Batch Metadata Editing

- 1 Batch Metadata Editing Tool
    - 1.1 Export Function
        - 1.1.1 Web Interface Export
        - 1.1.2 Command Line Export
    - 1.2 Import Function
        - 1.2.1 Web Interface Import
        - 1.2.2 Command Line Import
    - 1.3 CSV Format
        - 1.3.1 File Structure
    - 1.4 Editing the CSV
        - 1.4.1 Editing Collection Membership
        - 1.4.2 Adding Metadata-Only Items
        - 1.4.3 Deleting Metadata
        - 1.4.4 Performing 'actions' on items
        - 1.4.5 Migrating Data or Exchanging data
        - 1.4.6 Common Issues
            - 1.4.6.1 Metadata values in CSV export seem to have duplicate columns
            - 1.4.6.2 DSpace responds with "No changes were detected" when CSV is uploaded
    - 1.5 Batch Editing, Entities and Relationships
        - 1.5.1 Background about entities and virtual metadata
        - 1.5.2 Admin CSV export
        - 1.5.3 Admin CSV import

Batch Metadata Editing Tool

DSpace provides a batch metadata editing tool. The batch editing tool is able to produce a comma delimited file in the CSV format. The batch editing tool facilitates the user to perform the following:

- Batch editing of metadata (e.g. perform an external spell check)
- Batch additions of metadata (e.g. add an abstract to a set of items, add controlled vocabulary such as LCSH)
- Batch find and replace of metadata values (e.g. correct misspelled surname across several records)
- Mass move items between collections
- Mass deletion, withdrawal, or re-instatement of items
- Enable the batch addition of new items (without bitstreams) via a CSV file
- Re-order the values in a list (e.g. authors)

For information about configuration options for the Batch Metadata Editing tool, see Batch Metadata Editing Configuration

Export Function

Web Interface Export

Batch metadata exports (to CSV) can be performed from the Administrative menu:

- Login as an Administrative user
- In Sidebase, select "Export" "Metadata". Type in the Community/Collection name.
    - Alternatively, browse to the Community or Collection you wish to export, and then go to "Export" "Metadata". That Community /Collection will be preselected.
- Click "Export". A new Process will be created (in "Processes" menu). Once completed, download the resulting CSV.

Exporting search results to CSV was not added until DSpace 7.3

As of DSpace 7.3, it is possible to Export search results to a CSV *(similar to 6.x).* When logged in as an Administrator, after performing a search a new "Export search results as CSV" button appears. Clicking it will export the metadata of all items in your search results to a CSV. This CSV can then be used to perform batch metadata updates (based on the items in your search results). **- Release Notes#7.3ReleaseNotes**

Please see below documentation for more information on the CSV format and actions that can be performed by editing the CSV.

Command Line Export

The following table summarizes the basics.

Command used:	`[dspace]/bin/dspace metadata-export`
Java class:	org.dspace.app.bulkedit.MetadataExport
Arguments short and (long) forms):	Description
`-f` or `--file`	Required. The filename of the resulting CSV.
`-i` or `--id`	The Item, Collection, or Community handle or Database ID to export. If not specified, **all** items will be exported.

`-a` or `--all`	Include all the metadata fields that are not normally changed (e.g. provenance) or those fields you configured in the `[dspace]/config/modules/bulkedit.cfg` to be ignored on export.
`-h` or `--help`	Display the help page.

To run the batch editing exporter, at the command line:

```
[dspace]/bin/dspace metadata-export -f name_of_file.csv -i 1023/24
```

Example:

```
[dspace]/bin/dspace metadata-export -f /batch_export/col_14.csv -i /1989.1/24
```

In the above example we have requested that a collection, assigned handle '*1989.1/24*' export the entire collection to the file '*col_14.csv*' found in the '*/batch_export*' directory.

Please see below documentation for more information on the CSV format and actions that can be performed by editing the CSV .

Import Function

Importing large CSV files

It is not recommended to import CSV files of more than 1,000 lines (i.e. 1,000 items). When importing files larger than this, it may be difficult for an Administrator to accurately verify the changes that the import tool states it will make. In addition, depending on the memory available to the DSpace site, large files may cause 'Out Of Memory' errors part way through the import process.

