

# BatchDO 2.1 README

04/20/2012

This README details the BatchDO 2.1 plugin which automates the workflow for the creation and updating of digital objects, and transfers barcodes placed in the "Instance Type" field into the "Barcode Field". Though tailored to NYU's infrastructure, most of the functionality can be used by other institutions. It is designed as a RDE (Rapid Data Entry) plugin, so it is accessed through the RDE drop-down menu.

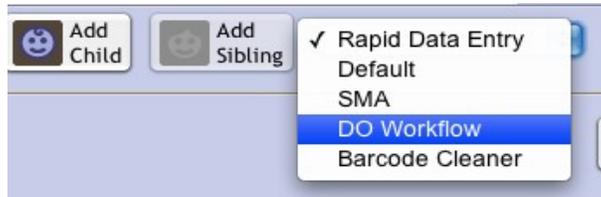


Figure 1

## INSTALLATION

Drop the batchDO2\_nyu.zip file into the plugin directory of the Archivists' Toolkit.

## USING FUNCTIONALITY

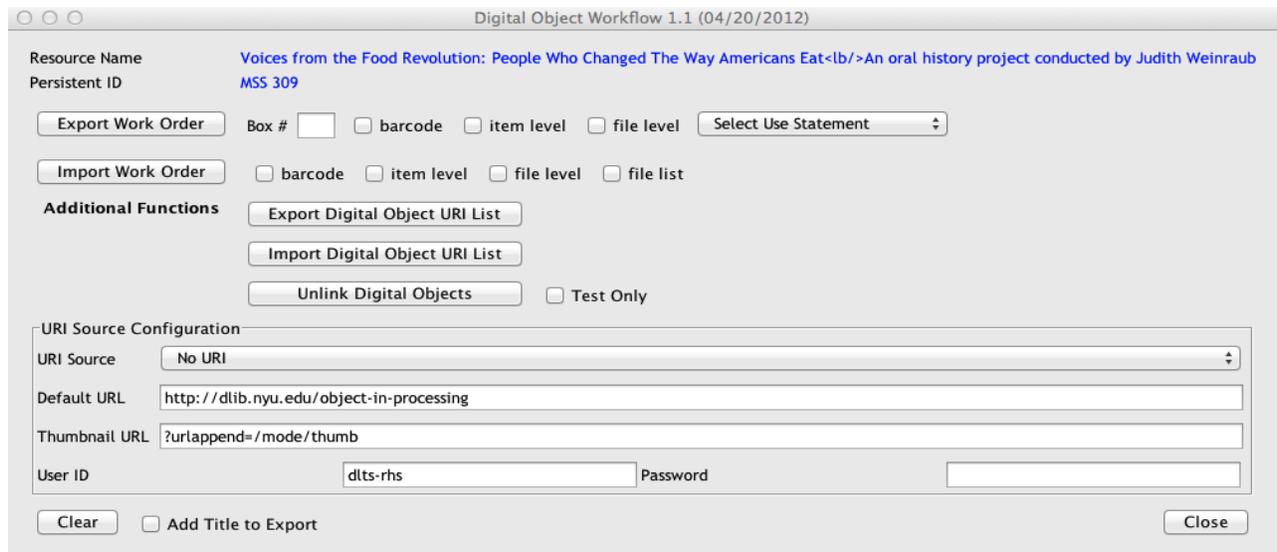


Figure 2

### Exporting Work Order Files:

Work orders are tab delimited files containing a list of box/folder/items instances, or “file level” components (in the case of born-digital collections), containing the unique “Persistent ID” of the

resource component. Such a file is used during the digitization process to map resource components/analog instances, to what's being digitized. Once the digitization process is completed, these files can be used for the automatic creation, and linking of digital objects to the correct resource component. In the case of born-digital collections, in which no digitization needs to be done, then this file can also be used to create and attached digital objects to the correct resource component. By default, the resource component title is not exported, but can be by selecting the “Add Title to Exported File” checkbox.

**Steps for creating a work order file containing box/folder/item level, or barcode information from an analog instance.**

1. Select the parent resource component you want information to be extracted from.
2. Select the “DO Workflow” from the RDE drop down menu. This will open the “Digital Object Workflow” dialog as seen in the above image.
3. Enter the box number in the “Box #” field. Leaving this blank will cause information for all boxes in the child records, of the selected component to be extracted.
4. If you want to export a list containing just the Persistent ID, and instance barcodes then select the “Barcode” check box.
5. If there are item level instances (Container Indicator 3 is populated with information), then select the “item level” check box, otherwise leave it unchecked.
6. Press the “Export Work Order” button. The plugin will go through all the child components, then a dialog (see figure 3) containing the extracted information will be displayed. The tab delimited text in this dialog can either be saved to a file, or copied into a spreadsheet program.

