Batch Metadata Editing Feature

Batch metadata editing

In the recent DSpace+1.6 survey, having a batch metadata editing facility in DSpace was voted one of the top three features that we should be concentrating on. We would like your input into how such a facility should work.

Please either edit this page and add a new section detailing your thoughts on how a batch editing facility should work, or email them to s.lewis@auckland.ac.nz Stuart Lewis who can add them here on your behalf. Please respond by the 20th May 2009.

Responses

University of Auckland

- Stuart Lewis, Vanessa Newton-Wade, Leonie Hayes (13/05/2009)
  We want to be able to perform the following functions:
- Add a new metadata element to every record in a set
- Remove a metadata element to every record in a set
- Change the metadata value of an element in every record in a set
- Be able to make metadata changes to many records at one
- Be able to use external tools that could offer facilities such as spell checking

We would like to see a facility to save an item / collection / community / browse results screen / search results screen to a CSV file. This can then be opened and manipulated externally in a spreadsheet application such as Microsoft Excel. The file can then be uploaded back into DSpace, and the metadata added back into the relevant items. The saving and uploading should work in both the jspui and xmlui, and via the command line.

CSV would have to comply with http://tools.ietf.org/html/rfc4180

A typical line in the CSV file might look like:

```
id      ,dc.contributor.author                               ,dc.title
2292/367,"Lewis, Stuart||Hayes, Leonie||Newton-Wade, Vanessa",How to say ""Hello""
2292/368,"Jones, John"                                       ,A simple title
2292/369,"Jones, John"                                       ,A complex title
```

When the file is uploaded, the changes will be highlighted for confirmation, and then if you confirm this is OK, they will be applied. E.g.:

```
id      ,dc.contributor.author                               ,dc.title
2292/367,"Lewis, Stuart||Hayes, Leonie||Newton-Wade, Vanessa",How to say ""Goodbye""
2292/368,"Jones, John||Smith, Simon"                         ,A simple title
2292/369,"Jones, John"                                       ,A complex title
```

When uploading this, it would say:

```
Item 2292/367: Changed: dc.contributor.author: Was: 'How to say 'Hello"' Now: 'How to say "Goodbye"'
Item 2292/368: Changed: dc.title: Was: 'Jones, John' Now: 'Jones, John' and 'Smith, Simon'
Item 2292/369: No changes

Commit changes to the database?
```

How to get the best out of the Batch Metadata Editing tool - Updated 28/7/09 Leonie Hayes

After exporting the results from your collection in a csv file, I found it much easier to follow these steps:

- 1. Open the file using Open Office Calc - Spreasheet application. Choose UTF8 format and select the whole column type to be text, then ok
- 2. The spreadsheet opens up and you save it down to an .xls file
- 3. Strip out the fields you don't need like identifier, provenance, as much as possible is best, so you are just left with the id, collection and the fields you want to edit/change.
- 4. A very good tip: this is excellent for importing new items, a whole lot easier than the DSpace import option if you do not have any files to attach. I created a single record with all the details then export this and use the + symbol to add new items. When I have bitstreams I just upload them with skeleton data and use batch editing to enhance the records.
- 5. When you have finished editing the xls file, save it back as a .csv file then import it back into the collection.
- 6. Detailed screen shots of the actual process. ^Batchmetadata.pdf

Vanderbilt University

- Ronee L. Francis (13/05/2009)
  I worked with an SQL based application called Collection Workflow Integrated System and so my idea of being able to edit metadata has come from this example. Below is the database I set up a few years back. Next to each subject is a number representing the number of records with that particular subject. When you are logged in as an administrator an "edit" option is also present next to each number. If you choose edit and change the subject it changes the subject in each record where it is present.

http://xserve2.reuther.wayne.edu/SPT--BrowseResources.php

I found it very useful.

MIT

We have identified several use cases for which administrative, batch-oriented tools that operate with smaller granularity than DSpace item could be useful.
Examples: individual metadata field addition/deletion/replacement, likewise bitstream.
Will likely be delivered as extensions to ItemImporter or a new app.

We concur with the batch metadata editing features as listed above, especially the first 4 bullet points.

- Add a new metadata element to every record in a set
- Remove a metadata element to every record in a set
- Change the metadata value of an element in every record in a set
- Be able to make metadata changes to many records at one

I think this captures most of what we want, but just wanted to add the following:

- More specifically, ideally the ability to change (add, edit, delete) content of existing metadata fields would include some kind of regular expression capability.

- Test mode, see what the changes would look like on the 'console' before committing to final run

Also, if I may digress slightly, a somewhat different need, although similar in kind, concerns working with bitstreams:

- add or delete a bitstream to every item in a set

For example, we have a use-case where we would like to add another category or bitstream to groups of items without having to reload the entire record.

Harvard OSC

We were also thinking of export and import in a tabular format like CSV, that looks fine.

Consider, though, that each metadata value actually has several components:

1. The text value
2. Language code (optional)
3. "Place" index (used for ordering multiple values, optional)
4. *in the future, possibly authority control key*

Given that multiple values of the same field occur frequently, I think it would make sense for the tabular format to have one value per row, broken out into its components. That also makes it easier to add a column for authority control someday.

On ingest, I'd recommend coding each row as a specific instruction, saying "delete the value matching this tuple of (Item, field, value, language)", and "add this new metadata value". Thus a change of value becomes two distinct operations (rows?). The columns might look like

> Handle, schema, element, qualifier, language, ADD|DELETE, value, place, *etc.*.

Also consider how to handle failures – does the whole operation either succeed or fail? (Not recommended, since it could end up being too huge a transaction for the RDBMS.) Does each Item succeed or fail on its own, or each row operation? The ingest process ought to produce a report of what succeeded and what failed.

See Authority+Control+of+Metadata+Values for a proposed change to the data model that would affect this project – and vice versa; this facility is another justification for making unattended ingest of metadata work properly with authority control.

Re Bitstream management, couldn't that also be done by MediaFilters? The LNI *does* give you a resource model down to the Bitstream level, although I don't believe the PUT verb was ever implemented to add them, and no DELETE was ever implemented at all. It would be straightforward to do as an LNI extension.