**Summary for AbID application development**

**AbID Enrollment log PDF scraper:**

The code for the standalone executable application for enrollment log pdf scraping is complete, but configuration may be required in the future. Most of the packages have been compiled in the ABID\_PDF\_to\_Exel.exe and will not need to be downloaded. However there are two requirements for this application to run.

1. Python 3 (preferably python 3.6 to 3.8) installed and PATH has been configured
2. Ghostscript url to download ghostscript: <https://www.ghostscript.com/download.html>

The application and its supporting folders are found here:

\USRED\_CLINICAL DATA\CLINTRIALS\_IN\_PROGRESS\IHD\IH-500 Antibody Identification\10 Data Management\DM Only\Code\Python\Python testing\

The executable file is here:

\USRED\_CLINICAL DATA\CLINTRIALS\_IN\_PROGRESS\IHD\IH-500 Antibody Identification\10 Data Management\DM Only\Code\Python\Python testing\dist\

The standalone executable python application searches for pdfs in two network folder locations:

root\_dir1="//global.bio-rad.com/groups/USRED\_CLINICAL DATA/CLINTRIALS\_IN\_PROGRESS/IHD/IH-500 Antibody Identification/08 Monitoring/Enrollment Monitoring/BLD PDFs/"

root\_dir2="//global.bio-rad.com/groups/USRED\_CLINICAL DATA/CLINTRIALS\_IN\_PROGRESS/IHD/IH-500 Antibody Identification/08 Monitoring/Enrollment Monitoring/BCW PDFs/”

A function scrapes the pdf data table and creates a dataframe. The dataframe is passed to two functions.

One of the functions creates an excel file. The excel file is placed in: USRED\_CLINICAL DATA/CLINTRIALS\_IN\_PROGRESS\IHD/IH-500 Antibody Identification/10 Data Management/DM Only/IHAB\_ELCombine.xlsx.

The other function sends the dataframe to the sql table dbo.tbl\_stg1\_EL in the IHAB database. This table is overwritten every time the script is run.

The excel file and pdf’s are moved to an archive folder with datetime stamp and a prefix is attached (arc). The file path to the archive folder. \USRED\_STORAGE\Clinicals\Data Management\IHD IH-500 ABID

When the script has successfully placed the data frame in the sql table a success message is written to a txt file as well as a tkinter success message pops up.

The success txt file is located: \USRED\_CLINICAL DATA\CLINTRIALS\_IN\_PROGRESS\IHD\IH-500 Antibody Identification\10 Data Management\DM Only\abid\_success.txt

The issue with the script is that IT’s antivirus software occasionally deletes the executable program and this is the reason why django development has started for this application in the ihab app. Permissions for the script to read into the network folders will be required. The django application cannot find //global.bio-rad.com/groups/ in the url. IT has not set it up for this to be discoverable in the browser.

There is another workaround which is to manually locate the files while in the django environment and scrape from there. The issue with this approach is that it will be slower and if there are two pdf’s the writing to sql will have to be amended to not overwrite if the insert date is the same during the day.

**Django applications:**

All django application code is located in the CentOS 7 linux box. /home/datamgt/repository/ihab/

**AbID django application (manual data entry):**

The Django web application ihab link has been commented out in the cdm\_home(the application dispatching app) application’s index.html page to ensure no data can be entered in the forms. Commenting this link out also prevents users from accessing the ihab metrics app as well.

The ihab application consists of 4 forms, bld primary, bld secondary, bcw primary & bcw secondary. These forms enter data into two tables in the IHAB database. This application is for the use of two CRA’s (clinical research associates) to manually enter data and submit into: dbo.stg1\_dataentryperson\_abid\_1 & dbo.stg1\_dataentryperson\_abid\_2 tables.

The ihab application in the django environment is complete. Ihab app has all database connections, templates, views and forms. These do not need to be reconfigured for the application to be functional when the time is required for its use.

**AbID metrics application:**

This is run by the bokeh library with data being pulled by django’s ORM. The view and template is still available in the ihab app. It would be useful to use it’s roughed out framework dashboard development during future studies.

All code for this application is still in development. Adding more charts to the page will take more development in the applications templates (html). Possible solutions would be to build out a grid pattern with css and html divs. Then when a new chart is needed it would be added in the div class that would provide the styling. This will need to be tested for compatibility.