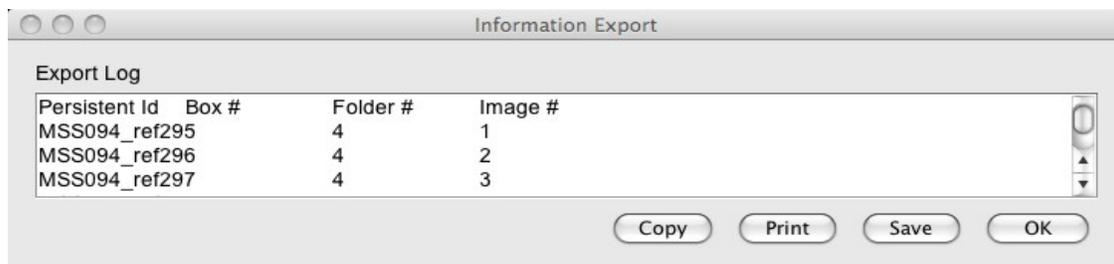


Figure 3

**Steps for creating a work order file for “file level” components, typical of born digital content.**

1. Select the parent component for which information is to be extracted, or just select the top level component to extract information from entire resource record.
2. Select the “DO Workflow” from the RDE drop down menu. This will open the “Digital Object Workflow” dialog as seen in figure 2.
3. Check the “file level” check box. This will extract information for all components whose level is specified as “file”.
4. Select the “Use Statement” from the drop-down menu next to the “file level” check box. If this is not selected then a default “Use Statement” of “Image-Service” is used.
5. Press the “Export Work Order” button. This will search through the file level components, and then a dialog (see figure 4) containing the extracted information will be displayed. As above,

the tab delimited text in this window can either be saved to a file, or copied into a spreadsheet program.

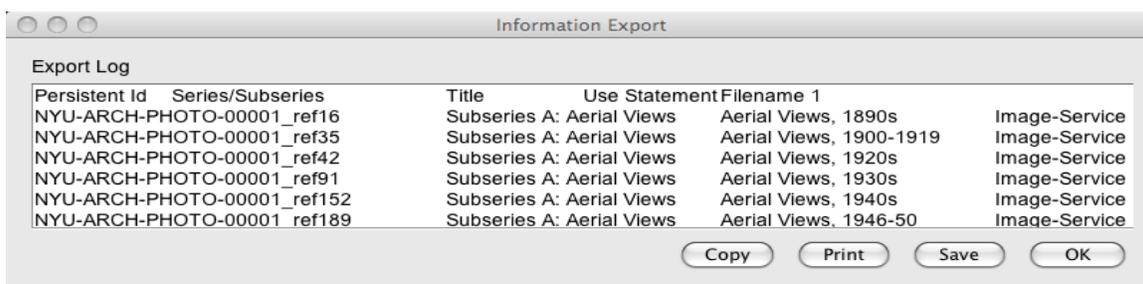


Figure 4

### Using the “Attachment” Keyword with File Level Export

It may be sometimes necessary to attach a digital object to resource components which are not “file level”, and have no analog instances associated with them. In this case, putting the “Attachment” keyword in the “Repository Processing Note” text field will result in an entry for this component when a work order file is exported. By default, “Text-Service” use statement is set on export for the component, but it can be specified as part of the Attachment keyword. For example, to have the use statement of “Image-Service” be exported, simply enter “Attachment:Image-Service” in the “Repository Processing Note” text field. Multiple entries for same resource component in the export file can be specified by putting “Attachment” keywords on separate lines. Each line with the attachment keyword will be exported as separate entry. Please note that once the export process is done, then all text in the “Repository Processing Note” text field is deleted.

### Importing Work Order Files:

After the relevant information has been entered into the export work order file, it can then be imported for the automatic creation, and attachment of digital object instances to the correct resource component. Depending on how the URI (Universal Resource Identifier) source is configured, URIs can be generated using the NYU's RESTfulHS webservice (Handle based URIs), generating by using the filename (this option is only valid when doing “barcode” or “file list” import), or can be read in from the work order file directly. See below for information on configuring the URI source. Once the import process has completed, the assigned handle URIs are appended to the information in the work order file, and saved to a new CSV file, with the text “\_URI” attached to the filename. This modified CSV file (see image below) serves to correlate persistent Ids, filenames, and URIs, so it can be used in the publication process later on.

