**LOCATION OF FILES**

The SAS and SQL queries referenced are in the QuarterlyTimeTracking repository. Changes to files in the repository may be made directly to the master branch.

**PHASE 1 – Get Started and Update Supporting Data**

*This phase may begin before the end of the reporting period but must be complete before Phase 3 is started.*

Background Information

You need a place to put copies of all of the files you receive and create, especially the final reports. Keeping those files in a designated, shared location makes them available to others and easy to find later if there are questions. In addition to Excel files, consider saving copies of e-mail messages as .msg files by opening the messages and selecting File > SaveAs.

The following CSYS tables contain supporting data used in Phase 4 to calculate the final cost center allocation percentages. NOCHOUSE data is used because these tables are used by other processes, are updated throughout the quarter, and reports must point to live data.

COST\_DAT\_BusinessUnitCostCenters

COST\_DAT\_CostCenters

COST\_DAT\_MailCodeCostCenters

Tasks

1. Set Up Folder
   1. Create an archive folder named after the quarter being reported (i.e. FY2021 Q1) in Q:\Support Services\Time tracking\ARCHIVE\Quarterly Time Tracking\
2. Submit Supporting Data for Review

**NOTE**: This is just to verify that the cost center information is complete: no new cost centers have been added that need to be accounted for, and that cost center names haven’t changed. If cost centers are added or the names change, the (**NOCHOUSE**) CSYS.COST\_DAT\_CostCenters table needs to be updated. The Excel time tracking spreadsheets are linked to CSYS.COST\_DAT\_CostCenters on **NOCHOUSE** so no changes are needed to the spreadsheets.

* 1. Download an excel file from SSRS Home > Quarterly Time Tracking > **Supporting Data for Review** and email it to Debbie.
  2. The report contains three tabs that are populated with the contents of the aforementioned **NOCHOUSE**.CSYS tables.
  3. Submit DCR’s as necessary to update NOCHOUSE based on Debbie’s feedback.

1. Update Supporting Data
   1. Receive updated data from Debbie for the information you asked her to review.
   2. Submit DCRs to update the applicable tables on NOCHOUSE according to the following instructions. Unless otherwise noted, there are no SAS jobs or SQL scripts to do this so you will need to create your own.
      1. COST\_DAT\_BusinessUnitCostCenters
         1. This is a historical table so the existing rows should not be updated or deleted. Instead, enter an EffectiveEnd date to inactivate a row and insert a new row with updated information.
         2. Enter the last day of the quarter previous to the quarter being reported in the EffectiveEnd field.
         3. Insert a new row with the current information.
            1. Enter the first day of the quarter being reported in the EffectiveBegin field.
            2. Enter NULL in the EffectiveEnd field.
            3. Verify the sum of values in the Weight field for each business unit is 100 (the business unit is 100% allocated to cost centers.
         4. INSERT new BU-CostCenter,sql can be used to add rows to this table.

The

* + 1. COST\_DAT\_CostCenters
       1. DO NOT delete CostCenterId 13. This id is hardcoded in the SAS code and must always represent the Cornerstone servicing cost center.
       2. Other cost center records may be updated or deleted and rows may be inserted for new cost centers as long as FK constraints to (**NOCHOUSE**) COST\_DAT\_BusinessUnitCostCenters and (**NOCHOUSE**) COST\_DAT\_MailCodeCostCenters are not violated.
    2. COST\_DAT\_MailCodeCostCenters
       1. This is a historical table so the existing rows should not be updated or deleted. Instead, enter an EffectiveEnd date to inactivate a row and insert a new row with updated information.
       2. Enter the last day of the quarter previous to the quarter being reported in the EffectiveEnd field.
       3. Insert a new row with the current information.
          1. Enter the first day of the quarter being reported in the EffectiveBegin field.
          2. Enter NULL in the EffectiveEnd field.
          3. Verify the sum of values in the Weight field for each mail code is 100 (the mail code is 100% allocated to cost centers.
       4. Although there is no FK relationship to enforce it, the MailCode field must be populated with values from the Code field of the (**NOCHOUSE**) BSYS..GENER\_LST\_UHEAACostCenters table.

