

**Office of the Chief Risk Officer (OCRO)
Grant Portfolio Risk Assessment Development and
Project Tracking and Reporting**

Automated Data Load Guide

IMPLEMENTED AND PREPARED BY:

Dagnaw Amare

Version 18.0

OCRO Grant Portfolio Risk Assessment Automated Data Load Guide

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Description of Change
1.0	Dan Amare	02/01/2022	Craig Pascoe	02/08/2022	<i>Initial version</i>
2.0	Dan Amare	2/18/2022	Craig Pascoe	02/25/2022	Add OIG, Mon Ref, IPERA, TrueScreen, & Fieldprint
3.0	Dan Amare	3/7/2022	Craig Pascoe	03/07/2022	Add Grant Review, PII Breach, Caution List, FAPIIS & IRS990
4.0	Dan Amare	5/24/2022	Craig Pascoe	05/25/2022	Add SAM Registrations, Exclusions files
5.0	Dan Amare	7/5/2022	Vanaja Jale	7/8/2022	Add SAM API, update FAPIIS for DUNS to UEI File
6.0	Dan Amare	8/1/2022	Vanaja Jale	8/4/2022	Add USA Spending API
7.0	Dan Amare	8/31/22	Vanaja Jale	9/7/2022	Update to IRS990 from IRS.gov
8.0	Dan Amare	10/31/22	Vanaja Jale	11/7/2022	Update FMS logic, FAPIIS automation
9.0	Dan Amare	11/22/22	Vanaja Jale	11/28/2022	Password files
10.0	Dan Amare	2/28/23	Vanaja Jale	3/1/2023	IRS990 data from ProPublica, Update RF 82, 86, 88, 89, 50 logic, New RF 90, Updates for True Screen and Field Print and OIG
11.0	Dan Amare	4/4/23	Vanaja Jale	4/5/2023	FAPIIS change to SAM.gov Entity API
12.0	Dan Amare	6/12/23	Vanaja Jale	6/13/2023	FAC GSA API implementation, IRS990 new filings 2021-2023
13.0	Dan Amare	9/18/23	Vanaja Jale	9/25/2023	FAC 4 new data loads, fiscal year update new script
14.0	Dan Amare	10/24/2023	Vanaja Jale	10/24/2023	FAC GSA API data load
15.0	Dan Amare	11/21/2023	Vanaja Jale	11/27/2023	FAC GSA API rollout fields
16.0	Dan Amare	1/24/2024	Vanaja Jale	1/29/2024	Code Refactoring, process excel files
17.0	Dan Amare	3/18/2024	Vanaja Jale	3/25/2024	FMS import to insert-only

**OCRO Grant Portfolio Risk Assessment
Automated Data Load Guide**

18.0	Dan Amare	4/22/2024	Vanaja Jale	4/23/2024	New Debt tracker data load, FAC perf improv
------	-----------	-----------	-------------	-----------	---

Table of Contents

Document Purpose	3
Requirements.....	3
File Preparation	4
Python Environment.....	4
Automated Data Load Process	5
Python Script Execution.....	6
Job Status Messages.....	7
Data Verification	8
Maintenance	8
Enhancements	11

Document Purpose

This document provides guidance on the automated data load process and associated scripts, configuration files and libraries. The manual data load process was found to be cumbersome and lacked error checking and other data transformation capabilities. The Python script automates most of the manual import steps and adds error handling, additional transformations, and logging capabilities. The data files still need to be downloaded from their respective sources and placed into a Data folder. New functionality has been added to make use of APIs to request and import data from some data sources like SAM.gov, FAPIIS, USA Spending, and IRS.gov.

