

Embargo on Bitstream v2 (JSP)

Contents

- 1 [Overview](#)
- 2 [How It Works](#)
- 3 [To Be Noted](#)
- 4 [Place New Files](#)
- 5 [Modify Files](#)
 - 5.1 [Modifying HandleServlet.java](#)
 - 5.2 [Modifying WorkflowManager.java](#)
 - 5.3 [Modifying input-forms.xml](#)
- 6 [Create Database Tables](#)
- 7 [Update DSpace and Test](#)
- 8 [Set Up Daily Embargo Job](#)
- 9 [Final Words](#)
- 10 [Files](#)

Overview

This is a write-up of a method of modifying DSpace to allow the embargoing of documents submitted to a repository. The [other method](#) on the DSpace wiki is written for an older version of DSpace and the instructions no longer (at least clearly) apply to 1.4.2.

This method was published to the DSpace users mailing list by Jose Blanco at the University of Michigan, but the current archive of the thread is on SourceForge does not contain the correct files needed to perform the modification, and is not easy to understand for those unfamiliar to complex DSpace customizations. With help from Graeme Fox at Massey University in New Zealand, I was able to apply the embargo modification to our repository. These instructions should hopefully guide you through the process.

How It Works

This modification works by essentially adding another piece of metadata to the documents that need to be embargoed, which is stored in a new table in the postgres database. The new code checks against the database to ensure that an item is allowed to be accessed, and if it is under embargo a custom page is displayed that states the item is not available. A script runs on a frequency you set (likely daily) that checks the table in the database and removes the embargo if the date initially specified has been reached, which allows it to be accessed. Note that I did not develop this code and am simply stating my understanding of the modification based on the code.

To Be Noted

Items placed under embargo DO NOT appear in the repository in any way. It is as if they were never entered in the first place. If you wish to see what items are currently under embargo, you will need to access your database directly and view the new table. Items can be released by manually modifying the embargo date. If you wish for the metadata or portions of the document to be accessible (such as an abstract), this solution will likely not be satisfactory.

Place New Files

1. Place Restrict.java in <dspace-source>/src/org/dspace/app/webui/util
2. Place ReleaseRestricted.java in <dspace-source>/src/org/dspace/administer/
3. Place ReleaseRestrictedItems.sh in <dspace>/bin
4. Place embargo.jsp in <dspace-source>/jsp/local/

Modify Files

It's likely a good idea to back up any files you are going to modify before you begin changing anything.

First, modify <dspace>/bin/ReleaseRestrictedItems.sh, replacing `userid@university.edu` with the proper administer email address. You may wish to modify <dspace-source>/jsp/local/embargo.jsp to include less generic information, as well.

Modifying HandleServlet.java

Open <dspace-source>/src/org/dspace/app/webui/servlet/HandleServlet.java.

Following the initial imports, add this:

```
import org.dspace.app.webui.util.Restrict;
```

Line 260 (in my file, DSpace 1.4.2) follows four lines stating:

```
private void displayItem(Context context, HttpServletRequest request,
    HttpServletResponse response, Item item, String handle)
    throws ServletException, IOException, SQLException,
    AuthorizeException
```

Following the last line, remove this code:

```
{
    // Tombstone?
    if (item.isWithdrawn())
    {
        JSPManager.showJSP(request, response, "/tombstone.jsp");

        return;
    }
}
```

and insert the following code:

```
{
    Item[] embargoItems = Restrict.getRestricted ( context );
    int item_id = item.getID();
    for (int i = 0; i < embargoItems.length; i++)
    {
        if ( item_id == embargoItems[i].getID() )
        {
            // This item is in embargo, not withdrawn
            JSPManager.showJSP(request, response, "/embargo.jsp");
            return;
        }
    }
}
```

I'm uncertain as to what the removal of the tombstone code does, but it appears to have no adverse effects. For more details concerning this, please see the [original thread](#).

Modifying WorkflowManager.java

Open <dspace-source>/src/org/dspace/workflow/WorkflowManager.java.

