t' is the tab character.

'BUNDLENAME' is the name of the bundle to which the bitstream should be added. Without specifying the bundle, items will go into the default bundle, ORIGINAL.

'PERMISSIONS' is text with the following format: `-[r|w] 'group name'`

'DESCRIPTION' is text of the files description.

Primary is used to specify the primary bitstream.

Configuring `metadata_[prefix].xml` for Different Schema

It is possible to use other Schema such as EAD, VRA Core, etc. Make sure you have defined the new scheme in the DSpace Metada Schema Registry.

1. Create a separate file for the other schema named `metadata_[prefix].xml`, where the `[prefix]` is replaced with the schema's prefix.
2. Inside the xml file use the same Dublin Core *syntax*, but on the `<dublin_core>` element include the attribute `schema=[prefix]`.
3. Here is an example for ETD metadata, which would be in the file `metadata_etd.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<dublin_core schema="etd">
     <dcvalue element="degree" qualifier="department">Computer
Science</dcvalue>
     <dcvalue element="degree" qualifier="level">Masters</dcvalue>
     <dcvalue element="degree" qualifier="grantor">Michigan Institute
of Technology</dcvalue>
</dublin_core>
```

Importing Items

Before running the item importer over items previously exported from a DSpace instance, please first refer to Transferring Items Between DSpace Instances.

Command used:	`[dspace]/bin/dspace import`
Java class:	`org.dspace.app.itemimport.ItemImport`
Arguments short and (long) forms:	Description
`-a or --add`	Add items to DSpace ‡
`-r or --replace`	Replace items listed in mapfile ‡
`-d or --delete`	Delete items listed in mapfile ‡
`-s or --source`	Source of the items (directory)
`-c or --collection`	Destination Collection by their Handle or database ID
`-m or --mapfile`	Where the mapfile for items can be found (name and directory)
`-e or --eperson`	Email of eperson doing the importing
`-w or --workflow`	Send submission through collection's workflow
`-n or --notify`	Kicks off the email alerting of the item(s) has(have) been imported
`-t or --test`	Test run, do not actually import items
`-p or --template`	Apply the collection template
`-R or --resume`	Resume a failed import (Used on Add only)
`-h or --help`	Command help
`-z or --zip`	Name of zipfile

‡ These are mutually exclusive.

The item importer is able to batch import unlimited numbers of items for a particular collection using a very simple CLI command and 'arguments'

Adding Items to a Collection from a directory

To add items to a collection, you gather the following information:

- eperson
- Collection ID (either Handle (e.g. 123456789/14) or Database ID (e.g. 2)
- Source directory where the items reside
- Mapfile. Since you don't have one, you need to determine where it will be (e.g. /Import/Col_14/mapfile)
  At the command line:

```
[dspace]/bin/dspace import --add --eperson=joe@user.com --
collection=CollectionID --source=items_dir --mapfile=mapfile
```

or by using the short form:

```
[dspace]/bin/dspace import -a -e joe@user.com -c CollectionID -s items_dir
-m mapfile
```

The above command would cycle through the archive directory's items, import them, and then generate a map file which stores the mapping of item directories to item handles. **SAVE THIS MAP FILE.** Using the map file you can use it for replacing or deleting (unimporting) the file.

Testing. You can add `--test` (or `-t`) to the command to simulate the entire import process without actually doing the import. This is extremely useful for verifying your import files before doing the actual import.

Adding Items to a Collection from a zipfile

To add items to a collection, you gather the following information:

- eperson
- Collection ID (either Handle (e.g. 123456789/14) or Database ID (e.g. 2)
- Source directory where your zipfile containing the items resides
- Zipfile
- Mapfile. Since you don't have one, you need to determine where it will be (e.g. /Import/Col_14/mapfile)
  At the command line:

```
[dspace]/bin/dspace import --add --eperson=joe@user.com --
collection=CollectionID --source=items_dir --zip=filename.zip --
mapfile=mapfile
```

or by using the short form:

