DLG AIP creation workflow for AIPs to be ingested by ARCHive system

using AIP Creation Helper v 2.0

Revised 10/24/2018

NOTE: These instructions assume you are working on a Windows PC (tested with Windows 7). You must have the following software applications installed and available on your system path so that they can be executed from the command line:

Software	Version	URL
JAVA	8 (1.8)	https://java.com/en/download/
Perl	5.16	https://www.perl.org/
Python	3.x	https://www.python.org/
FITS (File Information Tool Set)	1.3x	https://projects.iq.harvard.edu/fits/home
7zip	17	https://www.7-zip.org/
Bzip2 for Windows	1.0.2	http://gnuwin32.sourceforge.net/packages/bzip2.htm
Grep for Windows	2.5.4	http://gnuwin32.sourceforge.net/packages/grep.htm

1) Identify and stage digital objects to be included in AIP. As a general rule, DLG AIPS will be created at the digital object level, so usually there will be one item level record in dlgadmin per AIP. Generally AIPs will be named using record DLGadmin slugs or record IDs, however newspaper AIPs will be named using the batch name. Newspaper AIPs will be created at the batch level and will include the entire batch, including tif files.

Files should be unzipped. If files have been bagged previously, validate the bag before moving them into the prescribed AIP subfolder structure. Delete thumbs.db files.

Make sure you are at a location with enough space to accommodate duplication of the files necessary to create the AIP.

2) Rename files comprising digital objects and associated metadata to include DLG repository and collection codes ([dlgrepo]_[dlgcoll]_item).

3) Create the stub filesystem that will be transformed into the AIP. The following processes rely on a file system that strictly follows this pattern:

[aip id] parent folder

--**objects** subfolder

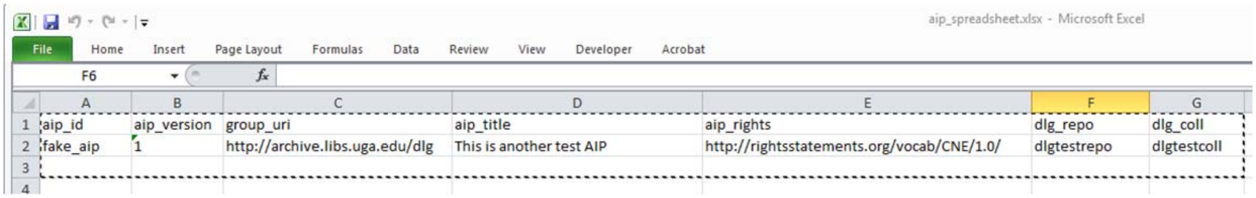
--[digital_object_id] Subfolder containing files. In most cases this is the same as the AIP ID.

In most situations, this can be done using one of the following Windows batch (.bat) files. Make sure that files and folders are named correctly before proceeding with any of the AIP creation steps:

noadmin_directories_AIP_stub_filesystem.bat	Use for a series of subfolders containing files for individual digital objects
noadmin_ind_AIP_stub_filesystem.bat	Use for a folder of files that correspond to a separate object.
news_AIP_creation_make_stub_filesystem.bat	Use for a series of newspaper batches (folders) that will each be archived as a separate AIP. This will result in a set of files in a folder named "rearranged". Use this as your working directory for all subsequent processes.

All subsequent steps will be executed in the directory containing these [aip id] subfolders. In these instructions, this will be called the *working directory*.

4) Create an XML file containing AIP level metadata. You can create this file using a spreadsheet and XML Blueprint to extract the XML from the spreadsheet. Save this file in your working directory. You can begin the spreadsheet with a TSV export of item level records from dlgadmin and modify it so that it follows this example:



	A	B	C	D	E	F	G
1	aip_id	aip_version	group_uri	aip_title	aip_rights	dlg_repo	dlg_coll
2	fake_aip	1	http://archive.libs.uga.edu/dlg	This is another test AIP	http://rightsstatements.org/vocab/CNE/1.0/	dlgtestrepo	dlgtestcoll
3							
4							

Then use XMLBlueprint or Excel to extract the xml from the spreadsheet.