Following the initial imports, add this:

```
import org.dspace.app.webui.util.Restrict;
```

After this section of code (I'm not sure if it needs to be placed here, but I followed Graeme's files as a model):

```
// Log the event
log.info(LogManager.getHeader(c, "install_item", "workflow_id="
    + wfi.getID() + ", item_id=" + item.getID() + "handle=FIXME"));
```

insert the following:

```

// Now that the item is archived. See if you need to withdraw it,
// because of restriction.
// Beginning. This code is new to handle restriction.
DCValue[] restriction = item.getDC("date", "available", Item.ANY);
//log.warn(LogManager.getHeader(c, "restriction",
//                                "restriction" + restriction[0].value));

int restrictionValue = -1;
if (restriction[0].value.equals("WITHHELD_THREE_MONTHS"))
{
    restrictionValue = 0;
}
else if (restriction[0].value.equals("WITHHELD_HALF_YEAR"))
{
    restrictionValue = 1;
}
else if (restriction[0].value.equals("WITHHELD_ONE_YEAR"))
{
    restrictionValue = 2;
}

//log.warn(LogManager.getHeader(c, "restrictionValue",
//                                "restrictionValue" + restrictionValue));

if (restrictionValue != -1)
{
    // apply the restrictions to the item
    Restrict.apply(c, item, restrictionValue);
}
//End of new code.

```

Modifying input-forms.xml

Open <dspace>/config/input-forms.xml.

Add the following field to the appropriate forms:

```

<field>
<dc-element>date</dc-element>
<dc-qualifier>available</dc-qualifier>
<repeatable>false</repeatable>
<label>Date thesis is available</label>
<input-type
value-pairs-name="embargo_thesis">dropdown</input-type>
<hint> </hint>
<required></required>
</field>

```

and the following code in the section for value-pairs:

```

<value-pairs value-pairs-name="embargo_thesis"
dc-term="date_available">
<pair>
<displayed-value>Immediate access to full description and
thesis</displayed-value>
<stored-value>NO_RESTRICTION</stored-value>
</pair>
<pair>
<displayed-value>Access to full description and thesis in 3
months</displayed-value>
<stored-value>WITHHELD_THREE_MONTHS</stored-value>
</pair>
<pair>
<displayed-value>Access to full description and thesis in 6
months</displayed-value>
<stored-value>WITHHELD_HALF_YEAR</stored-value>
</pair>
<pair>
<displayed-value>Access to full description and thesis in 1
year</displayed-value>
<stored-value>WITHHELD_ONE_YEAR</stored-value>
</pair>
</value-pairs>

```

Create Database Tables

Using your postgres tool of choice, issue the following command to create the proper table in your dspace database:

```

CREATE SEQUENCE umrestricted_seq;

CREATE TABLE umrestricted (
id integer DEFAULT nextval('umrestricted_seq'),
item_id integer,
release_date timestamp,
CONSTRAINT umrestricted_pkey PRIMARY KEY (id)
);

```

In Oracle, the DEFAULT clause isn't legal, and doing the equivalent with a trigger actually breaks it, so the following may be what's needed. (Code by gmcgath, not fully tested but seems to work.)

```

CREATE SEQUENCE umrestricted_seq;

CREATE TABLE umrestricted (
id integer,
item_id integer,
release_date timestamp,
CONSTRAINT umrestricted_pkey PRIMARY KEY (id)
);

```

Update DSpace and Test

Do whatever you usually do to update your installation of DSpace. Everything should (hopefully) compile correctly during your update. Test to see if the embargo code is functioning in your input form. You can also pull up the umrestricted table to see any embargoed documents you enter into the collection, and notice how the embargo date is entered. This is the only way I know of to change embargo dates and remove the embargo from a document (change the embargo date to the current one).

Set Up Daily Embargo Job

Once you are certain the code works, the final step is setting up the daily `ReleaseRestrictedItems.sh` job. You can do this in any way your system supports. The script checks the database table and releases items that are no longer under embargo.

Final Words

I hope this guide works for you. I've tried to remember all the small steps and issues I overcame in adding the embargo feature to our repository. Of course, thanks goes to Jose Blanco for sharing this in the first place, and Graeme Fox for helping me parse through everything.

Files

- - [Embargo.jsp.mht](#)
 - [ReleaseRestricted.java](#)
 - [ReleaseRestrictedItems.sh](#)
 - [Restrict.java](#)