Requirements

Users need to have the following software:

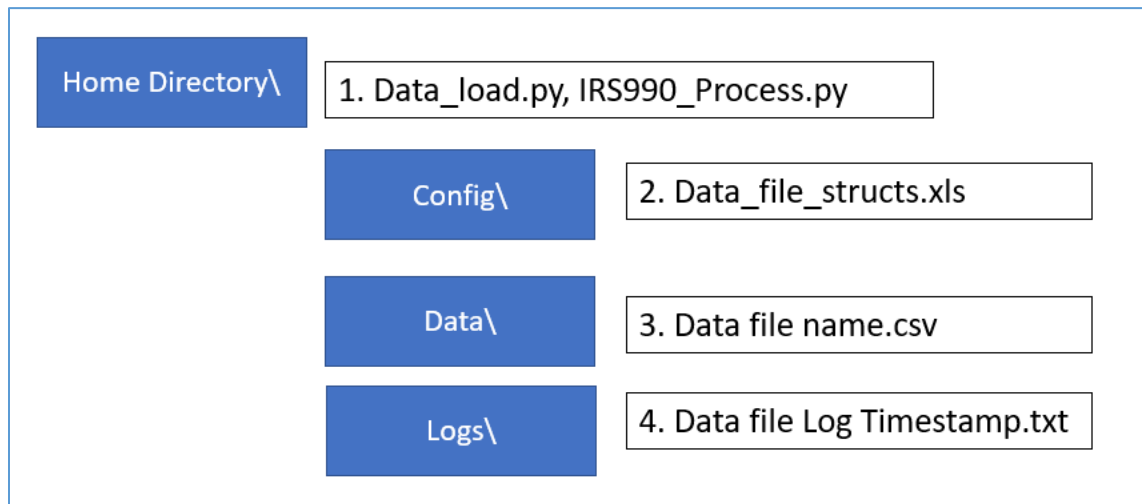
- Python 3.9.5 (and libraries listed in Appendix C)
- Oracle 64-bit client files (Available for download [here](#))
instantclient-basic-windows.x64-21.3.0.0.0.zip
- Oracle SQL Developer tool for data load verification.
- Users need to have access to ARESFQT/ARESPROD database with Select/Update/Insert privileges in ARES schema (DML privileges).
- Python Script (zipped file in JIRA ticket, [CR-1380](#)), **Data_load.zip**.
- Python Script for APIs (zipped file in [CR-1584](#)), **Data_load_API.zip**.
- API keys file (zipped file in [CR-1584](#)), **api_key.txt**.

File Preparation

The file to be imported into the table is expected to be in excel format as of sprint 34. Previously, this needed to be CSV file, but this requirement is no longer needed.

Python Environment

The Python script environment needs to be set up as seen below.



Home Directory: Base directory where the data load Python script resides.

1. HOME\Config: Sub folder under HOME where configuration file is kept.
2. HOME\Data: Sub folder under HOME where CSV data files are expected to be.
3. HOME\Logs: Sub folder under HOME where log files are stored.

Steps:

1. Download the Python script and associated files from [CR-1380](#), **Data_load.zip**.
2. Create a HOME directory and extract the zipped file to it. This will place the Python scripts in HOME directory and create the subfolders underneath HOME as shown in above diagram.
3. Additionally, the oracle 64-bit client software needs to be in following location on local drive.
C:\oracle\Product\instantclient_21_3
 - This oracle client software is available for download [here](#). Download the file, **instantclient-basic-windows.x64-21.3.0.0.0.zip**, and extract to **C:\oracle\Product**. These files allow Python to use oracle libraries to connect to Oracle.
4. The following Python library is also needed to read the configuration EXCEL file.
xlrd
 - This can be installed from the command line using PIP software as described in Appendix C, Section 1.4.
pip install --trusted-host pypi.org --trusted-host files.pythonhosted.org xlrd

Automated Data Load Process

In general, the Python script handles the process of loading data files into their respective tables, removing bad records, limiting data loaded to only data needed for RA process, and logging results into log files. The script has evolved into requesting data from APIs as in the case of SAM.gov and USA Spending, and automatically downloading and processing data from websites as in the case of IRS990 obtaining data from IRS.gov (initially from AWS, see Appendix A.).