1. Create Time-tracking Spreadsheet Template

**NOTE:** This is for the next quarter (the quarter after the quarter being reported), not the quarter being reported.

* + 1. Save his is for the next quarter, not the quarter being reported.efore proceeding in case cost center allocations have changed whica copy of the template for the quarter being reported (found in Q:\Support Services\Time tracking\) in Q:\Support Services\Time tracking\.
    2. Change the file name to reflect the quarter the template is for.
    3. Update date range validation criteria
       1. Select a cell in the Date column.
       2. Select the Data ribbon.
       3. Click the down arrow on the Data Validation icon on the Data Tools menu and select Data Validation.
       4. Change the start and end dates to the start and end dates of the quarter the template is for.
       5. Check “Apply these changes to all other cells with the same settings”.
       6. Click OK.
    4. The Excel time tracking spreadsheets are linked to CSYS.COST\_DAT\_CostCenters on **NOCHOUSE** so no changes are needed to the list of cost centers on Sheet2.
    5. Attach the template to an email and send the email to everyone with instructions to use new template for the following quarter.

1. Update Batch Script Time Tracking Average Spreadsheet

**NOTE:** The DCRs submitted in Phase 1 step **3** above to update supporting information must be complete before proceeding in case cost center allocations have changed which would affect how the time spent running batch scripts is allocated.

**NOTE:** This step addresses two types of errors in the Batch Script Time Tracking Average spreadsheet data: Invalid script names and Scripts allocated to overhead.

* 1. Generate Batch Script Time Tracking Average Error Reports
     1. Get updated Batch Script Time Tracking Average spreadsheet from Debbie.
     2. Save the spreadsheet as ***T:\SAS\Batch Script Time Tracking Average.xlsx***
     3. Run the **02 Batch Script Errors.sas** query.
  2. Clean up invalid script names

**NOTE:** The script names in the “Sacker Name” column of the Batch Script Time Tracking Average spreadsheet must match the name of the script in Sacker exactly or SAS can’t find the script to determine the business unit(s) associated with the script and use the business unit(s) to determine the cost center to which time spent running the script should be allocated. This step is to identify script names which are not in Sacker and update them in the spreadsheet.

* + 1. If the ***T:\SAS\Batch Script Name Errors.xlsx*** file was not created, there were no script name errors and you can skip to step **c** below.
    2. Get the correct script names from Sacker or Systems Support for the scripts on the ***Batch Script Name Errors.xlsx*** report (T:\SAS\Batch Script Name Errors.xlsx).
    3. Update the “Sacker Name” in the Batch Script Time Tracking Average spreadsheet
    4. See last page on this doc for a list of common errors.
  1. Once the **02 Batch Script Errors.sas** query returns no errors, create a new copy of the spreadsheet for System Support to use for the next quarter (the quarter after the quarter being reported).
     1. Delete the data entered in the green data entry cells.
     2. Send the spreadsheet to Systems Support.
  2. Review scripts allocated to business units.

**NOTE:** Some scripts may be assigned to business units that are not assigned to a cost center in the CSYS.COST\_DAT\_BusinessUnitCostCenters table which causes time spent running the script to be allocated to overhead. Scripts allocated to overhead need to be assigned to business units assigned to a cost center in the CSYS.COST\_DAT\_BusinessUnitCostCenters table (this has to happen now, instead of during review of final numbers as this may affect final numbers).

**NOTE:** The “ARC Add Database” batch script will appear on the Batch Scripts Allocated to Overhead report. Ignore it for now as it is divided up among three business units, with the Systems Support BU generating the error. See “ARC Add Database in Batch Scripts Allocated to Overhead” email in FY20 Q3 archive.

* + 1. If the ***T:\SAS\Batch Scripts Allocated to Overhead.xlsx*** file was not created, no time spent running batch scripts was allocated to overhead and you can skip to Phase 1 step **[6](#CalcBatchScriptAllocation)** below.
    2. Send the ***Batch Scripts Allocated to Overhead.xlsx*** report (T:\SAS\Batch Scripts Allocated to Overhead.xlsx) to Debbie with instructions to change the business unit assignments of the scripts in Sacker to units which are allocated to a cost center in the CSYS.COST\_DAT\_BusinessUnitCostCenters table.
    3. Wait until Debbie indicates the business unit assignments are up to date before proceeding.

