DRS Batch Processing

# Overview

This document describes the BDRC process which submits documents to Harvard University Library (HUL)’s Digital Resource Service (DRS). The DRS provides its users with interfaces to add, edit, delete and maintain materials. The process formats BDRC works so that the retriever can view them through Harvard’s customized interface which provides paging through the work. This is called the PDS (Page Delivery Service).

The process flowchart is shown in this figure:

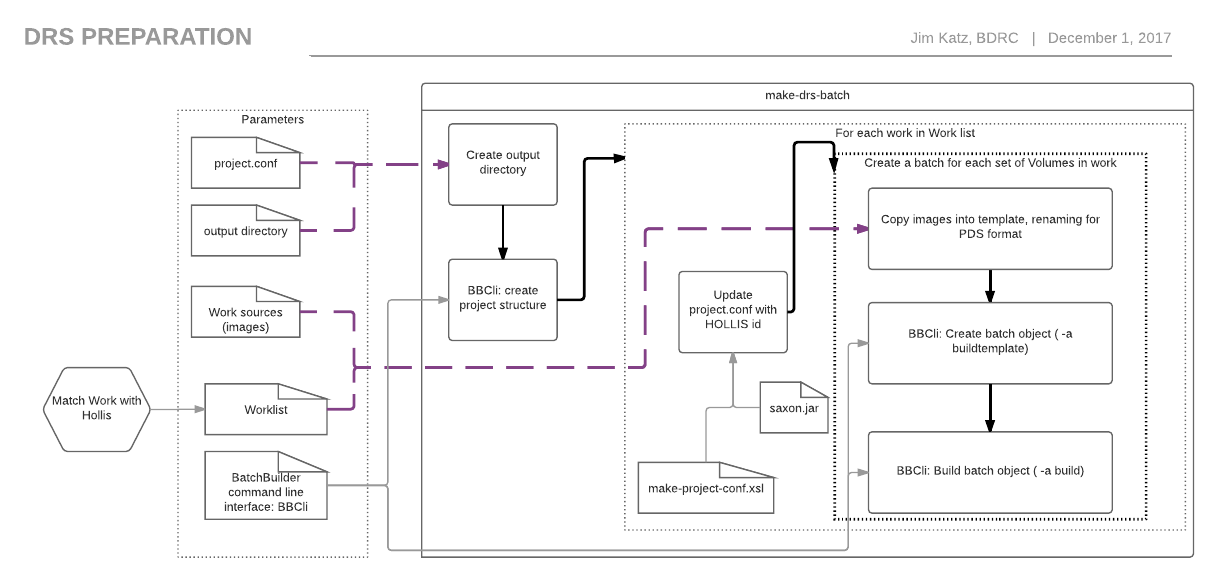


The process has two main phases:

* Preparation
* Submission

# Preparation

The end result of the preparation is a set of folders on disk which contains a DRS “batch” structure. This structure contains both the media to be uploaded (the images in the BDRC library) and the metadata to identify this media to Harvard’s repository. The process described below creates this complete structure.

The basic workflow of preparation is summarized in the following chart. [](https://www.lucidchart.com/documents/edit/5bed4858-2493-4e4e-972f-682e04d2cb6d/0?callback=close&name=docs&callback_type=back&v=1149&s=612)

Implementation details can be found [at the BDRC DRS Project Wiki Page](https://github.com/BuddhistDigitalResourceCenter/drs-deposit/wiki). This wiki’s audience is primarily technical: implementers and supporters.

The main components of the diagram are:

## Parameters

These are directories and files which are the parameters to the the make-drs-batch process. They are:

* A dictionary which matches BDRC Work Accession numbers to HOLLIS ids (Worklist)
* A pre-existing “project.conf” file. This is a template which all works can use to create batches.
* BatchBuilder Command Line interface (BBCli): this is a Harvard supplied program which creates the batches.
* The path to the output folder.
* The parent of all the Works which this process archives. The child folders this parent must have the same name as the dictionary’s BDRC Work Accession number.

## Batch structure

The end result of this process is a **project**. A project, in Harvard DRS terms, is a directory which contains one or more **batch** objects. A **batch** is the smallest. self-contained set of documents and metadata which the Harvard DRS can ingest at one time.

A batch contains one or more **objects**. Each object’s structure is determined by the batch’s **content model**. The content model of these batches are **Page Delivery Service.** This content model allows users to view each volume as a collection of pages, which can be navigated in the user’s web browser.

## Processing

1. Create the output batch directory - copy project.conf into it
2. Call BBCli to set up the batch directory structure
3. Loop over each work in the dictionary. For each one:
   1. Update the project.conf with the HOLLIS number of the work
   2. Loop over each set of (up to) 30 volumes in the work. For each volume:
      1. Copy the the volume’s images into the batch template location.
      2. Rename them on the way in to support Harvard’s PDS (Paged Delivery Service) format.
      3. Create a new batch in the project, using the items in the template.(BBCli ‘buildtemplate’ action. This action creates ‘object.xml’ descriptor files in the batch structure.
      4. Build the batch. This process populates the batch structure with an XML file (batch.xml) which describes all the objects in the batch. This process also pulls in data from Harvard sources:
         1. Premis: external information about the object.
         2. HulDrsAdmin: other information which Harvard uses to categorize the object.

This process is lengthy because the process has to make several web calls for each image to fill in this data.[[1]](#footnote-0)

## Batch Submission ftpScript.sh

The bash script ftpScript.sh uploads the batch directories in a project (and only the batch, not the supporting structures, such as \_aux and template). It uses a script-enabled ftp client , the standard Linux/MacOSX command sftp.

The script has to prevent the DRS ingestion process from starting before the upload is complete. DRS ingestion watches each user’s incoming/ folder for the presence of a file named batch.xml. When the file appears, DRS begins the ingestion process, without regard to the presence of all the files in the project.

To prevent this, just before upload, ftpScript.sh renames any file in the batch named batch.xml to batch.xml.wait. After the upload, the sftp script renames that batch’s batch.xml.wait to batch.xml. [[2]](#footnote-1)

1. In practice, this delay is negligible, 300msec / file. Also, the batch builder program multithreads. [↑](#footnote-ref-0)
2. Sftp only executes a command if the previous command has succeeded. [↑](#footnote-ref-1)