Python Script (Data_Load.py)

The script expects the configuration and data files to be available in locations below.

1. Configuration file:

This will automatically be available in Config folder upon extraction of Python zipped file.

It is the Excel file containing list of fields and their sequence for each data file type.

Configuration file Name: HOME\Config\Data_file_structs.xls

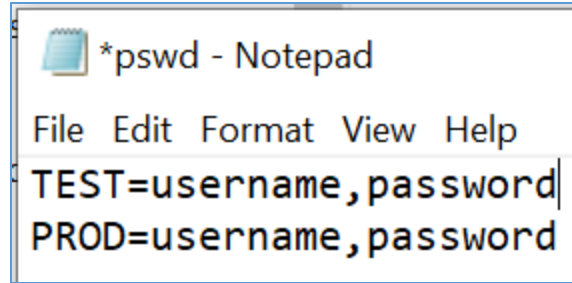
Configuration file structure: One excel sheet/tab for each data file type (FMS, FAC, OIG etc)

	A	B
1	APPLICATION_ID	
2	ORGANIZATION_LEGAL_NAME	
3	CITY_STATE_ASSOCIATED_WITH_EIN	
4	CFDA_NUM_ASSOCIATED_WITH_APP	
5	DUNS_NUM	
6	EIN	
7	PP_PERSONNEL_EMPLOYEE_HANDBOOK	
8	PP_FIN_INTERNAL_CONTROLS	
9	PP_SUB_AWRD_MONITOR_AND_OVRST	
10	PP_TIMEKEEPING	
11	PP_TRVL_GUIDANCE_CRDT_CRD_USE	
12	PP PROCUREMENT	
	FMS	FAC USAS OIG MONREF IPERA

Configuration file structure

2. Configuration file - Password file.

New addition to Config files is the password file which contains database credentials in format shown below. This password file (pswd.txt) will need to be updated with latest password for the relevant environment.



Password credentials file

3. Data file:

This is the data file to be loaded, expected to be excel format. As of sprint 34, files don't need to be CSV.

Data file name: HOME\Data\Data file name.xlsx

Data file structure: This must have same column structure and sequence listed in configuration files and database tables. See [Maintenance section](#) for details.

Python Script Execution

The script needs to be executed from windows command line. Click on search lens next to Windows icon and search for Command Prompt and select Command Prompt from the list. On command line, navigate to the folder where the python script resides. To navigate to a folder via command line, use 'CD' command which stands for 'Change Directory'. The folder name to navigate to can be relative to current default directory or listed as full path starting with 'C:\' directory. See the two examples below.

```
C:\Users\damare>cd documents\python  
  
C:\Users\damare\Documents\Python>cd C:\Users\damare\Documents\Python  
  
C:\Users\damare\Documents\Python>
```

The python script takes command line parameters to load data into appropriate table and database environment. To run the script, use the following format and parameters in the order shown. 'HOME' is just a placeholder for the location where the script resides. It does NOT need to be typed. In the above screenshot, HOME = C:\Users\damare\Documents\Python. But if already in this directory, only the script name and parameters need be typed as shown below.

HOME\Python Data_load.py *[Environment] [Data file type] [Data file name]*

- **Python:** This just tells Windows to use Python software
- **Data_load.py:** Python script that parses the excel file, extracts data, connects to oracle, and imports data and finishes by updating metadata date fields and creating log files.

- **Environment:** The value expected for this is either ‘Test’ or ‘Prod’, case insensitive, no quotes
- **User/Password:** This should have ‘username’/‘password’, case sensitive, no quotes required. –**DEPRECATED TO USE PASSWORD FILE in Config directory.**
- **Data file type:** Values expected are data file type in all caps, FAC, FMS, USAS etc. This must match the sheet name in the configuration file for the respective data file.
 - a. **File types = [FAC, FMS, USAS, OIG, MONREF, IPERA, TRUESCREEN, FIELDPRINT, REVIEW, OIT, CAUTION, FAPIIS]**
- **Data file name:** This needs to be the name of the data file to import and must be in Data subfolder. ***This needs to be in double quotes.***

Example:

Python Data_load.py Test FMS “FMS 02-01-2022.xlsx” [Hit Enter]

Job Status Messages

While script is running, it will display errors or informational messages as appropriate on terminal screen and in timestamped log files as needed.