The extracted XML file should have the following form:

```
<aip_level>
  <item>
    <aip_id>fake_aip01</aip_id>
    <aip_version>1</aip_version>
    <group_uri>http://archive.libs.uga.edu/dlg</group_uri>
```

```
     <aip_title>This is a test AIP</aip_title>
     <aip_rights>http://rightsstatements.org/vocab/CNE/1.0/</aip_rights>
     <dlg_repo>testrepo</dlg_repo>
     <dlg_coll>testcoll</dlg_coll>
   </item>
   <item>
     <aip_id>fake_aip02</aip_id>
     <aip_version>1</aip_version>
     <group_uri>http://archive.libs.uga.edu/dlg</group_uri>
     <aip_title>This is another test AIP</aip_title>
     <aip_rights>http://rightsstatements.org/vocab/CNE/1.0/</aip_rights>
    <dlg_repo>testrepo</dlg_repo>
     <dlg_coll>testcoll</dlg_coll>
   </item>
 </aip_level>
```

5) Move the AIP_creation_helper.zip file to your working directory and unpack it. Copy the current version of the AIP_creation_helper.zip file to your working directory. Right click this file and use 7zip to “unzip here”. If this is not an available option on your context menu, you can use any other unzip program for this step, but make sure you have 7zip correctly installed on your machine and also make sure that the unzipped files are directly in your working folder, not in a subfolder.

Unzipping the AIP_creation_helper_v2.0 will result in the following files being added to your working directory:

Name	Type
make_stub_filesystem	folder containing batch files to create stub filesystems
move_to_FITS_directory	folder containing batch file to be copied to FITS directory
1.0_move_fits_xml	Windows Batch File
2.3-big_combine_xml.bat	Windows Batch File
3.3-makemaster.bat	Windows Batch File
4.2-prep_bags.bat	Windows Batch File
7zprepare_bag.pl	PL File
bag_and_validate_all_folders_python.cmd	Windows Command Script

bagit.py	PY File
close.xml	XML File
dlg-fits-to-master-stylesheet.xsl	XSL Stylesheet
fart.exe	Application
open.xml	XML File
README_SIP_Creation.txt	Text Document
saxon9he.jar	Executable Jar File
split_TSV_source.xsl	XSL Stylesheet

6) Copy the runfits_external.bat file to the home directory of your FITS installation (where fits.bat is located).

You only have to do this one time and then you can re-use it whenever you create AIPs using this system. Double click to run the batch file. Enter the path (including drive letter) to your working directory when prompted.

7) Run the four .bat files to create the AIP.

Double click on each of the .bat files in numerical order to execute them. The second .bat file, 2.3-big_combine_xml.bat, will ask you to input the name of the xml file you extracted from the spreadsheet. You must enter this exactly as the file is named (case sensitive), including the .xml extension. Please be sure to follow any instructions given through the command prompt window. If you notice any errors you must investigate them and correct them before moving on with the next step or the process will not complete successfully.

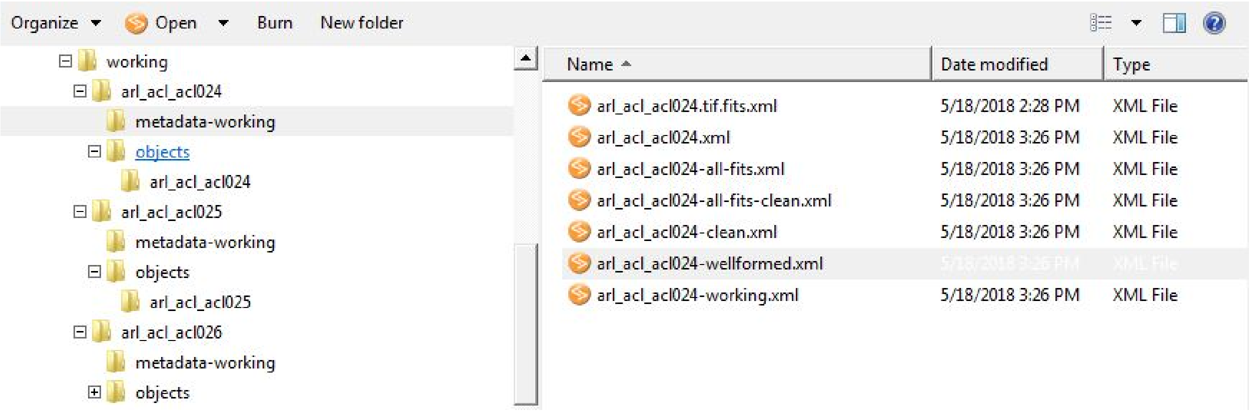
7a) Run the first .bat file, 1.x_move_fits_xml.bat

This step will create a metadata-working directory in the stub file system and move the fits.xml files to it.

7b) Run the second .bat file, 2.x-combine_xml.bat.

You will be prompted to enter the name of your AIP level xml file that you created in step 5. Be sure to include the .xml extension. This step will execute the following processes:

- Split the AIP level xml (extracted from the spreadsheet in step 5) into separate files for each AIP and move them to the appropriate metadata-working folder.
- Combine the .fits.xml files for each AIP into one xml file.
- Strip the xml declarations from the combined FITS xml.
- Regularize the file paths in the FITS xml so that it is relative to the “objects” folder of the filesystem.
- Combine the AIP level xml fragment with the combined FITS xml.
- Add an XML declaration, opening tag, and closing tag so that the resulting xml is well-formed.