1. Calculate batch script cost center allocation.
   1. Make sure the steps above have been followed to update the ***T:\SAS\Batch Script Time Tracking Average.xlsx*** spreadsheet before proceeding.
   2. Open the **03 Calculate Batch Script Cost Center Weights.sas** query.
   3. Update the EndDate variable to the last day of the quarter being reported.
   4. Run the query.
   5. Open the Checksum data set in the Work library.
   6. Correct any data errors if the check sum is not 100 (rounding errors are acceptable).
   7. Send the ***BatchScriptCostCenters.xlsx*** report to Debbie to review and approve the cost center allocation.
   8. Receive approval from Debbie.
2. Update the COST\_DAT\_BatchScriptWeights table in OPSDEV
   1. Open the **04 Update Batch Script Weights.sql** query.
   2. Step 1:
      1. Update the @BEGIN\_DATE variable to the first day of the quarter being reported.
      2. Update the @EFEND\_DATE variable (aka EffectiveEndDate) to the last day of the quarter previous to the quarter being reported (this date will be used to update existing records in the table where the EffectiveEnd date is NULL, the EffectiveEnd date of newly inserted rows for the quarter being reported will be left NULL).
      3. @INSERTS = the number of new rows to be inserted (the number of rows in the ***BatchScriptCostCenters.xlsx***file*)*
   3. Step 2:
      1. Scroll down in the code to step 2 (INSERT new data)
      2. Copy the BatchScriptWeights data from the Excel document that Debbie approved in the previous step (***BatchScriptCostCenters.xlsx***), paste the data into the VALUES list, and format it so that SQL Server can insert it.
   4. Save, commit, and push the changes.
   5. Run the query on **OPSDEV** (no other processes use this data so for ease of use the actual allocation calculation will use the **OPSDEV** table).
3. Add New Agents
   1. Open the **07 Time Tracking Pre-load Error Reporting.sas** and **08 Load Time Tracking Data.sas** queries.
   2. Add PROCIT macro call statements for new agents who are required to fill out a time tracking spreadsheet but who don’t already have a macro call statement.
      1. Check with Debbie or Melanie if you aren’t sure if there are new agents who should be filling out a time tracking spreadsheet.
      2. Search for “%PROCIT”.
      3. You should see a list of %PROCIT statements, one for each agent required to fill out a time tracking spreadsheet.
      4. The arguments are the name of the agent’s folder in “Q:\Support Services\Time tracking”, the agent’s name in the file name of their time tracking spreadsheet (e.g. “Jay Davis” in “FY 2021 Q1 Time Tracking - Jay Davis.xlsx”), and the agent’s e-mail address.
      5. Add the new statements to both queries.
   3. Save, commit, and push the changes.
4. Update Copies of Access Databases

NOTE: The SAS **07 Time Tracking Pre-load Error Reporting.sas**, **08 Load Time Tracking Data.sas**, and **09 Cost Center Allocation.sas** jobs use ODBC connections to access data from the Projects and Procedures Management Database (PMD) MS Access databases. SAS may put these databases in an inconsistent state so it is a good practice to connect to copies of the databases instead of the production databases themselves.

* 1. Copy the following files to X:\PADR\Budget Reporting Copies:
     1. X:\PADR\Procedures\Procedures.mdb
     2. X:\PADR\Projects\Projects.mdb

**PHASE 2 – Need Help Time Tracking Clean-Up**

Background Information

Users sometimes forget to click Stop when they are done working on Need Help tickets. An SSRS report sends daily notifications to those users who forgot to click Stop. However, some users tend to ignore these notifications, and sometimes users might not be available to log in to the system when the final time-tracking report is due. As a result, the time tracking records for some tickets indicate more time was spent than was actually spent or there is not end time for the records. Records with either condition need to be identified and corrected.