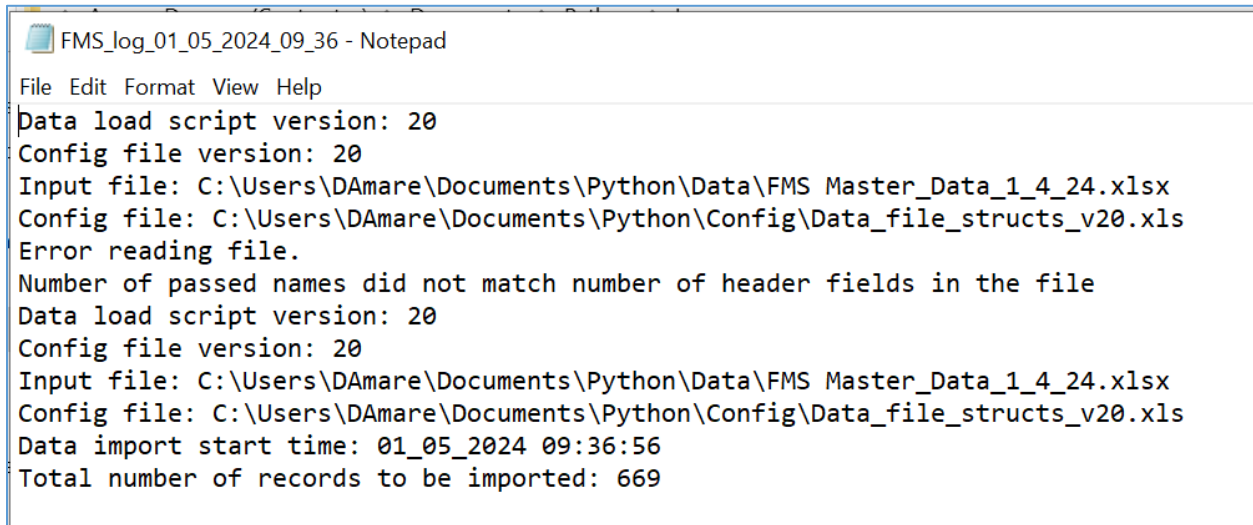
Error message types:

- Invalid format, meaning number of parameters is not 5, not counting ‘Python’.
- Environment name is not ‘Test’ or ‘Prod’
- Invalid username or password
- Incorrect data file format (FMS, FAC, USAS, OIG, etc)
- Configuration file not found in HOME\Config subfolder, Data_file_structs.xls.
- Data file indicated is not found in HOME\Data subfolder.
- Connection error to Oracle
- Other errors specific to data file such as file format and content is not as expected.

Information messages:

- Names and locations of config and data files found.
- Data import step start and completion times.
- Data import progress
- Data import summary
- Bad records skipped messages.
- Information messages are also logged to timestamped text files in HOME\Logs folder.

Example:



```
FMS_log_01_05_2024_09_36 - Notepad
File Edit Format View Help
Data load script version: 20
Config file version: 20
Input file: C:\Users\DAmare\Documents\Python\Data\FMS Master_Data_1_4_24.xlsx
Config file: C:\Users\DAmare\Documents\Python\Config\Data_file_structs_v20.xls
Error reading file.
Number of passed names did not match number of header fields in the file
Data load script version: 20
Config file version: 20
Input file: C:\Users\DAmare\Documents\Python\Data\FMS Master_Data_1_4_24.xlsx
Config file: C:\Users\DAmare\Documents\Python\Config\Data_file_structs_v20.xls
Data import start time: 01_05_2024 09:36:56
Total number of records to be imported: 669
```

Log File

Data Verification

As an optional step, once the file has been imported using the Python script, users can verify the data loaded by using Oracle SQL Developer to run SQL scripts to compare counts of records imported versus records in the data file. However, for some data files, the count of data imported may not match number of rows in the data file. For example, USA Spending limits the data imported to only Duns number in monitoring and application universes and excludes previously loaded data as well as fiscal years beyond the last 3 years. FMS also has update/insert logic whereby only new data is inserted while existing data is updated.

