**Daily Metrics DB Automation**

Last Updated: 2019.09.24

Objective: To produce the expected outputs of the Daily MetricsDB process without OA associate intervention.

Goals:

The automated reporting solution for the “MetricsDB” aims to be:

|  |  |
| --- | --- |
| **Goals** | **How they are met** |
| 1. Reproducible   *The solution produces the same results when no matter who runs it.* | This solution is designed to be self-contained with as few moving pieces as possible to simplify and stabilize the process, which in turn makes it more likely to be reproducible. |
| 1. Extendable   *The solution can easily handle new feature requests* | The functions were built with the plan in mind that the number of output tabs could change and/or the recipients could change, and therefore were not hard coded.  The process was also designed to provide the framework for what it would look like to drop in a new program when desired. |
| 1. Moveable   *The solution is not “locked into” a single location do to hardcoded file paths* | The project folder structure and code within it is set up to be easily copied and moved to the desired location due to dynamically name file paths. |
| 1. Built-in QC   *It is easy to tell if the program has run and easy to tell that order.* | Every time the process runs a html out of the data build is written out for the OA SME to be able to review and quickly assess potential failure points. |
| 1. Version control enabled   *The solution is set up to be easily ported over to a version control software solution* | The structure of the project is set up to make it easy to quickly see what items the essential code were and which were outputs and therefore not likely candidates for version control. |

Project Structure:

└── MetricsDB

├── Archive\_development

│   └── Build\_MetricDB\_Data.R

├── DataImports

│   └── Imports\_Lookups.xlsx

├── Documentation

│   └── Metrics DB Rebuild Design Doc.docx

├── Outputs

│   ├── DailyMetrics.xlsx

│   ├── DailyMetrics\_Report.html

│   ├── Archive

│    └── DailyMetrics\_*YYYYMMDD*.xlsx

│ ├── html

│    └── DailyMetrics\_*YYYYMMDD*.html

├── README.txt

├── Scripts

│   ├── BuildReport.Rmd

│   ├── BusinessDays.R

│   ├── Emails.R

│   ├── QC\_Differences.R

│   └── writeXLSX.R

├── Run\_MetricsDB\_Auto.bat

├── MetricsDB.Rproj

│   ├── Metrics\_DB\_Automation.Rout

│   └── Metrics\_DB\_Automation.R

└── Daily Metrics DB Automation Documentation.docx

Datasets involved:

|  |  |  |
| --- | --- | --- |
| **DB // Schema** | **Table** | **Notes** |
| MGTRPTP // NW | TASK\_history | This table contains the transactional history by task\_id of items processed through New Business Desktop.  ie the transactions worked (some with many steps). |
|  | TASK\_OBJECT | All transactions received (before they are routed and worked) |
|  | TASK\_VALUE | Correspods with Task\_object, translates the ID’s to human readable items like Task name, LOB, ST, policy number, etc |
| ADPP // DBO | jnl\_tbl\_Org\_Info | This table contains the associate information as stored within Workday |

Datasets:

Imports\_Lookups.xlsx

This excel file contains many tabs which are imported at the start of the process. There are used to help standardize the process in terms of what skills/task should be considered for the report, or who the notifications should go to when the process has finished.

Scripts:

source("./Scripts/BusinessDays.R")

This program pulls in the business days calendar and then converts it to determine when the last business day was. Also can be used to figure out what the date was for a few weeks back, which is then set as a variable to be used in the SQL

source("./Scripts/Emails.R")

This program contains a couple of functions to allow the user to send notifications at the end of the process. The “notification” could be an email with the output dataset attached sent to the public folders or it could be a text message sent to the point of contact to alert them that the process has finished.

source("./Scripts/writeXLSX.R")

This program contains a function to allow for multiple tabs to be dynamically written to an excel file. It will write 1 tab per name in the list.

Output:

HTML document of log written out

Text message to support team

Excel file with tabs determined necessary

QC elements

Email of log

Text message to support

What if I want to…

* Change what datasets are saved to the excel file?
  + Edit the list and names code chunk
* Add another tab to the output data?
  + Add to the list and names code chunk
* Want to change who the notification is sent to?
  + Add the day of week and the address to send to the “Email” tab as new rows.
    - *Can mix text and email*
    - *If text can use this site to figure out address: https://freecarrierlookup.com*

Overall process flow