```
[dspace]/bin/dspace import -a -e joe@user.com -c CollectionID -s items_dir
-z filename.zip -m mapfile
```

The above command would unpack the zipfile, cycle through the archive directory's items, import them, and then generate a map file which stores the mapping of item directories to item handles. **SAVE THIS MAP FILE.** Using the map file you can use it for replacing or deleting (unimporting) the file.

Testing. You can add `--test` (or `-t`) to the command to simulate the entire import process without actually doing the import. This is extremely useful for verifying your import files before doing the actual import.

Replacing Items in Collection

Replacing existing items is relatively easy. Remember that mapfile you saved above? Now you will use it. The command (in short form):

```
[dspace]/bin/dspace import -r -e joe@user.com -c collectionID -s items_dir
-m mapfile
```

Long form:

```
[dspace]/bin/dspace import --replace --eperson=joe@user.com --
collection=collectionID --source=items_dire --mapfile=mapfile
```

Deleting or Unimporting Items in a Collection

You are able to unimport or delete items provided you have the mapfile. Remember that mapfile you saved above? The command is (in short form):

```
[dspace]/bin/dspace import -e joe@user.com -d -m mapfile
```

In long form:

```
[dspace]/bin/dspace import --eperson=joe@user.com --delete --mapfile
mapfile
```

Other Options

- **Workflow**. The importer usually bypasses any workflow assigned to a collection. But add the `--workflow` (`-w`) argument will route the imported items through the workflow system.
- **Templates**. If you have templates that have constant data and you wish to apply that data during batch importing, add the `--template` (`-p`) argument.
- **Resume**. If, during importing, you have an error and the import is aborted, you can use the `--resume` (`-R`) flag that you can try to resume the import where you left off after you fix the error.
- **Specifying the owning collection on a per-item basis from the command line administration tool**