Maintenance

If there's a need to update the data structure of files to import, refer to following two types of changes that'll require different way of handling.

1. New fields to import:

These require not only database changes to add the new field(s) but also changes to the RA package to process the new field(s). In addition, the configuration file needs to be updated.

2. Names and sequence of fields:

Names and sequence of fields are kept in the configuration file in their respective data sheet. If column names or sequences need to be changed, the appropriate sheet needs to be updated with the new names and sequences. There're two types of data files.

2.1 All files except for USA Spending & Grant Review

Columns imported are consecutively placed starting from first column and these correspond to database fields as listed in configuration file. Therefore, their position is important, but column name changes doesn't matter as their position matches them

OCRO Grant Portfolio Risk Assessment Automated Data Load Guide

to database fields and not column names. If the column position shifts however, this will load next column over to wrong database field so the corresponding database field in configuration file also needs to shift to match position in the file. Only the database fields are listed in the configuration file in the consecutive order they're expected; no mapping to column name is needed as ordering is implicit. See example below.

	A	B	
1	APPLICATION_ID		
2	ORGANIZATION_LEGAL_NAME		
3	CITY_STATE_ASSOCIATED_WITH_EIN		
4	CFDA_NUM_ASSOCIATED_WITH_APP		
5	DUNS_NUM		
6	EIN		
7	PP_PERSONNEL_EMPLOYEE_HANDBOOK		
8	PP_FIN_INTERNAL_CONTROLS		
9	PP_SUB_AWRD_MONITOR_AND_OVRST		
10	PP_TIMEKEEPING		
11	PP_TRVL_GUIDANCE_CRDT_CRD_USE		
12	PP PROCUREMENT		
	FMS	FAC	USAS OIG MONREF IPERA ... (+)

2.2 USA Spending & Grant Review/Prohibited Activities

Columns imported are not consecutively placed so column names, not positions, are used to select the columns to load to a database field. Position is not significant in this case but column name is so its name must map to corresponding database field as listed in configuration file. No update needs to be made if column position shifts in the data file. Update needs to be made to configuration file if column name changes for corresponding database field mapped to it. See example below. First column in configuration file lists database fields, and second column lists corresponding column names in the data file. **Column names are CASE SENSITIVE and need to be listed exactly as written in the data file.**

	A	B	
1	PRIME_AWARD_FAIN	prime_award_fain	
2	PRIME_AWARD_AMOUNT	prime_award_amount	
3	PRIME_AWARD_FISCAL_YEAR	prime_award_base_action_date_fiscal_year	
4	PRIME_PRD_OF_PERF_START_DT	prime_award_period_of_performance_start_date	
5	PRIME_PRD_OF_PERF_END_DT	prime_award_period_of_performance_current_end_date	
6	PRIME_AWARD_AWARDING_AGENCY_NM	prime_award_awarding_agency_name	
7	PRIME_AWARDEE_DUNS	prime_awardee_duns	
8	PRIME_AWARDEE_NAME	prime_awardee_name	
9	SUBAWARD_AMOUNT	subaward_amount	
10	SUBAWARDEE_DUNS	subawardee_duns	
11	SUBAWARDEE_NAME	subawardee_name	
12			

Enhancements

These are suggested future enhancements to the data automation script:

1. Schedule the job to run daily and look for new data files in specific shared folders and email error and log files to users.