Web Interface Import

Batch metadata imports (from CSV) can be performed from the Administrative menu:

- First, complete all editing of the CSV and save your changes
- Login as an Administrative User
- In sidebar, select "Import" "Metadata" and drag & drop the CSV file

Validate a Batch Metadata CSV was not added until DSpace 7.3

As of DSpace 7.3, it is now possible to *validate* a Batch Metadata CSV before applying changes *(similar to 6.x)*. When uploading a CSV for batch updates (using "Import" menu), a new "Validate Only" option is selected by default. When selected, the uploaded CSV will only be validated & you'll receive a report of the detected changes in the CSV. This allows you to verify the changes are correct before applying them. (NOTE: applying the changes requires re-submitting the CSV with the "Validate Only" option deselected) **- Release Notes#7.3ReleaseNotes**

Command Line Import

The following table summarizes the basics.

Command used:	`[dspace]/bin/dspace metadata-import`
Java class:	org.dspace.app.bulkedit.MetadataImport
Arguments short and (long) forms:	Description
`-f` or `--file`	Required. The filename of the CSV file to load.
`-s` or `--silent`	Silent mode. The import function does not prompt you to make sure you wish to make the changes.
`-e` or `--email`	The email address of the user. This is only required when adding new items.
`-w` or `--workflow`	When adding new items, the program will queue the items up to use the Collection Workflow processes.
`-n` or `--notify`	when adding new items using a workflow, send notification emails.
`-t` or `--template`	When adding new items, use the Collection template, if it exists.
`-h` or `--help`	Display the brief help page.

Silent Mode should be used carefully. It is possible (and probable) that you can overlay the wrong data and cause irreparable damage to the database.

To run the batch importer, at the command line:

```
[dspace]/bin/dspace metadata-import -f name_of_file.csv
```

Example

```
[dspace]/bin/dspace metadata-import -f /dImport/col_14.csv
```

If you are wishing to upload new metadata **without** bitstreams, at the command line:

```
[dspace]/bin/dspace metadata-import -f /dImport/new_file.csv -e joe@user.com -w -n -t
```

In the above example we threw in all the arguments. This would add the metadata and engage the workflow, notification, and templates to all be applied to the items that are being added.

CSV Format

The CSV (comma separated values) files that this tool can import and export abide by the RFC4180 CSV format. This means that new lines, and embedded commas can be included by wrapping elements in double quotes. Double quotes can be included by using two double quotes. The code does all this for you, and any good csv editor such as Excel or OpenOffice will comply with this convention.

All CSV files are also in UTF-8 encoding in order to support all languages.

File Structure

The first row of the CSV must define the metadata values that the rest of the CSV represents. ***The first column must always be "id" which refers to the item's internal database ID***. *All other columns are optional.* The other columns contain the dublin core metadata fields that the data is to reside.

A typical heading row looks like:

```
id,collection,dc.title,dc.contributor,dc.date.issued,etc,etc,etc.
```

Subsequent rows in the csv file relate to items. A typical row might look like:

```
350,2292,Item title,"Smith, John",2008
```

If you want to store multiple values for a given metadata element, they can be separated with the double-pipe '||' (or another character that you defined in your `modules/bulkedit.cfg` file). For example:

```
Horses||Dogs||Cats
```

Elements are stored in the database in the order that they appear in the CSV file. You can use this to order elements where order may matter, such as authors, or controlled vocabulary such as Library of Congress Subject Headings.

Editing the CSV

If you are editing with Microsoft Excel, be sure to open the CSV in Unicode/UTF-8 encoding

By default, Microsoft Excel may not correctly open the CSV in Unicode/UTF-8 encoding. This means that special characters may be improperly displayed and also can be "corrupted" during re-import of the CSV.

