

Project Title: 4624S14DSpaceEmbargo

CS 4625

Virginia Tech

Blacksburg, VA

Members: Jeb Schiefer and Paul Sharma

Clients: Keith Gilbertson and Zhiwu Xie

May 8, 2014

# Table of Contents

[Table of Contents](#)

[Executive Summary](#)

[User Manual](#)

[Figure 1.1](#)

[Figure 1.2](#)

[Figure 1.3](#)

[Figure 1.4](#)

[Figure 1.5](#)

[Figure 1.6](#)

[Figure 1.7](#)

[Figure 1.8](#)

[Figure 1.9](#)

[Figure 1.10](#)

[Developer Manual](#)

[Lessons Learned](#)

[Acknowledgements](#)

[References](#)

## Executive Summary

DSpace (1) is an open source repository application used by many organizations and institutions. It provides a way to access and manage all kinds of digital documents. The 4624S14DSpaceEmbargo project was intended to extend the functionality of the ItemImport command line tool. Specifically the goal was to add the ability to embargo uploaded items until a specified date. This functionality was already implemented for the two web interfaces (XMLUI and JSPUI). DSpace is used by the Virginia Tech library in the form of VTechWorks (8).

The project was overseen initially by Keith Gilbertson and Zhiwu Xie who work for the Virginia Tech library. Near the end of the semester we were introduced to another software developer for the library, Jay Chen. We helped Jay set up the DSpace environment on his local computer and demonstrated to him how to use the ItemImport command line tool.

Embargoes are used to limit access until a specified date. An embargo can be applied as a resource policy on an item, group, or bitstream level. An item level embargo restricts access to all of the files uploaded for a particular item. A group level embargoes submissions from anyone that is a member of the specified group. By default, the Anonymous group is the group that is used. A bitstream level embargo restricts access only on a specific file that is uploaded. The date format expected for setting an embargo must adhere to the ISO 8601 date format (7), specifically the YYYY-MM-DD, YYYY-MM, and YYYY variations.

The deliverables for this project were the source and this documentation. Source code will be available on VTechWorks as well as GitHub. The GitHub repository (6) will be more up to date than the VTechWorks copy because we will continue some work on the project after the due date for this project based on feedback from the DSpace developers. The JIRA ticket for this feature to be implemented in DSpace 5.0 is DS-1996 (2).

## User Manual

Users are given three ways to submit items to a DSpace repository. The first two are in the form of the XMLUI and JSPUI which were not modified in this project. The third option is using the ItemImport command line tool. This tool imports items that are in the DSpace Simple Archive Format. There is one subdirectory for each item to be uploaded. This is a sample directory structure taken from the DSpace wiki page for importing items (5):

```
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
    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 contents file contains the name of each bitstream file, one per line:

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

Each bitstream may optionally have the following:

```
\tbundle:BUNDLENAME
\tpermissions:PERMISSIONS
\tdescription:DESCRIPTION
\tprimary:true
\tembargo:DATE
```

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.

'DATE' is the date to embargo the bitstream until with one of the following formats: YYYY-MM-DD, YYYY-MM, or YYYY.

Primary is used to specify the primary bitstream.

As sample contents file with an embargo on the bitstream would look like this:

```
test.pdf      embargo:2014-06-01
```

test.pdf will be embargoed until June 01, 2014. Note the the whitespace between test.pdf and embargo is a single tab character.

An optional item\_embargo\_date file may be used to embargo an entire item instead of just the individual bitstreams. The file simply contains the date to embargo the item until.

The command line item import workflow is as follows:

Step 1 - Use the import tool to import an item.

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

or

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

--add or -a mean an item is being added to the repository

--eperson or -e is the email of the eperson doing the importing

--collection or -c is the collection id the item is being added to

--source or -s is the directory where the items are located

--mapfile or -m is where the mapfile for the items can be found

--group or -g is optional and specifies the group that the an embargo is being applied against

--test or -t is used to perform a test import of an item

After using the import tool, the submission process is complete.

If an embargo is specified, the item will not be accessible by anyone except an administrator until the date set has passed. When a user tries to access an embargoed item, they will be presented with the following page:

The screenshot shows the DSpace Repository interface. At the top, there is a dark blue header with the DSpace logo and the text "DSpace Repository" in white. A "Login" link is visible in the top right corner. Below the header, a breadcrumb trail reads: "DSpace Home → Embargo Test Community → Embargo Test Collection → Restricted".

The main content area features a prominent heading: "This item is restricted". Below this heading, the text states: "You do not have the credentials to access the restricted item hdl:123456789/33." and "The selected item is access restricted and requires credentials to view. Please login to access the item." A link "Proceed to login screen" is provided.

On the right side, there are three utility boxes:

- Search DSpace:** Contains a search input field, a "Go" button, radio buttons for "Search DSpace" (selected) and "This Collection", and a link for "Advanced Search".
- Browse:** Lists navigation options for "All of DSpace" (Communities & Collections, By Issue Date, Authors, Titles, Subjects) and "This Collection" (By Issue Date, Authors, Titles, Subjects).
- My Account:** Contains links for "Login" and "Register".

Figure 1.1

The XMLUI workflow is as follows:

Step 1 - Click the Submissions button.



Figure 1.2

Step 2 - Start a new submission.

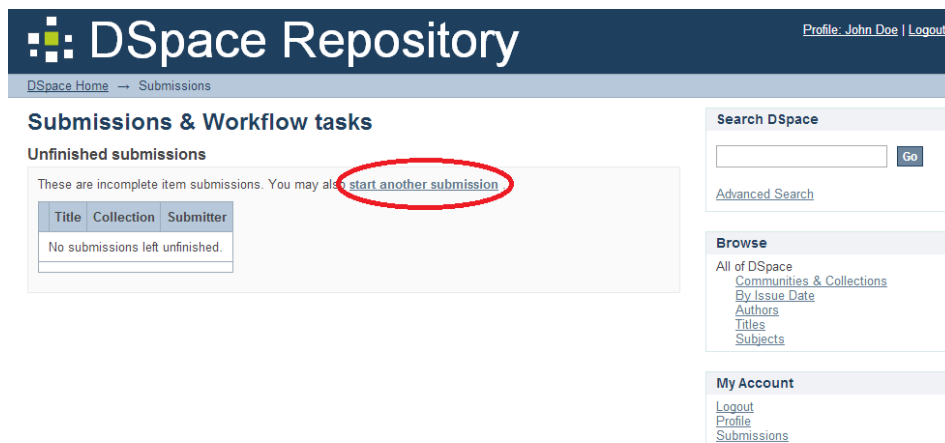


Figure 1.3

Step 3 - Select a collection to add the item to.

The screenshot shows the 'Item submission' page in the DSpace Repository. The main heading is 'Item submission' with a sub-heading 'Select a collection'. Below this, there is a 'Collection:' section with a dropdown menu labeled 'Select a collection...' which is circled in red. A 'Next' button is located below the dropdown. On the right side, there are three panels: 'Search DSpace' with a search box and 'Go' button; 'Browse' with links for 'All of DSpace', 'Communities & Collections', 'By Issue Date', 'Authors', 'Titles', and 'Subjects'; and 'My Account' with links for 'Logout', 'Profile', and 'Submissions'.

Figure 1.4

Step 4 - Describe the item. The title and year of issue are the only required fields.

The screenshot shows the 'Describe Item' section of the 'Item submission' page. It features a progress bar with steps: Describe (highlighted), Access, Upload, Review, License, and Complete. The 'Describe Item' section includes several form fields: 'Authors' (Last name and First name(s) with an 'Add' button), 'Title' (main title), 'Other Titles' (alternative titles with an 'Add' button), 'Date of Issue' (Year, Month, and Day), 'Publisher' (name of the publisher), 'Citation' (standard citation), 'Series/Report No.' (Series Name and Report or paper No. with an 'Add' button), and 'Identifiers' (ISSN and other codes with an 'Add' button'). On the right side, there are three panels: 'Search DSpace' with radio buttons for 'Search DSpace' and 'This Collection'; 'Browse' with links for 'All of DSpace', 'Communities & Collections', 'By Issue Date', 'Authors', 'Titles', 'Subjects', and 'This Collection'; and 'My Account' with links for 'Logout', 'Profile', and 'Submissions'. A 'Context' panel at the bottom right has links for 'Edit Collection' and 'Item Mapper'.



**Type:**

Select the type(s) of content of the item. To select more than one value in the list, you may have to hold down the "CTRL" or "Shift" key.

- Animation
- Article
- Book
- Book chapter
- Dataset
- Learning Object

**Language:**

Select the language of the main content of the item. If the language does not appear in the list below, please select "Other". If the content does not really have a language (for example, if it is a dataset or an image) please select "N/A".

[Save & Exit](#) [Next >](#)

Figure 1.5

**DSpace Repository** Profile: John Doe | Logout

DSpace Home → Embargo Test Community → Embargo Test Collection → Item submission

### Item submission

Describe → **Describe** → Access → Upload → Review → License → Complete

#### Describe Item

**Subject Keywords:**  
Enter appropriate subject keywords or phrases below.  