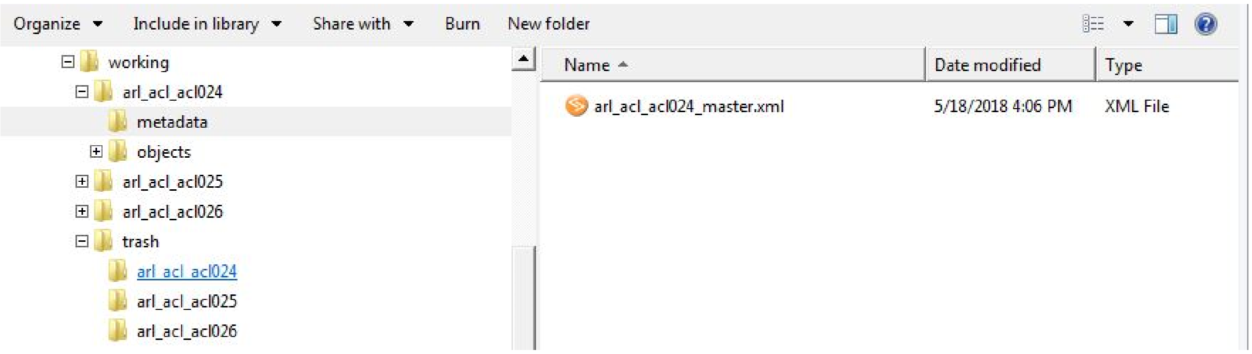
```
⊟ working
  ⊟ arl_acl_acl024
      metadata-working
    ⊟ objects
        arl_acl_acl024
  ⊟ arl_acl_acl025
      metadata-working
    ⊟ objects
        arl_acl_acl025
  ⊟ arl_acl_acl026
      metadata-working
    ⊞ objects
```

Name	Date modified	Type
arl_acl_acl024.tif.fits.xml	5/18/2018 2:28 PM	XML File
arl_acl_acl024.xml	5/18/2018 3:26 PM	XML File
arl_acl_acl024-all-fits.xml	5/18/2018 3:26 PM	XML File
arl_acl_acl024-all-fits-clean.xml	5/18/2018 3:26 PM	XML File
arl_acl_acl024-clean.xml	5/18/2018 3:26 PM	XML File
arl_acl_acl024-wellformed.xml	5/18/2018 3:26 PM	XML File
arl_acl_acl024-working.xml	5/18/2018 3:26 PM	XML File

Result of 2.x_combine-xml.bat

7c) Run the third .bat file, 3.x_makemaster.bat to transform each *_wellformed.xml file to *_master.xml

This step will create a “metadata” folder containing the [aip_id]_master.xml file and copy the contents of the metadata-working folders to a subfolder of the working directory called “trash”.