  If you omit the -c flag, which is otherwise mandatory, the ItemImporter searches for a file named "collections" in each item directory. This file should contain a list of collections, one per line, specified either by their handle, or by their internal db id. The ItemImporter then will put the item in each of the specified collections. The owning collection is the collection specified in the first line of the collections file.

  If both the -c flag is specified and the collections file exists in the item directory, the ItemImporter will ignore the collections file and will put the item in the collection specified on the command line.

  Since the collections file can differ between item directories, this gives you more fine-grained control of the process of batch adding items to collections.
- **Importing with BTE**

  The DSpaceOutputGenerator, which writes the metadata into the DSpace Simple Archive Format, has been updated to produce the collections file, if a metadata field named "collections" (reserved word) exists in the original metadata. This is mainly applicable to the CSV input format which is more flexible, but could also be implemented with a Modifier that adds the "collections" field to each Record in the BTE pipeline.

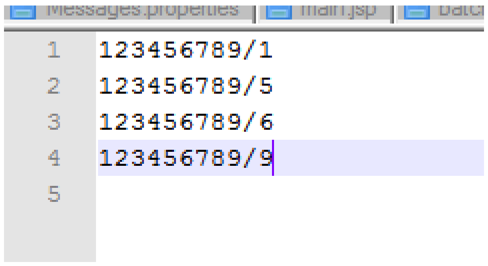
  Important note: an entry with the "collections" key should be in the output map that is used by the DSpaceOutputGenerator.
  More info in Importing Items via basic bibliographic formats (Endnote, BibTex, RIS, TSV, CSV) and online services (OAI, arXiv, PubMed, CrossRef, CiNii).

UI Batch Import (JSPUI)

Batch import can also take place via the Administrator's UI. The steps to follow are:

A. Prepare the data

1. Items, i.e. the metadata and their bitstreams, must be in the Simple Archive Format describer earlier in this chapter. Thus, for each item there must be a separate directory that contains the corresponding files of the specific item.
2. Moreover, in each item directory, there can be another file that describes the collection or the collections that this item will be added to. The name of this file must be "collections" and it is optional. It has the following format:



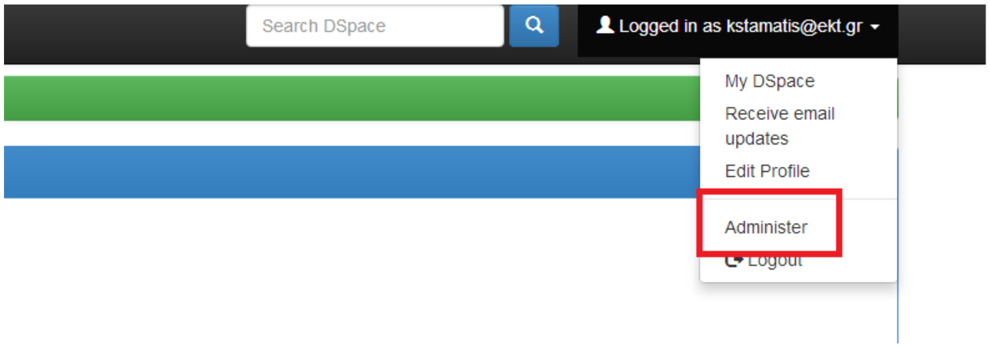
```
123456789/1
