Batch Metadata Editing Prototype

Batch Editing Prototype

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. Community consulatation on the features of such a facility took place on the wiki page: Batch metadata editing

This page describes a working prototype of such a feature. It is intended to show how the feature works, and what use cases it can fulfil.

The feature has been built by Stuart Lewis at the University of Auckland. A copy of the command line version of the code can be downloaded from http://jira.dspace.org/jira/browse/DS-161 or directly from the SVN repository.

Use cases

The metadata bulk edit tool aims to provide a tool that fulfil the following use cases:

- Bulk edit metadata (e.g. perform an external spell check)
- Bulk add metadata values (e.g. add an abstract to a set of items
- Bulk find and replace metadata values (e.g. correct a mispelled surname across several records)
- Mass move items between collections
- Enable the bulk addition of new items (without files) via a csv file
- Re-order the values in a list (e.g. authors)

Code overview

The code required for this feature is completely self-contained from the core DSpace API and sits in its own package within the org.dspace.app parent package. The code consists of five classes:

1. MetadataExport.java - Used to export collections of items into Comma Separated Values (csv) format. (via ItemIterator, or by Collection / Community or Site)
2. MetadataImport.java - Used to import csv files back into DSpace.
3. DSpaceCSV.java - A utility class to represent, marshal, and unmarshal DSpace items into csv lines.
4. DSpaceCSVLine.java - A single DSpace item represented as a csv line.
5. BulkEditChange.java - The changes made by the importer on an item.

Install the code

- Create the directory *dspace-source/dspace-api/src/main/java/org/dspace/app/bulkedit* and then download and copy the classes into that new package.
- Run *mvn package* to recompile and rebuild DSpace
- Run *ant update* (include *-Dconfig=...* if required) from *dspace-source/dspace/target/dspace-version-SNAPSHOT-build.dir/*
- Go to *dspace/bin/* to run the code.

Run the code

Export

To run the code use:

```
dsrun org.dspace.app.bulkedit.MetadataExport
```

The following flags are supported:

- -f *filename* (required) The filename of the resulting CSV
- -i *handle* The Item / Collection or Community to export. If not specified, all items will be exported
- -a include all metadata fields that are not normally changed (e.g. provenance)
- -h Display the help page

Import

To run the code use:

```
dsrun org.dspace.app.bulkedit.MetadataImport
```

The following flags are supported:

- -f -filename *filename* (required) - the filename of the CSV to load
- -s -silent silent mode - does not prompt if you are sure you want to make the changes - USE WITH CAUTION!
- -e -email *email* - the email address of the user (only required when adding new items)
- -w -workflow - when adding new items, use collection workflow
- -n -notify - when adding new items using a workflow, send notification emails
- -t -template - when adding new items use, the collection template (if it exists)
- -h Display the help page

dspace.cfg options

```
### Bulk metadata editor settings ###
# The delimiter used to separate values within a single field (defaults to a double pipe ||)
# bulkedit.valueseparator = ||
```

```
# The delimiter used to serarate fields (defaults to a comma for CSV)
# bulkedit.fieldseparator = ,
```

```
# A hard limit of the number of items allowed to be edited in one go in the UI
# (does not apply to the command line version)
# bulkedit.gui-item-limit = 20
```

```
# Metadata elements to exclude when exporting via the user interfaces, or when using the
# command line version and not using the -a (all) option.
# bulkedit.ignore-on-export = dc.date.accessioned, dc.date.available, \
#                             dc.date.updated, dc.description.provenance
```

The CSV files

The csv 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.

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 id. All other columns are optional.

A typical heading row looks like:

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

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

```
550,2292/30,Item title,"Lewis, Stuart",2008
```

If you want to store multiple values for a given metadata element, they can be spearated with a double-pipe '||'. E.g.:

```
Lewis, Stuart||Jones, John||Smith, Laura
```

. 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.