 [Add](#)

[Subject Categories](#)

**Abstract:**  
Enter the abstract of the item below.

**Sponsors:**  
Enter the names of any sponsors and/or funding codes in the box below.

**Description:**  
Enter any other description or comments in this box.

[< Previous](#) [Save & Exit](#) [Next >](#)

#### Search DSpace

 [Go](#)  
 Search DSpace  
 This Collection  
[Advanced Search](#)

#### Browse

All of DSpace  
[Communities & Collections](#)  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

This Collection  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

#### My Account

[Logout](#)  
[Profile](#)  
[Submissions](#)

#### Context

[Edit Collection](#)  
[Item Mapper](#)

Figure 1.6

## Step 5 - Set an embargo on the item if desired.

**Item submission**

Describe → Describe → **Access** → Upload → Review → License → Complete

### Access Settings

**Private item:**  
If selected, the item won't be searchable  
 Private

### Group policies

Policies listed in this section override any default policies for the collection you're submitting to. If you wish to set an embargo but the target collection allows access for any user, you must set a policy that allows access for the Anonymous group only from a specific date onwards.

No group policies have been set up for this item.

### Embargo

**Group:**  
Anonymous

**Access for selected group:**  
 Allow access once item is accepted into archive  Embargo until specific date  
The first day from which access is allowed. Accepted format: yyyy, yyyy-mm, yyyy-mm-dd

**Policy name:**  
A short, descriptive name for the policy (up to 30 characters). May be shown to end users.  
Example: "Staff-only". Optional but recommended.

**Embargo reason:**  
The reason for the embargo, typically for internal use only. Optional.

[Confirm Policy & add another](#)

[< Previous](#) [Save & Exit](#) [Next >](#)

**Search DSpace**  
 [Go](#)  
 Search DSpace  
 This Collection  
[Advanced Search](#)

**Browse**  
All of DSpace  
[Communities & Collections](#)  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)  
This Collection  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

**My Account**  
[Logout](#)  
[Profile](#)  
[Submissions](#)

**Context**  
[Edit Collection](#)  
[Item Mapper](#)

Figure 1.7

## Step 6 - Upload the items.

**Item submission**

Describe → Describe → Access → **Upload** → Review → License → Complete

### Upload File(s)

**File:**  
Please enter the full path of the file on your computer corresponding to your item. If you click "Browse...", a new window will allow you to select the file from your computer.  
 No file chosen

**File Description:**  
Optionally, provide a brief description of the file, for example "Main article", or "Experiment data readings".

[Upload file & add another](#)

[< Previous](#) [Save & Exit](#) [Next >](#)

**Search DSpace**  
 [Go](#)  
 Search DSpace  
 This Collection  
[Advanced Search](#)

**Browse**  
All of DSpace  
[Communities & Collections](#)  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)  
This Collection  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

**My Account**  
[Logout](#)  
[Profile](#)  
[Submissions](#)

**Context**  
[Edit Collection](#)  
[Item Mapper](#)

Figure 1.8

## Step 7 - Review the submission.

**Item submission**

Describe → Describe → Access → Upload → **Review** → License → Complete

### Review Submission

**Describe Item**

Title: test  
Date of Issue: 2014  
[Correct one of these](#)

**Describe Item**

[Correct one of these](#)

**Access Settings**

**Private item:**  
The item will be searchable

**Group policies:**  
No group policies have been set up for this item.  
[Correct one of these](#)

**Upload File(s)**

test.txt - Text (Known)  
[Correct one of these](#)

< Previous   Save & Exit   Next >

**Search DSpace**

[Go](#)

Search DSpace  
 This Collection  
[Advanced Search](#)

**Browse**

All of DSpace  
[Communities & Collections](#)  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

This Collection  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

**My Account**

[Logout](#)  
[Profile](#)  
[Submissions](#)

**Context**

[Edit Collection](#)  
[Item Mapper](#)

Figure 1.9

## Step 8 - Accept the license.

**Item submission**

Describe → Describe → Access → Upload → Review → **License** → Complete

### Distribution License

There is one last step: In order for DSpace to reproduce, translate and distribute your submission worldwide, you must agree to the following terms.

Grant the standard distribution license by selecting 'Grant the License'; and then click 'Complete Submission'.

Special licence

If you have questions regarding this license please contact the system administrators.

**Distribution license:**

Grant the License

< Previous   Save & Exit   Complete submission

**Search DSpace**

[Go](#)

Search DSpace  
 This Collection  
[Advanced Search](#)

**Browse**

All of DSpace  
[Communities & Collections](#)  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

This Collection  
[By Issue Date](#)  
[Authors](#)  
[Titles](#)  
[Subjects](#)

**My Account**

[Logout](#)  
[Profile](#)  
[Submissions](#)

**Context**

[Edit Collection](#)  
[Item Mapper](#)

Figure 1.10