123456789/5
123456789/6
123456789/9
```

Each line contains the handle of the collection. The collection in the first line is the owning collection while the rest are the other collection the item should belong to.

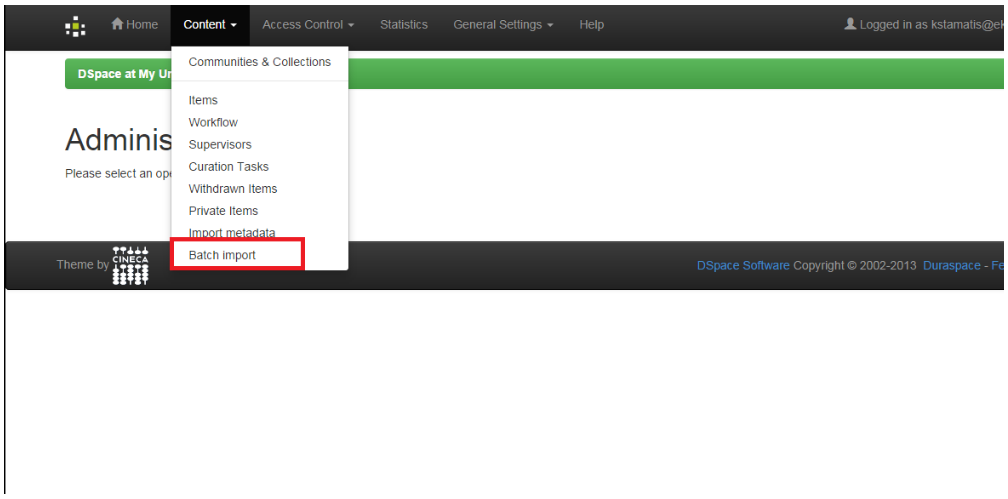
3. Compress the item directories into a zip file. Please note that you need to zip the actual item directories and not just the directory that contains the item directories. Thus, the final zip file must directly contain the item directories.
4. Place the zip file in a public domain URL, like Dropbox or Google Drive or wherever you have access to do so. Since such a zip file can be very big in size, the batch import UI needs the URL to download it for a public location rather than just upload it and get a timeout exception

B. Import the items via the UI

1. Login as an administrator
2. Find the menu on the top right of the page, and select the "Administer" option

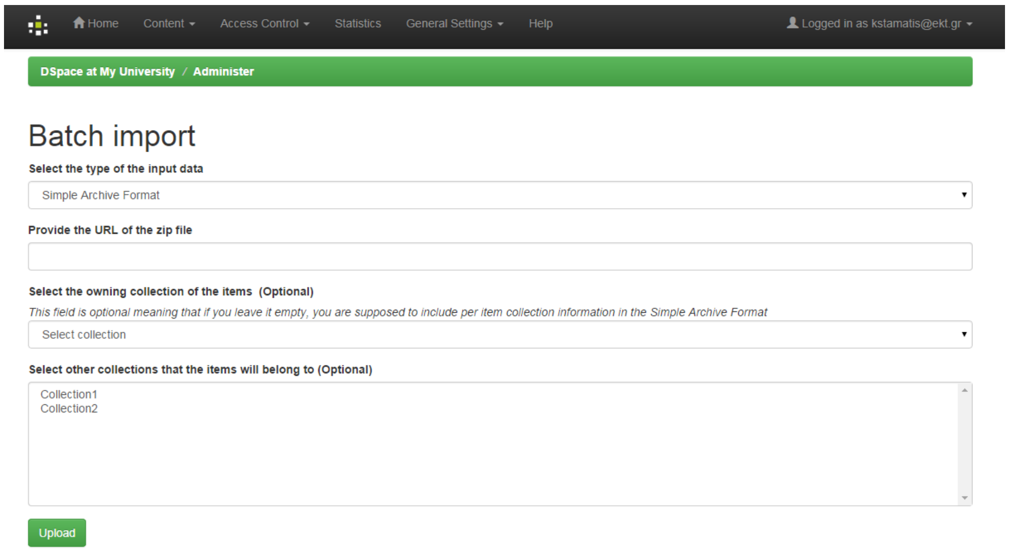


3. Select the "Batch Import" option from the "Content" drop down menu on the top of the page



4. Fill in the form that appears as follows:

- Field #1: select the type of the input data that you want to batch import. Be sure to select "Simple Archive Format" in this drop down menu
- Field #2: Copy/Paste the public URL where the zip file mentioned earlier is located
- Filed #3: Select the owning collection of the items you are importing. This field is optional meaning that if you leave it empty, you are supposed to include per item collection information (via the "collections" file mentioned before) in the Simple Archive Format
- Field #4: Select the other collections the item will belong to. You can select more than one collection by just holding down the Ctrl key on your keyboard. If you select the owning collection in this multiselect input control, it will be ignored at the very end.


Home
Content
Access Control
Statistics
General Settings
Help
Logged in as kstamatis@ekt.gr
DSpace at My University / Administer

Batch import

Select the type of the input data

Simple Archive Format

Provide the URL of the zip file

Select the owning collection of the items (Optional)