The separator can be changed if required by setting a parameter in dspace.cfg: *bulkedit.valueseparator* = ||

If you prefer a different field delimiter to a comma, this can be set in dspace.cfg: *bulkedit.fieldseparator = $*. If you wish to use a tab, semicolon or hash (#) sign as the delimiter, set the value to be 'tab', 'semicolon' or 'hash' (without the quotes). Otherwise just use the value.

When importing a csv file, the importer will *overlay* the data 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. If you only want to edit item abstracts, you can remove all of the other columns and just leave the abstract column.

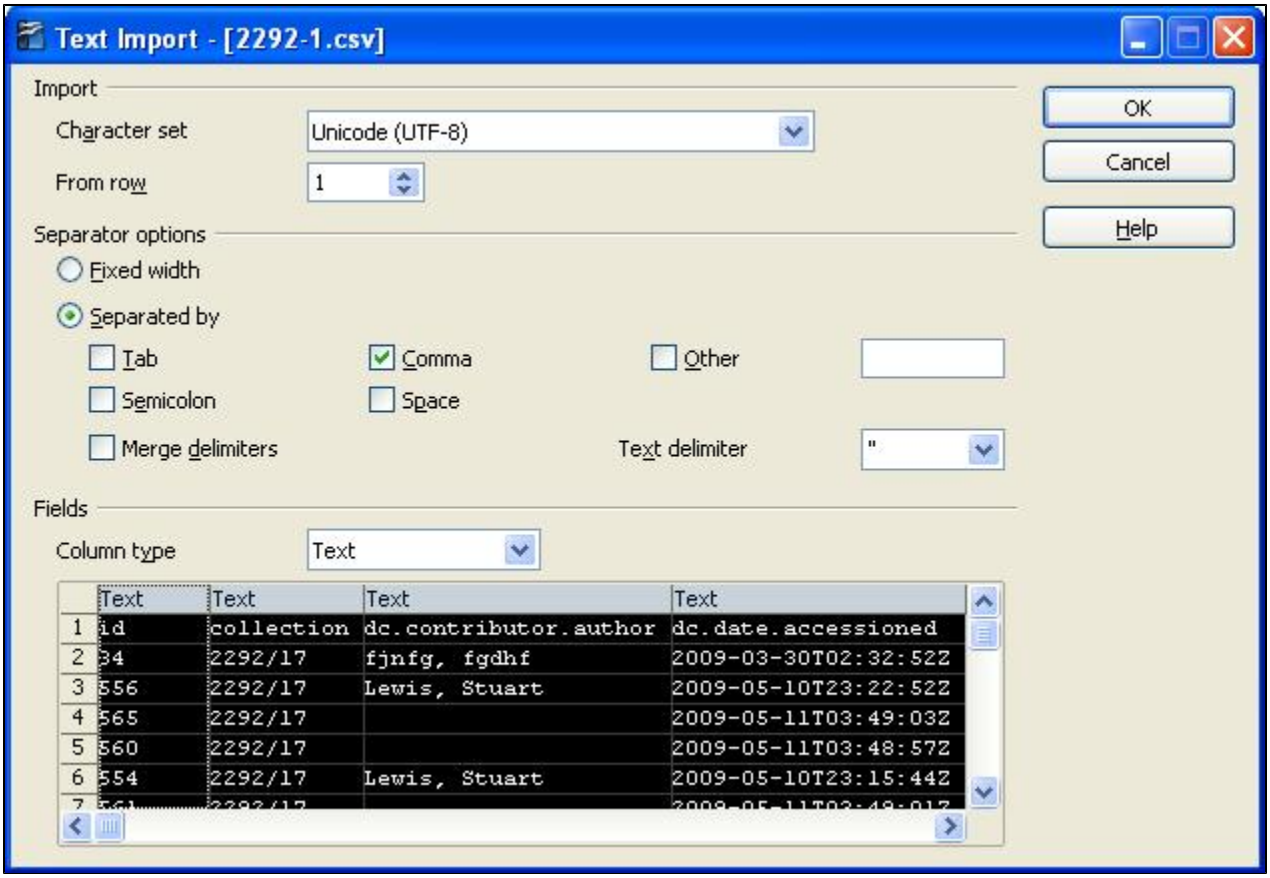
CSV editing hints

Using the spreadsheet function of OpenOffice is probably the easiest solution. Why?