## Step 9 - Complete the submission.

## Developer Manual

For developers and library personnel who want to extend and improve upon our work should go through the following steps and understand the changes we've made to the project –

- 1). The only place we've made modifications is `ItemImport.java` as that's the main class for the batch item import application.
- 2). We added a new switch (-g) to the instance of `Options` class to add support for group name against which the embargo will be applied.
- 3). If there is -g switch specified in the command line, we retrieve the group name and assign that to a global variable.
- 4). We added a new method “`setEmbargo(Context c, Date embargoDate, String groupName, DSpaceObject dspaceobj)`” that takes the context, embargo date, group name, dspace resource and applies the embargo on the resource. The resource could be an item, bundle or bitstream. If the group name is not null then it applies the embargo against that specific group else it applies embargo against ANONYMOUS group which is everyone, so that once embargo expires only that group has access to that resource. The way it's implemented is that first we check whether we are authorized to add read related resource policies pertaining to that resource. We call the `AuthorizeManager.authorizeAction` method for that. If we are not allowed to do the latter then the method throws `AuthorizeException`, otherwise we go ahead and call the `AuthorizeManager.createOrModifyPolicy` that either creates a new resource policy or updates an existing one with the new information provided. Then we call the `update` method on that `ResourcePolicy` object which adds those queries to the context/transaction to be committed later.
- 5). In `addItem` method after an instance of `Item` class is created we call the “`setEmbargo`” method on that item, then call the same method on all the bundles under that item and all the bitstreams under those bundles. In every item directory there is an `item_embargo_date` file that we look for. This file supposedly contains the embargo date to be applied on that item.
- 6). In the `processContentsFile` method which parses the contents file in every item folder, we add a block of code that looks for `\tembargo: substring` (\t is tab character). If this substring exists then there should be an embargo date specified that will be applied against that bitstream in the same row. This block of code is placed next to pieces of code that search for and retrieve other kinds of data about the bitstream in that row for e.g:- `\tbundle: (bundle to which this bitstream is added) , \tdescription: (description of this bitstream),etc`. In this method we retrieve all this information, add it to a string and return it. In the `addItem` method all these strings are added to a `List`.
- 7). After the control comes back from `processContentsFile` and we've added embargo on the given item, its bundles and its bitstreams, we proceed with parsing all the metadata about every bitstream we collected in the `processContentsFile` method. This work is done in the `processOptions` method. In this method we again do a similar thing and retrieve embargo date from every string in list of options (that was retrieved from contents file) and convert the string fetched into an instance of `Date` class using a method that parses and understands ISO-8601 compliant date formats. Then we call the `setEmbargo`

method with that date. If an earlier resource policy record/embargo has been created with the item embargo date entered then this date will overwrite the previous date. So, the given bitstream will have a different embargo date from the one applied to the item/bundle to which it belongs.

The way embargo works in Item Import is identical to that in JSPUI/XMLUI. In the latter if you go ahead and submit an item, then in the workflow you are first given an option to enter an embargo date for an item. Once you enter the date, the name of the embargo, the reason for the embargo and press next, in the next step you will see resource policies/embargo been applied to that item and all the bundles that belong to it. In the next step you are then asked to upload bitstreams and embargo dates for those bitstreams. If you leave the embargo date blank then automatically a resource policy with the item embargo date will be applied to that bitstream. But if you specify a date, then an embargo policy with that date will be applied for that bitstream. And if you do not enter an item embargo date but specify embargo dates for bitstreams, then only embargo dates for bitstreams will be created. We mimicked that process in our item import. First we parse the `item_embargo_date` file, retrieve the item embargo date and apply embargoes to the item, the bundles and the bitstreams. Once we are in the processOptions and applying embargoes to bitstreams then the above embargoes applied to bitstreams with the item embargo date will be overwritten. And if there is no `item_embargo_date` file in the item folder i.e. no embargo date specified for an item, then no embargo will be applied to the item, its bundles and bitstreams. And if any embargo dates are mentioned in the contents file for bitstreams, then separate resource policies/embargo dates will be applied to for those bitstreams.

Given below is a sample resource policy record –