You need to tell Excel this CSV is Unicode, by importing it as follows. (*Please note these instructions are valid for MS Office 2007 and 2013. Other Office versions may vary*)

- First, open Excel (with an empty sheet/workbook open)
- Select "Data" tab
- Click "From Text" button (in the "External Data" section)
- Select your CSV file
- Wizard Step 1
  - Choose "Delimited" option
  - Start import at row: 1
  - In the "File origin" selectbox, select "65001 : Unicode (UTF-8)"
    - NOTE: these encoding options are sorted alphabetically, so "Unicode (UTF-8)" appears near the bottom of the list.
  - Click Next
- Wizard Step 2
  - Select "Comma" as the only delimiter
  - Click Next
- Wizard Step 3
  - Select "Text" as the "Column data format" (*Unfortunately, this must be done for each column individually in Excel*)
    - At a minimum, you MUST ensure all date columns (e.g. dc.date.issued) are treated as "Text" so that Excel doesn't autoconvert DSpace's YYYY-MM-DD format into MM/DD/YYYY
    - To avoid such autoconversion, it is safest to ensure each column is treated as "Text". Unfortunately, this means selecting each column one-by-one and choosing "Text" as the "Column data format".
  - Click Finish
- Choose whether to open CSV in the existing sheet or a new one

Tips to Simplify the Editing Process

When editing a CSV, here's a couple of basic tips to keep in mind:

1. The "id" column MUST remain intact. This column also must always have a value in it.
2. To simplify the CSV, you can simply remove any columns you do NOT wish to edit (except for "id" column, see #1). Don't worry, removing the entire column won't delete metadata (see #3)
3. When importing a CSV file, the importer will *overlay* the metadata onto what is already in the repository to determine the differences. It *only* acts on the contents of the CSV file, rather than on the complete item metadata. This means that the CSV file that is exported can be manipulated quite substantially before being re-imported. Rows (items) or Columns (metadata elements) can be removed and will be ignored.
   a. For example, if you only want to edit "dc.subject", you can remove ALL columns EXCEPT for "id" and "dc.subject" so that you can just manipulate the "dc.subject" field. On import, DSpace will see that you've only included the "dc.subject" field in your CSV and therefore will only update the "dc.subject" metadata field for any items listed in that CSV.
4. Because removing an entire column does NOT delete metadata value(s), if you actually wish to delete a metadata value you should leave the column intact, and simply clear out the appropriate row's value (in that column).

Editing Collection Membership

Items can be moved between collections by editing the collection handles in the 'collection' column. Multiple collections can be included. The first collection is the 'owning collection'. The owning collection is the primary collection that the item appears in. Subsequent collections (separated by the field separator) are treated as mapped collections. These are the same as using the map item functionality in the DSpace user interface. To move items between collections, or to edit which other collections they are mapped to, change the data in the collection column.

Adding Metadata-Only Items

New metadata-only items can be added to DSpace using the batch metadata importer. To do this, enter a plus sign '+' in the first 'id' column. The importer will then treat this as a new item. If you are using the command line importer, you will need to use the -e flag to specify the user email address or id of the user that is registered as submitting the items.

Deleting Metadata

It is possible to perform metadata deletes across the board of certain metadata fields from an exported file. For example, let's say you have used keywords (dc.subject) that need to be removed *en masse*. You would leave the column (dc.subject) intact, but remove the data in the corresponding rows.

Performing 'actions' on items

It is possible to perform certain 'actions' on items. This is achieved by adding an 'action' column to the CSV file (after the id, and collection columns). There are three possible actions:

1. ***'expunge'*** This permanently deletes an item. Use with care! This action must be enabled by setting 'allowexpunge = true' in `modules/bulkedit.cfg`
2. ***'withdraw'*** This withdraws an item from the archive, but does not delete it.
3. ***'reinstate'*** This reinstates an item that has previously been withdrawn.

If an action makes no change (for example, asking to withdraw an item that is already withdrawn) then, just like metadata that has not changed, this will be ignored.

Migrating Data or Exchanging data

It is possible that you have data in one Dublin Core (DC) element and you wish to really have it in another. An example would be that your staff have input Library of Congress Subject Headings in the Subject field (dc.subject) instead of the LCSH field (dc.subject.lcsh). Follow these steps and your data is migrated upon import:

1. Insert a new column. The first row should be the new metadata element. (We will refer to it as the TARGET)
2. Select the column/rows of the data you wish to change. (We will refer to it as the SOURCE)
3. Cut and paste this data into the new column (TARGET) you created in Step 1.
4. Leave the column (SOURCE) you just cut and pasted from empty. Do not delete it.