- It opens a dialogue first allowing you to set options.
- Excel will not do this unless you rename your csv file to have a .txt extension.

Ensure you set the following options:

- Character set: Unicode (UTF-8)
- Separator by: Comma
- Fields: Text
    - Apply this to all fields by clicking on the little box in the top left hand corner of the Field matrix (it will all turn black) and then select Text from the Column type dropdown



How to get the best out of the Batch Metadata Editing tool

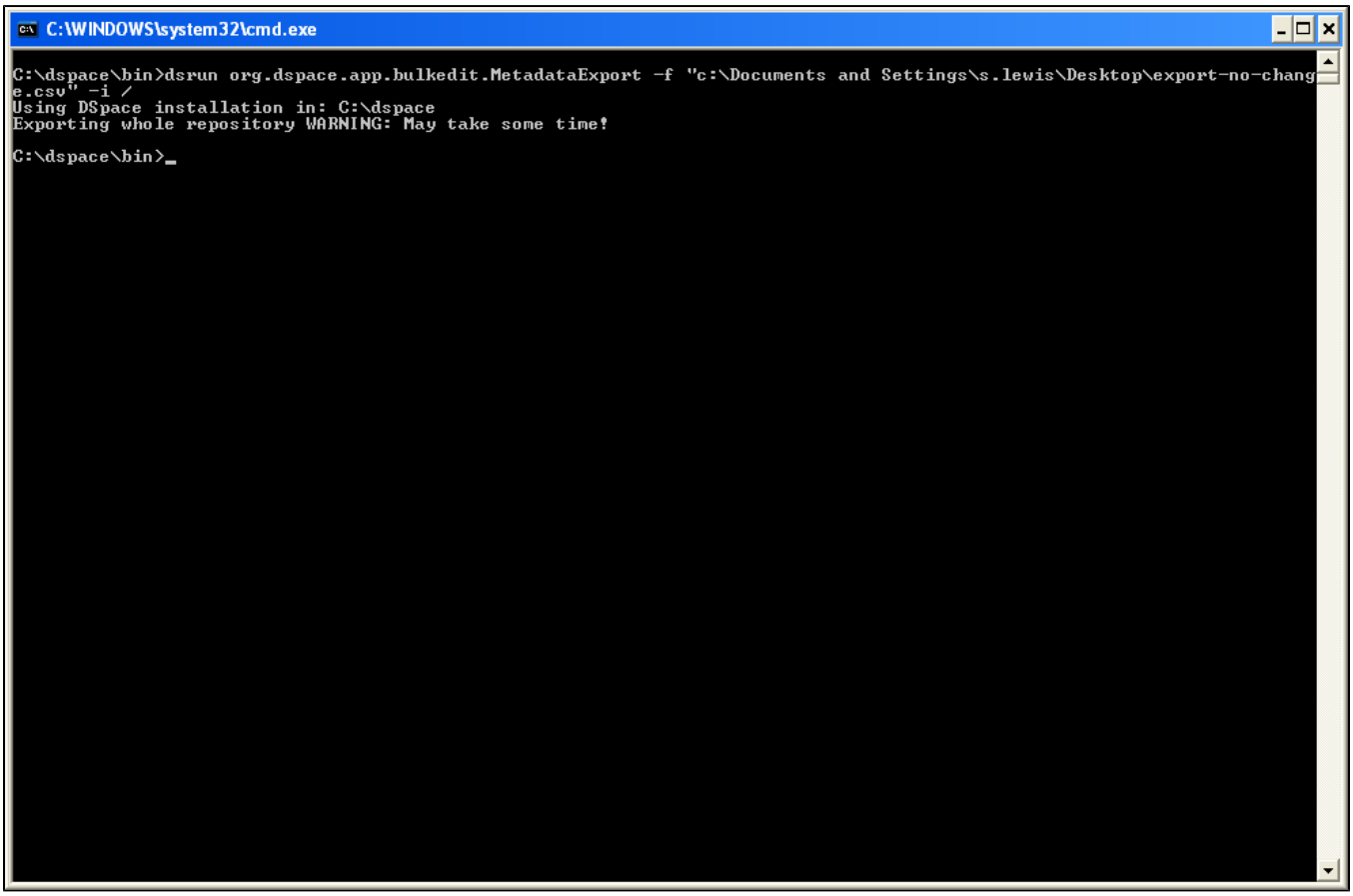
Updated 28/7/09 Leonie Hayes

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

Screen shots

Command line interface

