**ETLDashboard Documentation:**

* Repository: [R:\Enterprise Risk Dashboard\Repo\ETLDashboard](file:///R:\Enterprise%20Risk%20Dashboard\Repo\ETLDashboard)
* Scripts used in application:
  + **GenerateNewETL.py:**

1. Reads one or more files in folder matching regular expression containing data that you wish to implement in a new ETL.
2. Generates column report detailing appropriate T-SQL types, attributes, relationships between columns. If multiple files were read then details how columns have changed over time.
3. Generate .sql table definition based on column types using predetermined template (currently for use in MetricsDYETL).
4. Append new ETL to DynamicETL.Service appsettings-template.json file with default attributes.
   * **TestETLPipeline.py:** 
     + If testing in LOCAL:
       1. Post arguments using postargs.json to local instance of DynamicETL.WebAPI.
       2. Run DynamicETL.Service to push data into target table.
       3. Query ETL’s T-SQL table for inserted data and generate report comparing inserted data with data in source file, detailing specific differences between rows.
     + If testing in QA/UAT/STG:
       1. Using file configured in DynamicETL.WebApi postargs.json, drop file to ETL’s configured FileWatcher location.
       2. Wait for file to be sucked up by ETL.
       3. Query ETL’s T-SQL table for inserted data and generate report comparing inserted data with data in source file, detailing specific differences between rows.
   * Every script has .json file with same name as script (ex: GenerateNewETL.json, TestETLPipeline.json).

* **Software Requirements:**
  + Python >= 3.6
  + Packages listed in requirements.txt.
* **Command line usage:**
  + “python <PathToScript.py>”

**Script Arguments:**

* **GenerateNewETL.py:**
  + Required Arguments (in GenerateColumnAttributesReport.json):
    - **“data”**: Dictionary containing following keys:
      1. **“path”**: Path to folder containing one or more files of different dates but same dataset.
      2. **“sheets” (Optional):** List of strings containing sheets you wish to analyze. If used then “path” must point to an Excel file, and each sheet will be viewed as its own ETL.
    - **“reportpath”:** Pathto generated report detailing column attributes. Must point to Excel file.
    - **“filedatereg”:** Dictionary containing following keys:
* **TestETLPipeline.py:**
  + Required Arguments (in TestETLPipeline.json):
    - **“etlname”:** Name of ETL. Must be configured in DynamicETL.Service in QA/UAT/STG and in local copy.
    - **“filedate”**: Date of file used to test pipeline (listed in **postargs.json** or in optional **“comparefile”).**
    - **“reportpath”:** Path to output report comparing input data and output data. Must be an .xlsx file.
    - **“postargspath”:** Path to **postargs.json** file containing REST arguments POSTed to DynamicETL.WebAPI. Required whether testing locally or QA/UAT/STG.
    - **“testmode”:** One of LOCAL/QA/UAT/STG (case insensitive) denoting which version of the ETL you are testing.
  + Optional Arguments:
    - **“pkey”:** List ofstrings or string denoting which columns you want to use to compare two rows, i.e. the primary key of the dataset.
    - **“ignorecols”:** List of strings or string denoting which columns you want to ignore when comparing two rows. FileDate and RunDate columns are always ignored.
    - **“comparefile”:** Path to file containing data you want to compare to data loaded into ETL table, in case you do not want to use the file listed in postargs.json.