Common Issues

Metadata values in CSV export seem to have duplicate columns

DSpace responds with "No changes were detected" when CSV is uploaded

Unfortunately, this response may be caused in many ways

- It's possible the CSV was not saved properly after editing. Check that the edits are in the CSV, and that there were no backend errors in the DSpace logs (which would be an indication of an invalid or corrupted CSV file)
- Depending on the version of DSpace, you may be encountering this known bug with processing linebreaks in CSV files: https://github.com/DSpace/DSpace/issues/6600
- If you are setting a new embargo date in the CSV, ensure that the embargo lift date is a future date. It's been reported that past dates may cause DSpace to ignore item changes.

Batch Editing, Entities and Relationships

Consider the following page for this topic: Configurable Entities

Background about entities and virtual metadata

- In DSpace 7, we have entities. Entities are items with an entity type (there can still be items without an entity type).
- Two entities can be linked to each other. For this purpose relations are defined, which indicate the relationship between the entities. Relationships between two entities are defined by the metadata schema *relation*. The relation reflects how two entities are related to each other, for example *isPersonOfProject* or *isPublicationOfAuthor*.
- Until the introduction of entities, we could only link items to each other by inserting DOIs or URLs of other items into metadata fields *dc.relation.**. What is new about the linking between entities in DSpace 7 is that UUIDs are entered into the fields, i.e. internal identifiers of other entities that DSpace can easily resolve. DSpace "knows" which entities are linked to each other and how.
- On the item view of an entity (remember: an entity is an item with an entity type) metadata of other entities can be displayed. DSpace refers to this as virtual metadata. Virtual metadata does not belong to the item in whose item view it is displayed, but to a linked entity. They can be changed only in the linked entity. As an example: we have the entities journal and journal issue. All journal issues display the title of the journal in their item view. This title is stored only in the journal and is only (dynamically) displayed in the issues.

Admin CSV export

- Virtual metadata is exported with the entities in which it is included. For example, when you export projects, you see a column for the *project.investigator* field. Here, the names of two people have been included as virtual metadata. However, the names are not stored in the project, but exported from the respective person entities at the time of export. The specification *::virtual::* marks this. The specifications *::8585::* and *::27946::* are examples for this documentation and represent IDs of the relations. The specification *::600* comes from the DSpace Authority, which is also specified due to technical circumstances.
- The relation itself is also included in the CSV export, in the *relation.isPersonOfProject* field. Additionally, a *relation.isPersonOfProject.latestForDiscovery* field is created. This field has internal reasons in DSpace and should help to make things faster discoverable. In the fields you again see the *::virtual::8585::600* specification, which are already explained above. Instead of the values of individual metadata fields, you now have the UUIDs of the items that are linked. You can always get these UUIDs from the URL of the item view of an item.

An example heading row of the CSV export file (project entity):

```
id,collection,dc.title,project.investigator,relation.isPersonOfProject,etc,etc,etc.
```

Subsequent example row in the CSV export file (project entity):

```
350,2292,Project title,"Smith, John::virtual::8585::600||Doe, Jane::virtual::27946::600","d89c1eb1-2e7c-4912-a1eb-f27b17fd6848::virtual::8585::600||e3595b14-6937-47b9-b718-1972cb683943::virtual::27946::600"
```

Admin CSV import

- As always, only the columns and rows that will be changed should be specified. You do not want to import the columns that contain virtual metadata, because they are not stored in the imported items, but in the linked items. So in the above example you don't want to import the *project.investigator* column, but delete it from the CSV.

- To link one item to another you need to create a corresponding column of the *relation* metadata schema, so in our example above *relation.isPersonOfProject*. All columns of the form *relation.*.latestForDiscovery* are created and updated automatically, so you don't want to import them. If you want to create a new relation, of course you don't know the ID of the relation, you can replace it with a +, then DSpace will assign it on its own. Of course, people can also be removed from the column or completely new relations can be created for new items, even if there are no old ones to be taken over.

An example heading row for the CSV import file (project entity):

```
id,collection,dc.title,relation.isPersonOfProject,etc,etc,etc.
```

Subsequent example row for the CSV import file (project entity):

```
350,2292,Project title,"d89c1eb1-2e7c-4912-a1eb-f27b17fd6848::virtual::8585::600||e3595b14-6937-47b9-b718-1972cb683943::virtual::+::600"
```