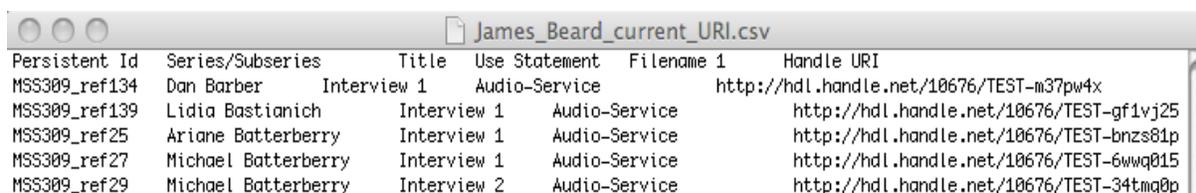


Figure 5

Steps for importing a work order file.

1. Select/Configure the URI source (see below).
2. Based on the type of work order file being imported, select the appropriate checkbox. If the “item level” checkbox is selected, then both a “Service” and “Thumbnail” digital object will be created for each entry in the import file. Please note that the URI of the “thumbnail” digital object is derived from the service URI + the entry in the “Thumbnail URL” text box.
3. Press the “Import Work Order” button to begin the import process.
4. Verify that the digital objects have been created, and attach to the correct resource components.
5. Save the Resource record.

## Configuring the URI Source

Before importing a work order file, the source of the URIs must be selected from the “URI Source” drop-down menu (see figure 6). For production work, select the “Production Handle Service” which generates Handle based URIs using NYU's RESTfulHS web-service. You will need to obtain the password from DLTS personnel if you want to use the RESTfulHS web-service for URI creation. You can also select “Read From File”, to read URIs from the work order file directly. If the import work order is a “barcode” or “file list”, then you can also select “Default Handle URL + Filename” option. This will generate URI by combining the text in the “Default URL” field with the imported filenames. The other choices in the drop-down menu are for testing purposes only, and should not be used in production work.

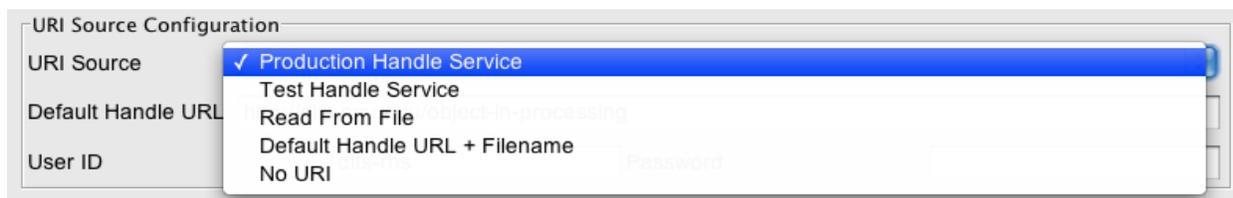


Figure 6

## Importing a File List

Another option for creating and attaching digital objects, is to import a text file containing a list of filenames (one filename per line). However, in order for this to work, the filenames must contain either the instance's container indicator 1,2,3, or barcode (or straight forward derivative in which a single character needs to be replaced), or the resource component's “Persistent Identifier” (i.e refNNN) or “Component Unique Identifier”. For example, if the container indicator 1 is 111-3456, then a filename of PHOTO\_111-3456m.tif, or PHOTO\_111\_3456m.tif, would work.

The header section of this file can contain text delimited with # character to provide information about what this file contains. The first none blank line following this section, contains information used during the import process and is in the following csv format.

“key by” (either instance1, instance2, instance3, barcode, componentID, or persistentID), “search and replace string” (none or i.e. -:\_), “option” (none or thumbnail), “file extension 1:use statement 1” (i.e.

m.tif:Image-Service), “file extension 2:use statement 2” (i.e. t.jpg:Image-Thumbnail),..., “file extension N:use statement N”

The “key by” option instance1, specifies that resource components should be keyed by "Container Indicator 1", while instance2, and instance3 specifies "Container Indicator 2", or "Container Indicator 3" respectively. The “key by” option barcode, specifies resource components should be keyed by the “Container Barcode” of the analog instances attached to them. The “key by” option componentID specifies resource components be keyed by “Component Unique Identifier”, while the option persistentID specifies they are keyed by their “Persistent ID”. The "thumbnail" option will create a thumbnail digital object using the URI of the service image + "?urlappend=/mode/thumb" by default. The append url can be specified by including it as part of the thumbnail option i.e. "thumbnail:?imageType=thumb". As many “extensions:use statement” pairs can be specified as needed.