*Clean-up must occur after the end of the reporting period but may begin before phase 1 is complete.*

Tasks

1. After the end of the reporting period, run the following SSRS Quarterly Time Tracking reports to get lists of tickets where there is no end time (still running) or it appears the end time is incorrect (long running - the time elapsed is more than eight hours).
   1. Need Help Time Tracking – Long Running
   2. Need Help Time Tracking – Still Running
   3. Need Help Time Tracking - Overlapping
2. Send the results to individuals on the report and instruct them to update their time tracking where needed and then report back to you.
3. Follow up if you don’t get a response.
4. Verify there are no long running or still running Need Help time tracking records before proceeding.

**PHASE 3 – Load Excel Time Tracking Data**

Background Information

Data in individual time tracking spreadsheets needs to be reviewed, errors need to be corrected, and then the data needs to be loaded into the **OPSDEV**.CSYS.COST\_DAT\_TimeTracking table.

*Phase 3 must begin after the end of the reporting period to ensure tracking spreadsheets are complete for the quarter but it can be completed before phase 2 is complete.*

Tasks—Two SSRS tools were created to help staff with time tracking. See Home > Support Services > KPI > Need Help Daily Hours / Need Help Total Hours. An SSIS package was also developed for daily error checks and scheduled on the JAMS computer to run every weekday morning. There is also a SAS script (TimesheetHoursAggregator.sas) that compiles all daily NH and Excel time together. See Github > Quarterly Time Tracking > TimesheetProcessing folder.

1. A week before the end of the quarter, send out a general email to staff asking if they are going to be out of the office (vacation, etc) and process their timesheet before they leave. Process timesheet manually with SAS.
2. Manually review spreadsheets to ensure the following:
   1. All agents have saved their spreadsheet in their folder in Q:\Support Services\Time tracking\.
   2. The agent names in the spreadsheet names correspond to the agent names in the **07 Time Tracking Pre-load Error Reporting.sas** and **08 Load Time Tracking Data.sas** queries
      1. Change the spreadsheet names and/or add/remove/update %PROCIT statements towards the bottom of the queries as needed).
      2. See phase 1 step **8** above for more information about adding %PROCIT statements.
      3. The spreadsheet names follow the “[Quarter] Time Tracking – [Agent Name].xlsx” pattern.
   3. The spreadsheets include the primer data in rows 2 through 9 and have the correct number and order of columns.
3. Create and distribute the error reports.
   1. Open the **07 Time Tracking Pre-load Error Reporting.sas** query.
   2. Change the QTR variable in the %LET statement at the top to reflect the quarter being reported.Save, commit, and push the changes.
   3. Run the query.
      1. The query was designed with the intent that the user would be able to uncomment all of the PROCIT macro call statements and run them all at once. However, that doesn’t always work so you may need to uncomment and run one PROCIT macro call statement at a time.
      2. The SAS script may prompt you to allow or deny an e-mail message to be sent to automatically distribute error reports to agents.
      3. Click Allow after the green timer bar runs out or click Deny depending on whether you want SAS to send the messages.