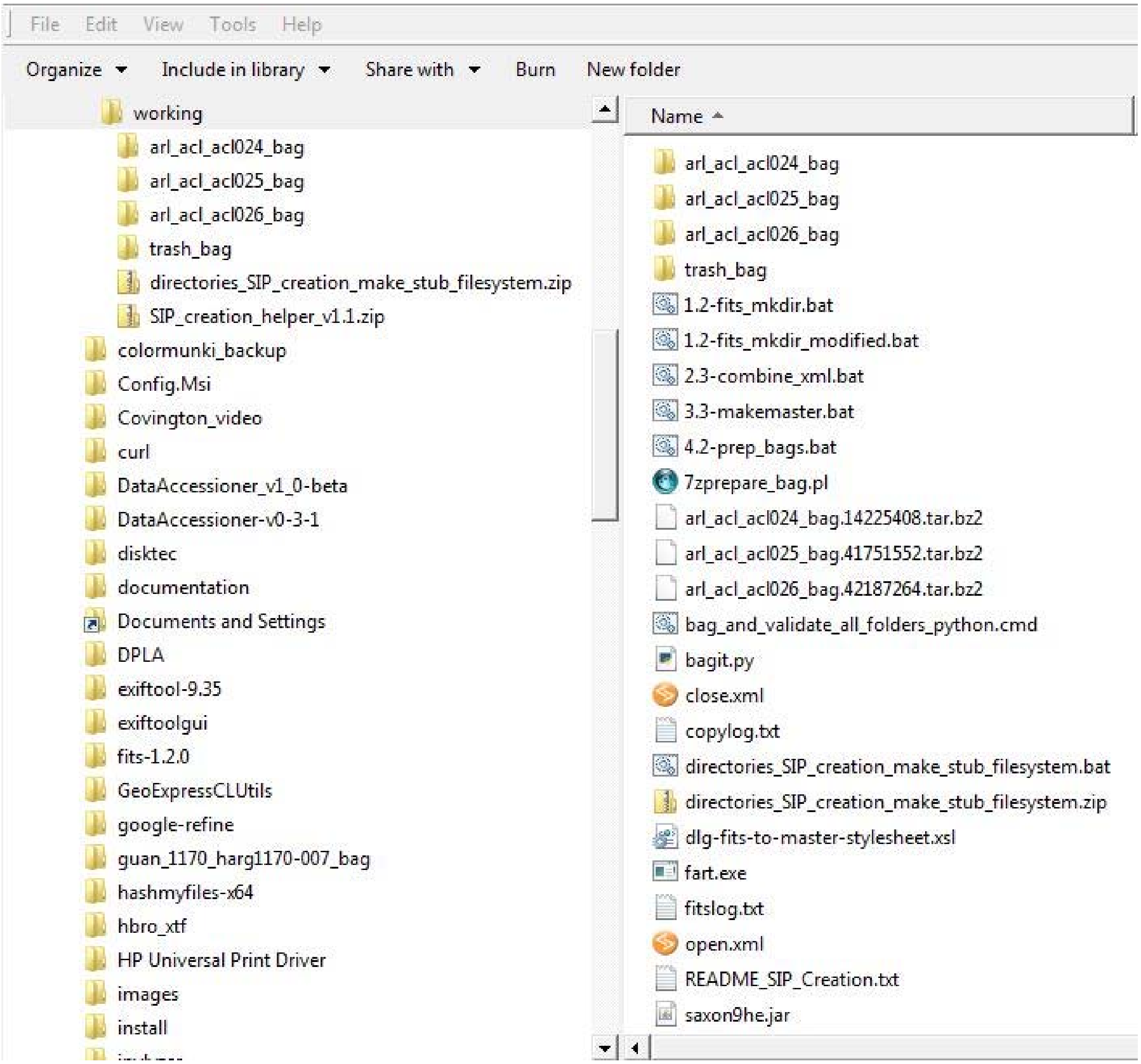
```
⊟ working
  ⊟ arl_acl_acl024
      metadata
    ⊞ objects
  ⊞ arl_acl_acl025
  ⊞ arl_acl_acl026
  ⊟ trash
      arl_acl_acl024
      arl_acl_acl025
      arl_acl_acl026
```

Name	Date modified	Type
arl_acl_acl024_master.xml	5/18/2018 4:06 PM	XML File

Result of 3x-makemaster.bat

7d) Run the fourth .bat file, 4.x-prep_bags.bat.

After completing this step, your original subfolders will have the suffix _bag appended to them. This indicates that the bagging step was successfully completed. There will also be a set of files ending in a string of numbers and the extension .tar.bz2. These are the completed AIPs that are ready for upload to ARCHive.



working
arl_acl_acl024_bag
arl_acl_acl025_bag
arl_acl_acl026_bag
trash_bag
directories_SIP_creation_make_stub_filesystem.zip
SIP_creation_helper_v1.1.zip
colormunki_backup
Config.Msi
Covington_video
curl
DataAccessioner_v1_0-beta
DataAccessioner-v0-3-1
disktec
documentation
Documents and Settings
DPLA
exiftool-9.35
exiftoolgui
fits-1.2.0
GeoExpressCLUtils
google-refine
guan_1170_harg1170-007_bag
hashmyfiles-x64
hbro_xtf
HP Universal Print Driver
images
install

arl_acl_acl024_bag
arl_acl_acl025_bag
arl_acl_acl026_bag
trash_bag
1.2-fits_mkdir.bat
1.2-fits_mkdir_modified.bat
2.3-combine_xml.bat
3.3-makemaster.bat
4.2-prep_bags.bat
7zprepare_bag.pl
arl_acl_acl024_bag.14225408.tar.bz2
arl_acl_acl025_bag.41751552.tar.bz2
arl_acl_acl026_bag.42187264.tar.bz2
bag_and_validate_all_folders_python.cmd
bagit.py
close.xml
copylog.txt
directories_SIP_creation_make_stub_filesystem.bat
directories_SIP_creation_make_stub_filesystem.zip
dlg-fits-to-master-stylesheet.xsl
fart.exe
fitslog.txt
open.xml
README_SIP_Creation.txt
saxon9he.jar

Result of 4x-prep_bags.bat

8) Prepare MD5 manifest using report from ExactFile or other program capable of producing an MD5 checksum. This manifest file should be a tab delimited text file containing the checksum and the name of the final zipped and tarred bag. If you are going to upload multiple AIPs in a batch, there should be one manifest for all of the files to be uploaded.