This field is optional meaning that if you leave it empty, you are supposed to include per item collection information in the Simple Archive Format

Select collection

Select other collections that the items will belong to (Optional)

Collection1
Collection2

Upload

<u>Comments:</u>

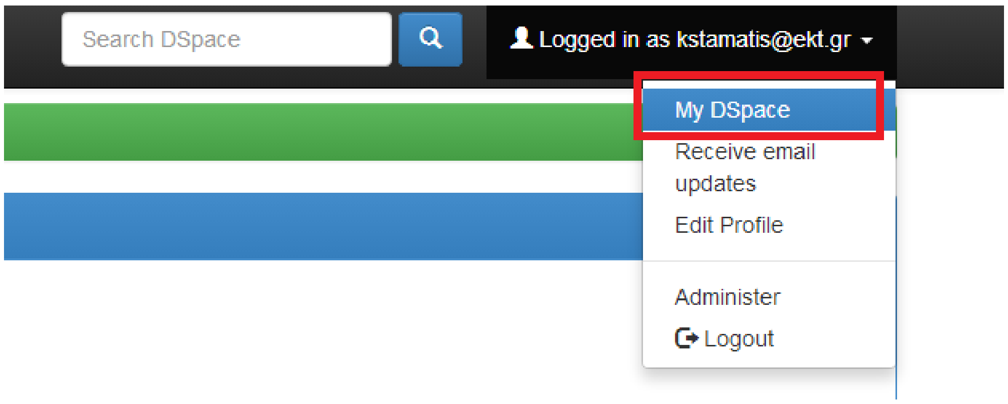
1) If you select an owning collection from this form, then the "collections" file that may be included in the item will be ignored.

2) If you do not specify an owning collection, and for some items no "collections" file exists in the item directory, then the item will not be imported in DSpace

Finally, when you submit the form you will receive a message informing you that the import process is being executed in the background (si nce it may take long). At the end, you will receive a success or failure email (to the email address of your DSpace account) informing you of the status of the import.

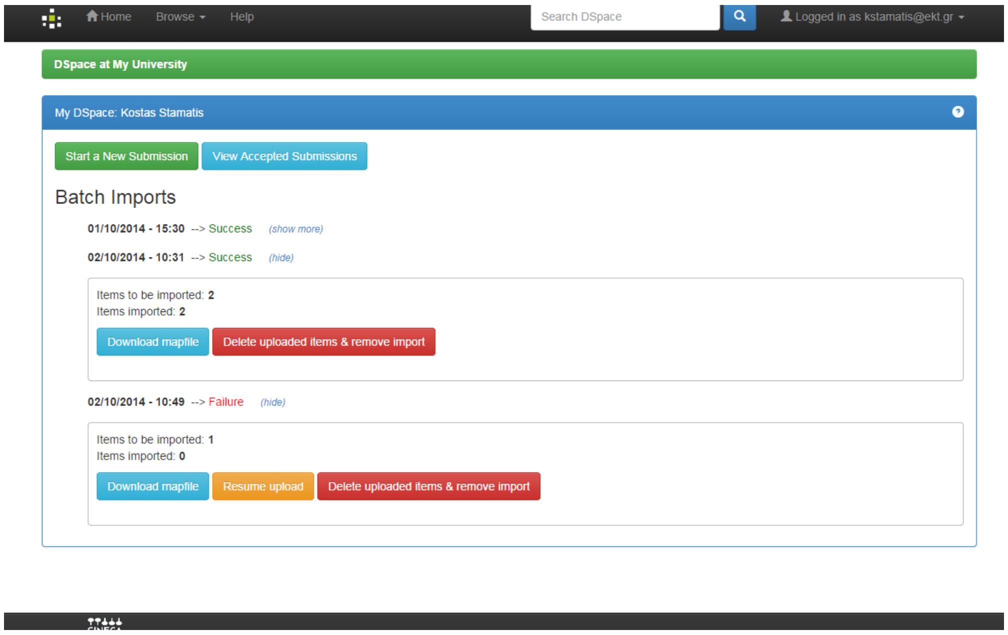
C. View past batch imports (that have be done via the UI)

1. Login
2. Visit "My DSpace" page



3. On the next page, you can see the history of batch imports. For each import, the following information is available:

The status of the batch import (success or failure)
The number of items that the user tried to import
The number of items that were actually imported


Home
Browse
Help
Search DSpace
Logged in as kstamatis@ekt.gr
DSpace at My University
My DSpace: Kostas Stamatis
Start a New Submission
View Accepted Submissions

Batch Imports

01/10/2014 - 15:30 --> Success (show more)

02/10/2014 - 10:31 --> Success (hide)

Items to be imported: **2**
Items imported: **2**

Download mapfile
Delete uploaded items & remove import

02/10/2014 - 10:49 --> Failure (hide)

Items to be imported: **1**
Items imported: **0**

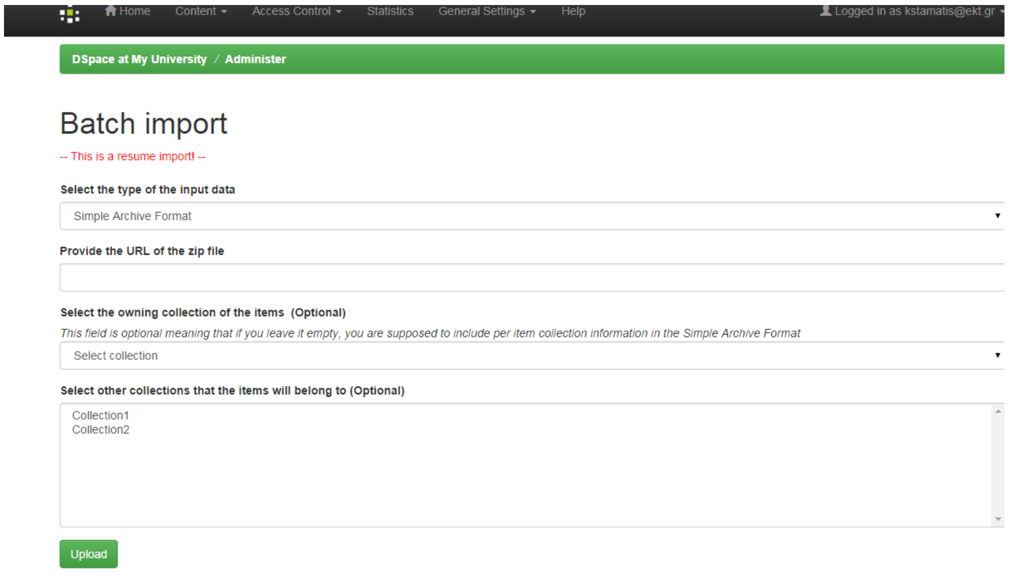
Download mapfile
Resume upload
Delete uploaded items & remove import

Moreover, the user can take the following actions:

Download the map file that was produced during the import. This file contains a list of items that were imported with the corresponding handle assigned to them by DSpace.