      4. If you allow SAS to send the message and it is successful, the message should be in your Sent Items mailbox in Outlook.
      5. The agent error reports are in T:\SAS\.
      6. If the agent has errors, the file name will be “Time Tracking Errors – [Agent Name].html”.
   4. If the agent’s time tracking spreadsheet was processed and the agent has no errors there will still be a report but the file name will be “No Time Tracking Errors – [Agent Name].html”.Review the ***Time Tracking Errors - Time Tracking Not Processed.html*** report (T:\SAS\Time Tracking Errors - Time Tracking Not Processed.html) and the SAS log to identify any processing errors. In addition to error messages, “>>>ERROR” will be written to the SAS log indicating an error condition was handled and written to the report.
   5. Resolve processing errors and run the query again.
      1. The most common processing error is the time tracking spreadsheet is open by the agent. If they are slow to close it, you can make a copy and modify the SAS temporarily to process the copy.
   6. Rerun the query and distribute the error reports or open the agent’s time tracking spreadsheet and fix the errors yourself until all errors are resolved.
4. Load the Data (Trial Run). *Phase 3 up to this point must be complete before proceeding.*
   1. Create a copy of the **OPSDEV**.CSYS.COST\_DAT\_TimeTracking table as you may need to restore the existing data if something goes wrong. (SELECT \* INTO CSYS..COST\_DAT\_TimeTracking[quarter name(FYXXXQX)] FROM CSYS..COST\_DAT\_TimeTracking
   2. Delete the data from **OPSDEV**.CSYS.COST\_DAT\_TimeTracking\_Trial\_Run.
   3. Open the **08 Load Time Tracking Data.sas** query.
   4. Change the QTR variable in the %LET statement at the top to reflect the quarter being reported.
   5. Verify that “%LET TBL=COST\_DAT\_TimeTracking\_Trial\_Run;” is uncommented and that “\*%LET TBL=COST\_DAT\_TimeTracking;” is commented out.
   6. Save, commit, and push the changes.
   7. Run the query.
      1. The query was designed with the intent that the user would be able to uncomment all of the PROCIT macro call statements and run them all at once. However, that doesn’t always work so you may need to uncomment and run one PROCIT macro call statement at a time.
   8. Complete the following steps until the job runs without any errors.
      1. Examine the log and the data in **OPSDEV**.CSYS.COST\_DAT\_TimeTracking\_Trial\_Run and identify any errors.
      2. Resolve the errors.
      3. If running all of the PROCIT macro call statements, delete the data from **OPSDEV**.CSYS.COST\_DAT\_TimeTracking\_Trial\_Run. Otherwise, only delete the data for the agent that may not be correct.
      4. Rerun the query.
5. Load the Data (Final Run)
   1. If you are confident the time tracking spreadsheets loaded correctly and **OPSDEV**.CSYS.COST\_DAT\_TimeTracking\_Trial\_Run has a clean load of the spreadsheets, you can use **08 Copy Trial Run Data.sql** to insert the data from COST\_DAT\_TimeTracking\_Trial\_Run into COST\_DAT\_TimeTracking. Otherwise, follow the steps below to rerun the **08 Load Time Tracking Data.sas** query for the final load into COST\_DAT\_TimeTracking.
   2. Comment out “%LET TBL=COST\_DAT\_TimeTracking\_Trial\_Run;” and uncomment “\*%LET TBL=COST\_DAT\_TimeTracking;”.
   3. Save, commit, and push the changes.
   4. Rerun the query.
   5. Verify there were no errors.
   6. Account for agents who left at some point during the quarter. Instructions are in code comments.