## ADDITIONAL FUNCTIONS

### Exporting Digital Object URI List

A tab delimited file containing the URI for digital objects linked to the resource record can be generated by using the “Export Digital Object URI List” button. The format of this information can be seen in figure 7. The “Add Title to Export” checkbox can also be used to add the digital object title if desired.

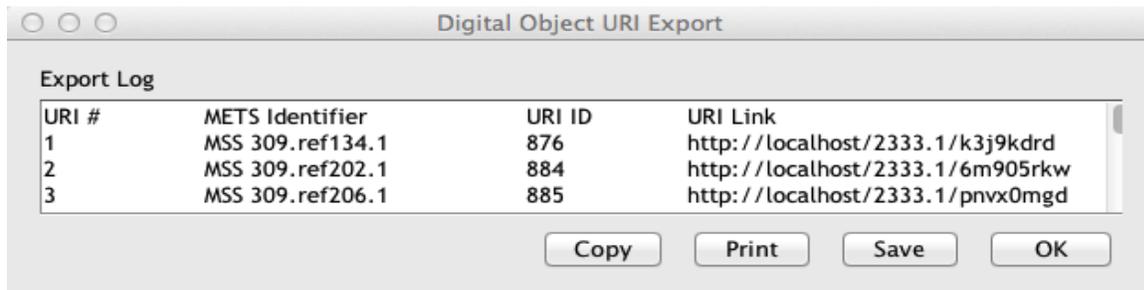


Figure 7

### Import Digital Object URI List

Using the tab delimited file created above, the current URIs of digital objects in the resource record can also be replaced with new ones. The format of this import file is the same as above, except for the addition of a column named “New URI Link”. As the name implies, this column contains the new URIs which will replace those already in the digital object (see figure 8).

URI #	Digital Object Title	METS Identifier	URI ID	URI Link	New URI Link
1	Interview 1	MSS 309.ref134.1	876	http://hdl.handle.net/2333.1/k3j9kdrd	http://localhost/2333.1/k3j9kdrd
2	Interview 2	MSS 309.ref202.1	884	http://hdl.handle.net/2333.1/6m905rkw	http://localhost/2333.1/6m905rkw

Figure 8

## Unlinking Digital Objects

It may sometimes be necessary to unlink all digital objects from resource record, for example, to update those digital objects with new URIs. To use this functionality, simply select the resource component, or the top-level resource record from which digital objects should be unlinked, launch the DO Workflow Dialog, then press the “Unlink Digital Objects” button. There is also a check box that can be select to test the unlinking process, without actually unlinking any digital objects. **Please note, that once unlinking is done, there is no way to relink the digital objects in an automated manner: they can be relinked manually.** Also, the digital object that are unlinked will still remain in the database, and will have to be deleted manually if they are no longer needed.

## CLEANING BARCODES

Due to the way barcodes stored in the “Container Barcode” field are encoded in finding aids when exported from the AT, re-importing those finding aids will cause the barcodes to be placed in the wrong location (not in the “Container Barcode” field). The “Barcode Cleaner” functionality will iterate through all the analog instances, and place the barcodes back in the “Container Barcode” field. This functionality is especially useful for people who do batch search and replace on finding aids, then re-import them into the AT.

## SETTING UP THE DEVELOPMENT ENVIRONMENT

This assumes you already have basic knowledge of how to compile and run Java programs using your IDE of choice. Once the source have been imported into an IDE project, add all the jar files in the lib folder of the Archivists' Toolkit installation folder. Doing so should take care of any dependency issues. Please note that in order to edit the user interface source code files, JFormDesigner is needed. This can be done by hand, but it will be a lot more difficult.

## FILE LAYOUT DESCRIPTION

"makep" - This is a simple batch script that packages up the class and xml files in a zip file that the JPF (Java Plugin Framework) can load when the AT starts up. It will need to be modified to work in your particular development environment. The output directory where your IDE places the class files need to be specified, and the location of the plugin folder of the AT also needs to be set.

"plugin.xml" - Describes information about the plugin and associated libraries for the JPF.

"src" - Contains the source code and all associated files. In this case, associated files are the GUI description files (\*.jfd) used by JFormDesigner. JFormDesigner is needed to modify the user interface code. Look at comments in source files to to see what each file does.