Delete the imported items. Everything that was imported will be deleted (including the history directory in the "[dspace]/import" directory)

In case of failure, the user can "Resume" the import. The user is taken to the upload form again, but the system recognizes the initial import (and the map file) in order to resume the old import. There is a red label in the form that informs the user about the "Resume" form.


Home
Content
Access Control
Statistics
General Settings
Help
Logged in as kstamatis@ekt.gr
DSpace at My University / Administer

Batch import

-- This is a resume import! --

Select the type of the input data
Simple Archive Format

Provide the URL of the zip file

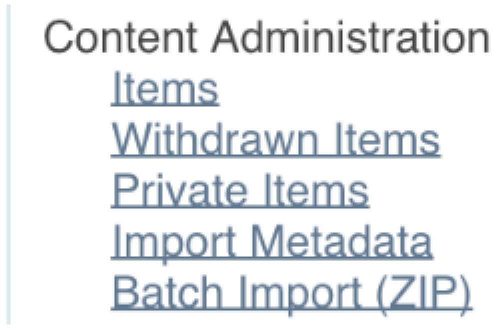
Select the owning collection of the items (Optional)
This field is optional meaning that if you leave it empty, you are supposed to include per item collection information in the Simple Archive Format
Select collection

Select other collections that the items will belong to (Optional)
Collection1
Collection2

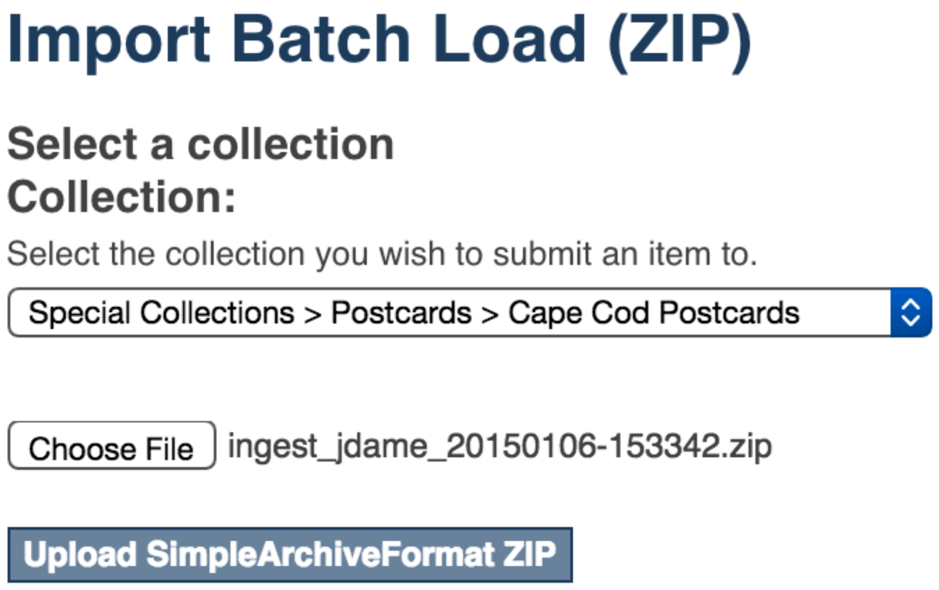
UI Batch Import (XMLUI)

A SimpleArchiveFormat package can be imported by an administrator in XMLUI. The SimpleArchiveFormat package needs to be compressed into a ZIP file, and it be will be uploaded to XMLUI through the browser. DSpace will then process that ZIP, and ingest items into DSpace. A stable network connection is recommended, as your browser will need to upload a potentially large ZIP file, and then wait while DSpace processes that ZIP file.

While logged in as an administrator, click on Batch Import (ZIP):



Then, choose the owning collection from the collection dropdown, and browse to the ZIP file on your computer that has the SimpleArchiveFormat ZIP file.

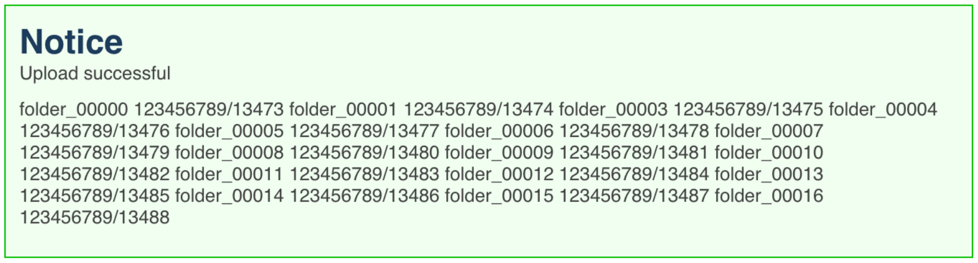


Select the collection you wish to submit an item to.

Special Collections > Postcards > Cape Cod Postcards

ingest_jdame_20150106-153342.zip

If successful, you will get a green message with a list of handles that were imported. It is what is considered the "map file".

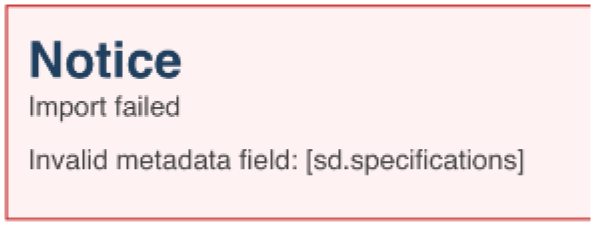


Notice

Upload successful