**PHASE 4 – Preview Run**

Background Information

The **09 Cost Center Allocation.sas** query creates an ***Allocations Detail.xlsx*** file which lists all of the Excel time tracking and NeedHelp tasks and the cost center(s) to which they are allocated. Debbie goes through the tab of tasks allocated to overhead and identifies tasks which can be allocated to a cost center. While the final report cannot be created until the agents weights are received and updated in Phase 5, the agent weights are not needed for Debbie to perform her review as she is only looking at the cost centers (which isn’t affected by the agent weights) not the allocation percent. Since the agent weights are not received from HR until right before the deadline for the final report, it is a good idea to give Debbie something to review as soon as possible.

*Phase 4 may not begin until after phase 3 is complete.*

Tasks

1. *Phase 3 must be complete before proceeding.*
2. Follow the steps in Phase 6 to create a preview version of the final report (the only difference between the preview version being created and the final version you will create in Phase 6 is the preview report will not have the correct allocation percentages because Phase 5 is not yet complete.
3. Receive Debbie’s list of tasks to be reallocated.
   1. Save a copy of Debbie’s message in the ARCHIVE folder you created in Phase 1 step **1**.
   2. **EITHER** update the data in **OPSDEV**.CSYS.COST\_DAT\_TimeTracking table directly to reallocate the tasks as Debbie requested **OR** complete the steps under line 4 below to modify the underlying data and reload the time tracking spreadsheets.
   3. Save copies of the scripts you write to update the data in **OPSDEV**.CSYS.COST\_DAT\_TimeTracking in the ARCHIVE folder you created in Phase 1 step **1**.
4. Reallocate Excel Time Tracking tasks.
   1. Have agents update their Excel time tracking spreadsheets (or do it yourself) so tasks will be allocated as Debbie has indicated. This usually involves selecting a cost center for Generic Meeting tasks.
   2. For SR, SASR, LTS, and PMD tasks,
      1. Open the task in the respective database.
      2. Access the detail screen for the asset.
      3. Write down the request and the business unit that is causing the task to be misallocated so you can change the business unit back later during Phase 7.
      4. Temporarily change the business unit on the detail screen to one that is mapped to the desired cost center in **NOCHOUSE**.CSYS.COST\_DAT\_BusinessUnitCostCenters.
   3. For Project tasks,
      1. Open the project in the Project tracking database.
      2. It is likely no business units are listed.
      3. Find out from the project manager which units should be listed but make sure they are mapped to the desired cost center in **NOCHOUSE**.CSYS.COST\_DAT\_BusinessUnitCostCenters.
      4. Update the business units.
   4. Restore the **OPSDEV**.CSYS.COST\_DAT\_TimeTracking table from the backup or copy made during Phase 3.
   5. Follow the steps in Phase 3 to reload the corrected data.
5. Reallocate NeedHelp tasks.
   1. Insert new rows into the **OPSDEV**.CSYS.COST\_DAT\_NhManualAllocation table for NeedHelp tasks to be reallocated. OPSDEV is used for ease of use and because no other processes use the data.
      1. **08 Insert NeedHelp Tickets for Manual Allocation.sql** may be used for this.
      2. TicketID = the NeedHelp ticket number.
      3. CostCenterDeterminant = a short description of the manual allocation such as “LPP CS manual allocation to Finance”.
      4. CostCenterId = the **NOCHOUSE** ID of the cost center to which the tasks for the request should be allocated.
      5. CostCenterWeight = the integer percent of the tasks for the request to be allocated to the cost center (i.e. 100 if the tasks are to be allocated 100% to the cost center) Make sure the sum of the percentages for each request equals 100)
      6. EffectiveBegin = the first day of the quarter being reported
      7. EffectiveEnd = leave this NULL unless the reallocation is only for a specific quarter in which case enter the last day of the quarter being reported
6. Follow the steps in Phase 6 to create a preview version of the final report for verification of the manual allocations.
   1. Verify the tasks were allocated as desired.
   2. Correct any errors and run the preview report again until the tasks were allocated as desired.
7. Debbie does not need to see the preview report again.
8. Wait to receive the updated agent weights spreadsheet from Debbie before proceeding.

**PHASE 5 – Update OPSDEV.CSYS.COST\_DAT\_AgentWeights**

Background Information

Data in this table is used to weight agents’ time according to how much they are paid. Debbie is very good about requesting this information from HR without being reminded but she doesn’t get it until right before the deadline. It is the last piece of information needed for the final phase (calculating the cost center allocation percentages). COST\_DAT\_AgentWeights is a historical table that exists only in OPSDEV. Enter an EffectiveEnd date to inactivate any NULL rows and insert a new row with updated information.

Tasks

1. Receive the updated agent weights spreadsheet from Debbie.
2. **04 Update Agent Weights.sql**
   1. Step 1: UPDATE effective end date
      1. To inactivate the active rows, enter the last day of the quarter previous to the quarter being reported in @END\_DATE (ie. the effective end date).
   2. Step 2: INSERT new data
      1. Enter new agent rows and update the existing rows with the updated information from HR. You may round the weight from HR to 6 decimal points (you may have to expand the width of the weight column and adjust the number of decimal points displayed to see the exact number provided by HR) and it doesn’t matter if the sum of the weights being inserted is not exactly 1 as long as it is greater than .999995. You will have to add the SqlUserId from production (NOCHOUSE) to the information provided by HR for new agents. Enter the first day of the quarter being reported as the effective begin date. Leave the effective end date NULL.
      2. Put the number of inserts in @INSERTS.

**PHASE 6 – Calculate Cost Center Allocation Percentages**

Background Information

Now that all of the necessary data has been gathered, you are ready to calculate the percent of Support Services costs to allocate to each cost center.

