**Monthly Cost Allocation – CTS Challenge (PERL Version)** Brad Galyean 05/12/2016 *updated (07/17/2016)*

This is a very robust version of the CTS Coding Challenge …it performs many checks and balances, fully configurable, etc – please refer to the Python version for a much more simple version.

|  |
| --- |
| **Important:** After downloading from github, you must manually update the ROOT\_DIR in /cfg/monthly\_allocations.cfg to be your installed directory.  # Define the root/installed directory.  ROOT\_DIR|C:/CTS\_Coding\_Challenge  The scripts also compare the date timestamps of /data/employees.dat file to today’s date. You will want to manually change the date of employees.dat header to today’s date (as explained later, there are flags to turn off these checks in the config)  #HEADER|20160717  950525|WELLS|DEL|Developer|Production|402233  645819|HOWE|LAMAR|Developer|Purchasing|926260  537385|OWEN|VALENTIN|Developer|Research and Development|152341 |

First, please refer to ‘monthly\_allocation\_high\_level.pptx’ for a quick high level overview.

Also, refer to /test\_cases\_files/test\_cases.xlsx to gain some insight on the various functionality of this project (as well as the tests performed).

The monthly cost allocation script takes in a 6-digit unique employee id and returns the allocation cost of the manager’s organization to a user. This script originally pulls from a database on my localhost, however it was been configured to only read from a provided flat file for the purpose of being reviewed and executed outside my environment.

To get started testing this script, you may use the CEO’s employee ID of ‘101776’ to have the script calculate the entire company’s cost allocation.

Usage: (output from my Windows box, command-line)

C:\CTS\_Coding\_Challenge\bin>perl monthly\_allocation.pl 101776

43400.00

The log in /log will provide a readable hierarchy, in which you will find the other manager’s employee IDs easily. Or, you can look at /data/employees.dat and pick employee ids from the provided flat file.

**Files**

\bin\monthly\_allocation.pl Main Program. Usage: monthly\_allocation.pl 123456 (6-digit employee id)

\bin\data\_check.pl Check data for errors. Refer to config settings for various options.

\bin\query\_database.pl Responsible for connecting to database and writing results to /data/employees.dat

(*query\_database is disabled for purpose of CTS Challenge. Flat file is provided in /data.*)

\cfg

\cfg\accessDB.dat Database connection/login details. Used by query\_database.pl

\cfg\cost\_allocations.cfg Defines roles and costs (user defined)

\cfg\monthly\_allocations.cfg Main config settings. See config setting section below for explanations.

\data

\data\employees.dat Data to be processed by this script. See data schema section below.

\lib

\lib\ALLOCATION\_CONFIGS.pm Module used to read the config files /cfg, check for errors, and make config

settings available via functions to other scripts.

\log Logs created in YYYYMMDD.log format. Log retention available. See config settings

section.

\test\_case\_files Various files used within the test cases for ease of testing.

\test\_case\_files\5-13\_employees.dat

\test\_case\_files\5-15\_employees.dat

\test\_case\_files\6-10\_employees.dat

\test\_case\_files\create\_logs.pl

\test\_case\_files\test\_cases.xlsx

**Config settings**

monthly\_allocations.cfg

ROOT\_DIR|C:/CTS your installed root directory. **You must manually change this when installed.**

EMPLOYEE\_DATA\_FILE|/data/employees.dat option to change the data file location/name.

RUN\_SQL|N When ‘Y’, grabs data from the database and dump to EMPLOYEE\_DATA\_FILE

HEADER\_CHECK|Y Option to check the header of data file. (Y|N)

TRAILER\_CHECK|Y Option to check the trailer of the data file. . (Y|N)

DATE\_CHECK|Y Option to check that today’s date matches the date on the header. (Y|N)

RECORD\_COUNT\_CHECK|Y Option to check number of records in the file with count in trailer record. (Y|N)

MIN\_RECORD\_COUNT|50 If number of records is less than this amount, error out.

MAX\_RECORD\_COUNT|100 If number of records is more than this amount, error out.

LOG\_RETENTION|20 Max number of logs to keep in /log (retention).

PRINT\_HIERARCHY|N Option to print manager’s hierarchy to screen.

1|INFO Option to change the logging descriptors in the log.

2|WARN

3|ERROR

cost\_allocations.cfg

role|cost User defined roles/costs. \*\*All roles must be defined. Script will alert when a role is not found.

Default Values:

Developer|1000

QA Tester|500

Manager|300

**DATA Schema**

employees.dat data is structured as follows:

#HEADER|20160512 #HEADER static. Show definitive beginning of file.

20160512 reference to when the data was

pulled. In YYYYMMDD format.

950525|BOONE|GARRY|Developer|Production|402233 950525 Employee ID

BOONE Last Name

GARRY First Name

Developer Role

Production Department

402233 Manager’s Employee ID.

#TRAILER|67 #TRAILER static. Show definitive end of file.

67 Number of employee records contained in this file.