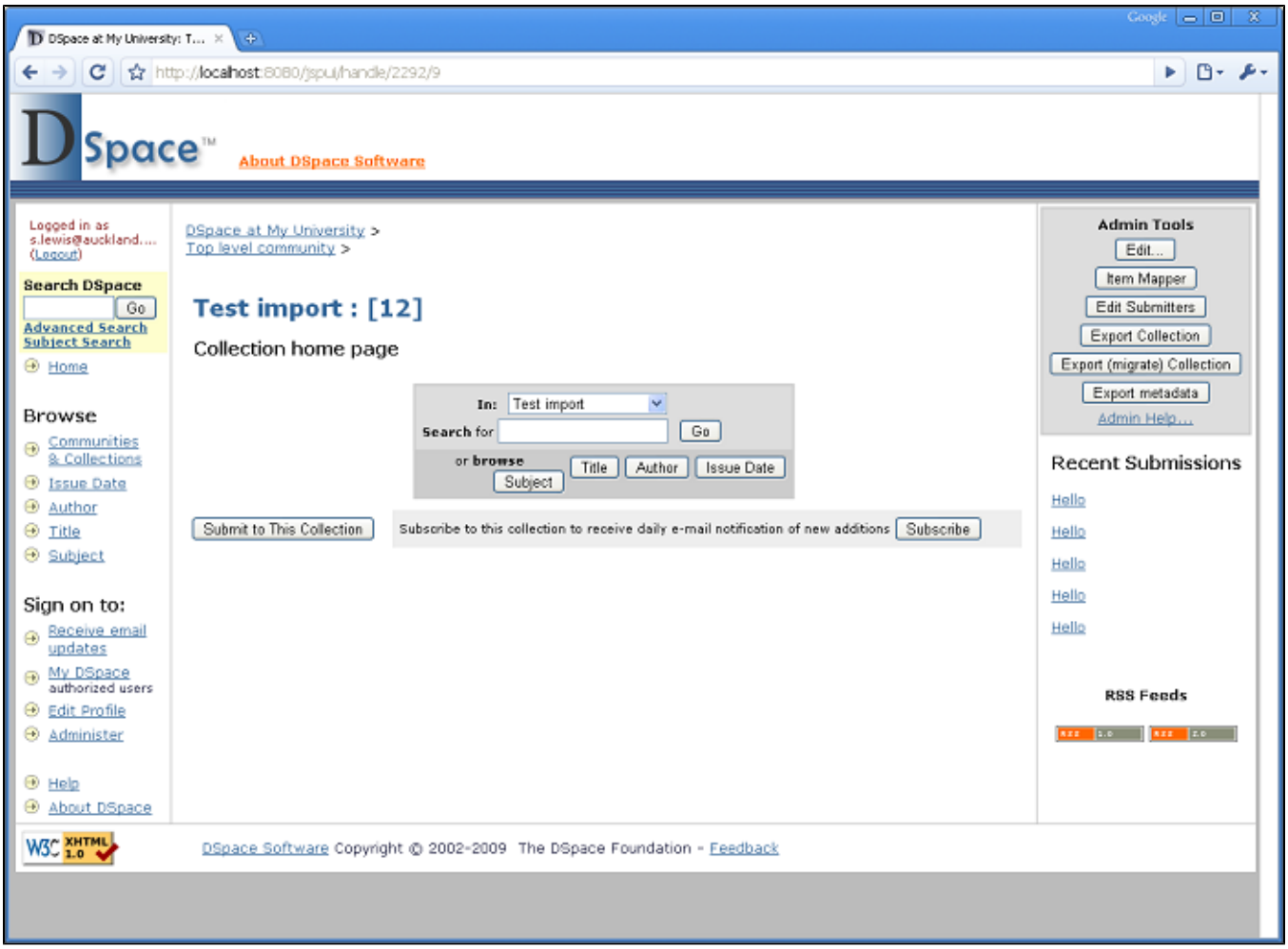


- Export via the command line interface (not very exciting!)

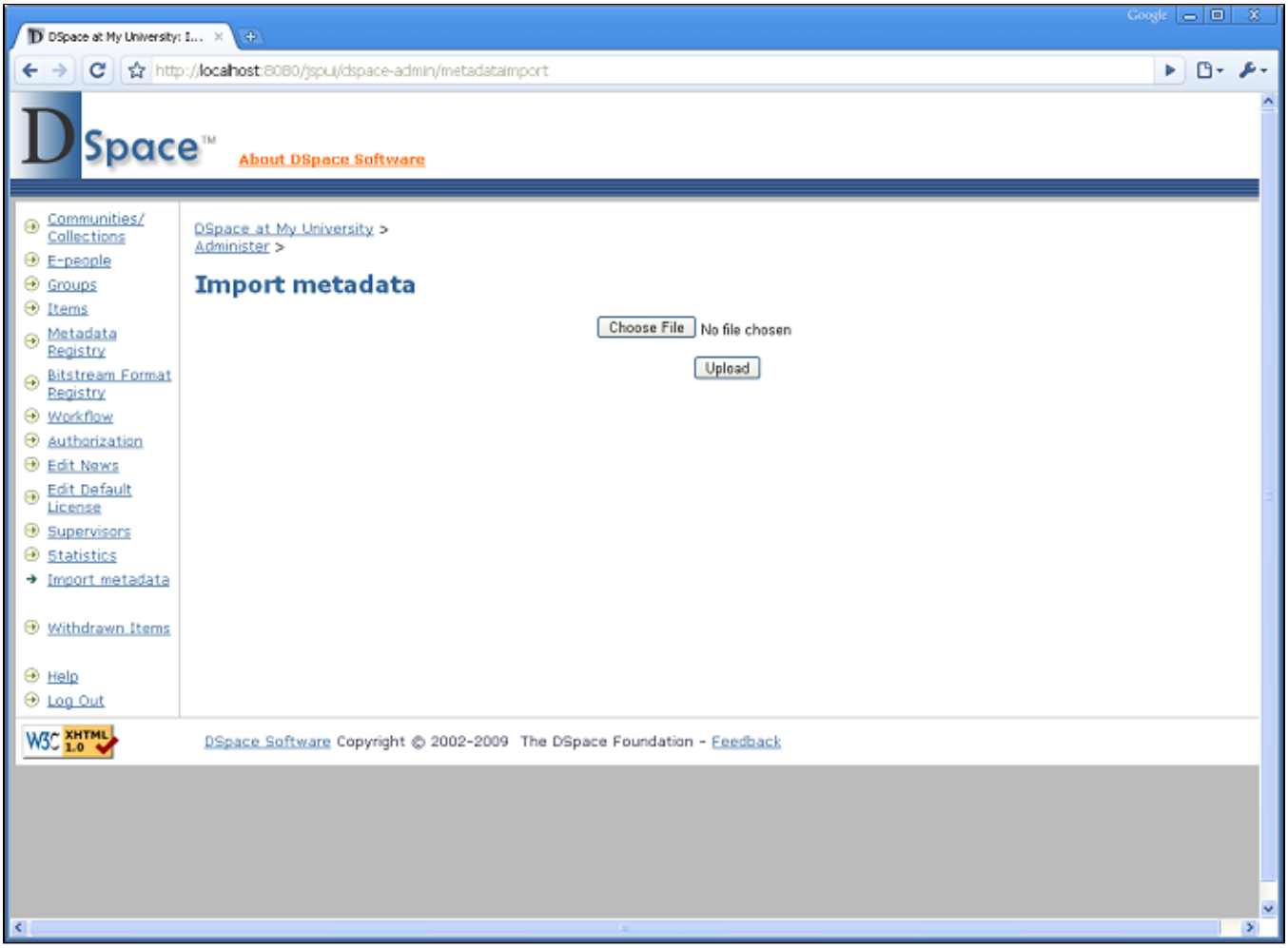
```
C:\WINDOWS\system32\cmd.exe

C:\dspace\bin>dsrun org.dspace.app.bulkedit.MetadataImport -f "c:\Documents and Settings\s.lewis\Desktop\export.csv" -e
s.lewis@auckland.ac.nz
Using DSpace installation in: C:\dspace
------------------------------------------------------------
Changes for item: 14 (2292/16)
 + Add    (dc.title[en_US]): Test - full import
 + Add    (dc.contributor.author): Jones, Tom
 - Remove (dc.title[en_US]): Test - full input-forms
 - Remove (dc.contributor.author): Hayes, Leonie
------------------------------------------------------------