```
folder_00000 123456789/13473 folder_00001 123456789/13474 folder_00003 123456789/13475 folder_00004
123456789/13476 folder_00005 123456789/13477 folder_00006 123456789/13478 folder_00007
123456789/13479 folder_00008 123456789/13480 folder_00009 123456789/13481 folder_00010
123456789/13482 folder_00011 123456789/13483 folder_00012 123456789/13484 folder_00013
123456789/13485 folder_00014 123456789/13486 folder_00015 123456789/13487 folder_00016
123456789/13488
```

If an error occurred, you will get a red error message with the issue:



Notice

Import failed

Invalid metadata field: [sd.specifications]

Exporting Items

The item exporter can export a single item or a collection of items, and creates a DSpace simple archive in the aforementioned format for each exported item. The items are exported in a sequential order in which they are retrieved from the database. As a consequence, the sequence numbers of the item subdirectories (item_000, item_001) are not related to DSpace handle or item ids.

Command used:	`[dspace]/bin/dspace export`
Java class:	*org.dspace.app.itemexport.ItemExport*
Arguments short and (long) forms:	Description
`-t` or `--type`	Type of export. *COLLECTION* will inform the program you want the whole collection. *ITEM* will be only the specific item. (You will actually key in the keywords in all caps. See examples below.)
`-i` or `--id`	The ID or Handle of the Collection or Item to export.
`-d` or `--dest`	The destination path where you want the file of items to be placed.
`-n` or `--number`	Sequence number to begin export the items with. Whatever number you give, this will be the name of the first directory created for your export. The layout of the export directory is the same as the layout used for import.
`-m` or `--migrate`	Export the item/collection for migration. This will remove the handle and metadata that will be re-created in the new instance of DSpace.
`-h` or `--help`	Brief Help.

Exporting a Collection

The CLI command to export the items of a collection:

```
[dspace]/bin/dspace export --type=COLLECTION --id=collectionID_or_handle --
dest=/path/to/destination --number=seq_num
```

Short form:

```
[dspace]/bin/dspace export -t COLLECTION -i collectionID_or_handle -d /path
/to/destination -n seq_num
```

Exporting a Single Item

The keyword *COLLECTION* means that you intend to export an entire collection. The ID can either be the database ID or the handle. The exporter will begin numbering the simple archives with the sequence number that you supply. To export a single item use the keyword *ITEM* and give the item ID as an argument:

```
[dspace]/bin/dspace export --type=ITEM --id=itemID_or_handle --dest=/path
/to/destination --number=seq_num
```

Short form:

```
[dspace]/bin/dspace export -t ITEM -i itemID_or_handle -d /path/to
/destination -n seq_num
```

Each exported item will have an additional file in its directory, named "handle". This will contain the handle that was assigned to the item, and this file will be read by the importer so that items exported and then imported to another machine will retain the item's original handle.

The `-m` Argument

Using the `-m` argument will export the item/collection and also perform the migration step. It will perform the same process that the next section Exchanging Content Between Repositories performs. We recommend that section to be read in conjunction with this flag being used.