```
policy_id: 4847
resource_type_id: 2
resource_id: 89
action_id: 0
eperson_id:
epersongroup_id: 0
start_date: 2013-01-01
end_date:
rpname: Embargo Policy
rpdescription: Embargoed through 2012
rptype: TYPE_CUSTOM
```

The `policy_id` is the primary key. The `resource_type_id` denotes the type of the resource for e.g.:- item, bundle, bitstream, collection, etc. The `resource id` is the resource identifier. The `action_id` denotes the type of action for e.g.:- read, write, etc. For embargo the action type is always READ. The `eperson_id` is left empty for embargo. The `eperson_group_id` is the user group that will have access to the resource after embargo expires. The group id 0 is everyone which is the most common one unless a specific group name is supplied via the `-g` switch in item import. The `start_date` is where the embargo date is stored. The `end_date` field is left empty. Then you have the resource policy, name, type and description.

The name and description is set by our code. The resource policy type is always TYPE\_CUSTOM for user generated embargo.

## Lessons Learned

Given below are the lessons we learned –

- 1). We understood the architecture of dspace, how items are submitted, what is their metadata structure, how is it searched for using solr.
- 2). We understood what embargo is and how it works.
- 3). We perused ItemImport.java and understood all the classes in the dspace core API for e.g.:- Context.java, Item.java, Bitstream.java, Bundle.java, ResourcePolicy.java, etc.
- 4). We went through the source code in XMLUI/JSPUI to understand how embargo is being applied over there. We played around with the above web apps, uploaded sample items and placed embargo dates, then checked the resource policy table to see the records in the resource\_policy table.
- 5). We went through some hurdles setting up DSpace in our systems. We had to go through the tutorials in dspace wiki in order to get comfortable.
- 6). Initially we thought that by just setting embargoes on bitstreams, we would be done but at that time we didn't know exactly how embargo was being done in XMLUI/JSPUI. After deploying web apps on our local machine and playing with the latter, we came to understand exactly how embargo is supposed to work and we modified our approach.

## Acknowledgements

We would like to thank our clients Keith Gilbertson (keith.gilbertson@vt.edu) and Zhiwu Xie (zhiwuxie@vt.edu) for allowing us to work on the DSpace embargo project. Their assistance in setting up and using DSpace was invaluable. We would also like to thank Jay Chen (jaychen@vt.edu) for providing valuable feedback on our testing procedures. Additionally we would like to thank Tim Donohue (tdonohue@duraspace.org) and all the other DSpace developers for their feedback and guidance for merging the ItemImport embargo feature back into the main source code branch. Finally, we would like to thank Dr. Fox (fox@vt.edu) for his guidance throughout CS 4624.

## References

- 1) DuraSpace, “DSpace”, 2014, <http://dspace.org/>
- 2) DuraSpace JIRA, “[DS-1996] Embargo Support in ItemImport,” 2014, <https://jira.duraspace.org/browse/DS-1996>
- 3) DuraSpace Wiki, “Business Logic Layer,” 2014, <https://wiki.duraspace.org/display/DSDOC4x/Business+Logic+Layer>
- 4) DuraSpace Wiki, “Embargo,” 2014, <https://wiki.duraspace.org/display/DSDOC4x/Embargo>
- 5) DuraSpace Wiki, “Importing and Exporting Items via Simple Archive Format”, 2014, <https://wiki.duraspace.org/display/DSDOC4x/Importing+and+Exporting+Items+via+Simple+Archive+Format>
- 6) GitHub, “jebischief/DSpace,” 2014, [https://github.com/jebischief/DSpace/tree/cli-embargo-4\\_x](https://github.com/jebischief/DSpace/tree/cli-embargo-4_x)
- 7) ISO, “Date and time format - ISO 8601,” 2014, <http://www.iso.org/iso/home/standards/iso8601.htm>
- 8) Virginia Tech, “VTechWorks”, 2014, <http://vtechworks.lib.vt.edu/>