Changes for item: 30 (2292/28)
 + Add    (dc.title[en_US]): DS-117
 - Remove (dc.title[en_US]): DS-116
------------------------------------------------------------
Changes for item: 7 (2292/8)
 + Remove from collection (2292/2): No workflow
------------------------------------------------------------
New item:
 + Add to collection (2292/2): No workflow
 + Add    (dc.title[en_US]): J&W
 + Add    (dc.contributor.author): Wooster, Bertie

4 item(s) will be changed

Do you want to make these changes? [y/n] y
------------------------------------------------------------
Changes for item: 14 (2292/16)
 + Added   (dc.title[en_US]): Test - full import
 + Added   (dc.contributor.author): Jones, Tom
 - Removed (dc.title[en_US]): Test - full input-forms
 - Removed (dc.contributor.author): Hayes, Leonie
------------------------------------------------------------
Changes for item: 30 (2292/28)
 + Added   (dc.title[en_US]): DS-117
 - Removed (dc.title[en_US]): DS-116
------------------------------------------------------------
Changes for item: 7 (2292/8)
 + Removed from collection  (2292/2): No workflow
------------------------------------------------------------
New item: 1152 (2292/984)
 + Added to collection  (2292/2): No workflow
 + Added   (dc.title[en_US]): J&W
 + Added   (dc.contributor.author): Wooster, Bertie

C:\dspace\bin>_
```