Tasks

1. Before you proceed,
   1. Make sure there are no Need Help tickets where there is no end time and the start time is during the reporting period. You can use the **05 Need Help Time Tracking.sql** query or SSRS queries for this but don’t worry about the long running tickets, just those that are still running.
   2. Delete the following files from T:\SAS\
      1. Allocations Detail.xlsx
      2. Allocations Validation.xlsx
   3. Allocations.xls
2. Open the **09 Cost Center Allocation.sas** query.
3. Update the PeriodBegin and PeriodEnd variables to the first day of the quarter being reported and the last day of the quarter being reported respectively.
4. Run the **09 Cost Center Allocation.sas** query.
5. Check the ***Allocations Validation.xlsx*** report and correct any errors in the data or the query.
   1. The Allocations Validation spreadsheet contains items that weren’t included in the final calculations. These need to be resolved.
   2. If the numbers on the first tab aren’t equal, check the other tabs to see which records are not making it from the source (Reporting.TimeTracking for Need Help or CSYS.COST\_DAT\_TimeTracking for Excel) to the final results.
   3. The problem is usually there is a join that is excluding the record because either the data is wrong or the query isn’t accounting for a new situation.
   4. If necessary, update Sacker (SAS & Scripts) and/or the Letter Tracking Database with the correct Business Unit (one that is mapped in BusinessUnitCostCenters) from the drop-down menu in Job Details.
6. Send the results to Debbie for review.

**PHASE 7 – Final Step: Wrap it Up**

Background Information

Do this as soon as you hear back from Debbie and she says everything looks good.

Tasks

1. Update the business units changed in Phase 4 to their original values.
2. Commit and push changes made to files in the QuarterlyTimeTracking repository.
3. Create and submit a Uheaa Need Help ticket to have the DBA transfer the new time tracking data in CSYS.COST\_DAT\_TimeTracking on OPSDEV to CSYS.COST\_DAT\_TimeTracking on NOCHOUSE where it is used by managers and others for various purposes.
   1. Select “Data Change Request” under Create New.
   2. Select “Budget Reporting” as the DCR Category.
   3. The DBA can transfer this data w/o a specific script. See UNH 58578 for example)
   4. Include instructions to only copy records with a task date on or after the begin date of the quarter being reported such as “All time tracking data for last quarter (TaskDate is on or after [first day of quarter being reported]) in the [CSYS].[dbo].[COST\_DAT\_TimeTracking] table on OPSDEV needs to be transferred to the corresponding [CSYS].[dbo].[COST\_DAT\_TimeTracking] table on OCHOUSE.”.
4. Transfer all approved documents to the **Q:\Support Services\Time tracking\ARCHIVE\Quarterly Time Tracking** folder created in Phase 1**.**
5. Move the time tracking spreadsheet template for the quarter being reported from Q:\Support Services\Time tracking\ to Q:\Support Services\Time tracking\ARCHIVE\.

**NOTES:**

**All original queries are saved in the SVN\SAS\WorkInProgress\Jay\\_Budget Reporting\ folder.**

**All current queries are saved on Github > UHEAA > Quarterly-Time-Tracking**

**Previous quarter reports are saved at Q:\Support Services\Time tracking\ARCHIVE\Quarterly Time Tracking**

**Batch Script Name Errors**

|  |  |  |
| --- | --- | --- |
| Script ID | Script Name | Sacker Name |
| ACHREVFED | ACH Review- FED | ACH Review - FED |
| DFACDFED | DFACDFED | Defer & Forb ACD Letters - FED |
| DTX7L | DTX7L Error Letter Records | DTX7L Errored Letter Records |
| FINALBFED | Final Bill-FED | Final Bill Letter - FED |
| PRETNFRNOT | Pre-Transfer Notifications | Pre-Transfer Notifications - FED |
| SCRAINTUPD | SCRA | SCRA Interest Updates from DOD |
| SCRAIUDFED | SCRA-FED | SCRA Interest Updates from DOD - FED |
| SOALTRFED | SOALTRFED | SOA Letter - FED |