Description:

We periodically receive Security and Pricing reference files from our data vendors. We process these files and import them into our Security Reference System. The system stores data in a database, but for this assignment we will work with ‘reference’ files only.

Inputs:

1. Example of Corporate Bond / Preferred Equity Vendor reference file: **corp\_pfd.dif**

References:

1. In-use field reference file: **reference\_fields.csv**
2. Complete Security Reference file: **reference\_securities.csv**

Outputs:

1. New Security Reference file:  **new\_securities.csv**
2. Field-Value Reference file: **security\_data.csv**

Input file is structured as follows:

* Column names are stored in a block between lines: START-OF-FIELDS and END-OF-FIELDS
* Data is stored between lines: START-OF-DATA and END-OF-DATA
* Data is pipe ‘|’ delimited

Steps:

1. Read Security Reference input file and convert it to DataFrame using column names supplied in input file: **corp\_pfd.diff**
2. Limit columns in DataFrame to only those found in **reference\_fields.csv**
3. Compare securities in input file with **reference\_securities.csv file**. Use *ID\_BB\_GLOBAL* as the unique key. Create **new\_securities.csv**, which should include all securities NOT found in **reference\_securities.csv**. **new\_securities.csv** structure should match **reference\_securities.csv** (Note. if *ID\_BB\_GLOBAL* is NOT unique in input file – use first available row with same *ID\_BB\_GLOBAL*)
4. Create **security\_data.csv**  (comma ‘,’ delimited) with the following structure:

|  |  |
| --- | --- |
| Column | Description |
| ID\_BB\_GLOBAL | Unique Securiti |
| FIELD | Field Name from input DataFrame (all fields listed in reference\_fields.csv) |
| VALUE | Field Value from input DataFrame |
| SOURCE | Input File name: corp\_pdf.diff |
| TSTAMP | Date/Time when result was processed |

Example:

A screenshot of a computer

Description automatically generated

1. Commit your code/configuration to a new Github project. Provide Email link to the project when completed