- Import via the command line interface (changes require confirmation before they are made)

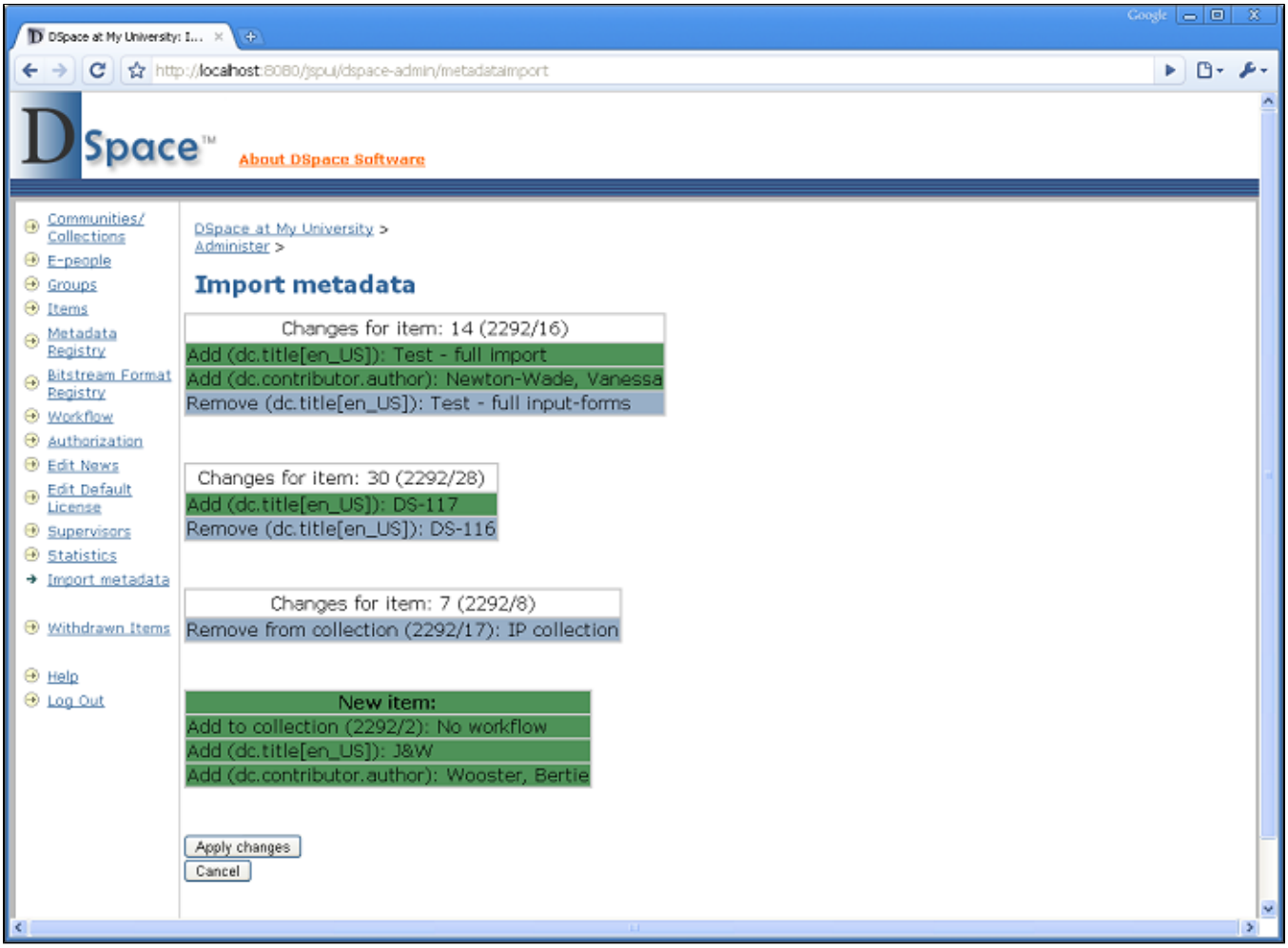
JSPUI screenshots



- Extra button to export metadata in items / collections / communities / search results



- New admin option to upload a CSV file that has been exported and edited



- See what changes it has detected. Choose to make changes, or cancel. (Needs a lot of work to tidy up the user interface)

XMLUI screenshots

Screenshots not prepared yet.

To do

- Add metadata language to field id (e.g. dc.titleen_US)
- Make value separator configurable (e.g. || as a separator)
- Make field separator (currently a comma) configurable
- Allow the id to be '+', meaning the item does not yet exist, so create it
- Implement export and import via the user interfaces (and export for search)
- Allow the export to include the owning collections for each item, allowing you to move items via this method
- Give new items the ability to use a collection template
- Give new items the ability to go through a workflow
- Allow new items to be added to multiple collections
- Write the id or handle of newly created items back to the csv file
- Implement a *silent* option to perform the import without checking first
- Order exported fields by name (e.g. dc.date comes before dc.title)
- Provide option to not export special fields such as dc.decscription.provenanace and dc.date.accessioned