

Introduction

See 'project_flow' file to get an overview of the entire automation flow.

Step 1:

There are two files:

- `Send2dps.py`: main python script
- `Util.py`: utility functions and helper methods used by `send2dps.py`

The python scripts in step 1 do the following:

1. Query `caps.qtr_dps_cvs` table, transform the data, and spit out two files (a txt file, a xlsx file).
2. Txt file is sent to DPS SFTP server.
3. Txt file and xlsx file are copied to shared network drive

Step 2:

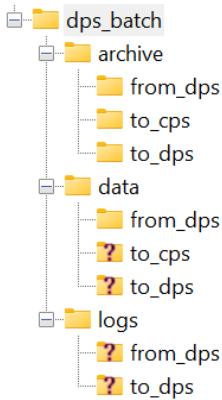
There are two files:

- `Receiveddps.py`: main python script
- `Util.py`: utility functions and helper methods used by `send2dps.py`

The python scripts in step 2 do the following:

1. Fetch two text files from DPS Server
2. One of the text files (`dfps-missing-person-result.txt`) is loaded into a staging table in data warehouse through SQL*LOADER (then a pl/sql procedure loads the data in staging table into a permanent table).
3. Query data warehouse tables
 - a. `sa_sum`
 - b. `cnty_reg_tableau`
 - c. `sa_98`
4. Read the text file sent to DPS from step 1.
5. Do data transformation with data from 1,3, 4, and spit out three xlsx files
4. All three xlsx files are copied to shared network drive
6. All three xlsx files are archived to `archive/to_cps` directory on XDB1.
7. The two output files from steps are archived to `archive/to_dps` directory on XDB1 (they are archived in step 2 because they are needed in step 2 and can't be archived in step 1).

Directory structure on XDB1



Testing

1. Set up a python virtual environment on XDB1 in my home directory at: /home/fuh2
 - `python3 -m venv .venv`
2. Install python libraries needed:
 - `pip install library_name`
3. Upload all python scripts, config and control files to my home directory.
4. Make sure all python scripts are executable (usually are by default). If they are not, use
 - `chmod +x send2dps.py`
5. Execute python scripts using:
 - a. `python3 send2dps.py dev`
6. Compare the output from python with output from R script.
7. I also checked whether files are uploaded to DPS server, network drive, archived into proper directories on XDB1. All passed.

Files:

Main_code

1. send2dps.py – main script used for step 1
2. receivedps.py – main script used for step 2
3. util.py - utility function used for step 1 and 2
4. dps_config_test.yml – YAML configuration file for directories, network drive, database username and passwords, SFTP credentials, etc.
5. dps_results.ctl – Control file used by SQL*LOADER to load flat file into staging table in DW.
6. stg_dps_results.sql – DDL script for creating staging table in DW.
7. send2dps.sh – shell script to kickstart step 1 by calling send2dps.py (will be automated through cron job)
8. receivedps.sh – shell script to kickstart step 2 by calling receivedps.py (will be automated through cron job)