CompleteDailyInventory

**Overview:**

This macro is to be run inside of Outlook. The macro relies on an Outlook rule to move specific emails from Supplier Reporting into a specific folder. When run, the program loops through this folder and downloads each report and saves them to a SharePoint folder. It also calls a python script to get the Lindner report, and finally it calls the DailyInventoryTable macro in excel to create a new report.

**Subs/Functions:**

completeDailyInventory

This sub is the main driver for the program. It starts by checking to see if a report has already been ran, and then will ask the user if they would like to rerun the report. Next, the program will delete the old reports in the SharePoint folder. Then, the program will download all of the new reports and save to the SharePoint folder. After that, the program will call the python script to run to get the Lindner inventory. Finally, the program will create the report by calling the CreateTable sub and move any old emails to an old reports folder.

Currently this is set-up to read and write everything to a SharePoint folder, however, this could all be run locally by changing the paths throughout both the excel and outlook macro. This could be made simpler with a config file that contains the path of the working directory for the report to be run, and then having each macro read the config file.

ReportRun

This function loops through all the files in the sharepoint folder and if it finds a file that contains the month and day that matches with today’s date, it will return true and the user will then be prompted if they would like to re-run the report. Otherwise, this function will return false and the program will proceed.

DeleteReports

The old reports must be deleted before creating a new report as the excel macro does not have a way of determining which reports are most recent. This sub simply loops through the SharePoint folder and deletes every file except for the “ProductInformation.xlsm” excel workbook and the general log file.

The ProductInformation file is necessary for getting ax numbers, prod8 codes, descriptions, and ship-by dates when running the excel macro. If this file gets deleted, there is a backup in the SharePoint at Projects\InventoryReports\Macro Utility\Misc. The general log is the log for the currently running macro.

moveLindner

The Lindner script saves the report it generates into the directory it was called in. This sub is used to move it to the SharePoint folder. If given more time, this sub could be removed by calling the Lindner report directly from the SharePoint folder (have not tested and there could be problems using a cloud folder). There would need to be an exception made for Scrape.exe in the DeleteReports sub.

DownloadReports

This function is responsible for downloading and extracting all of the reports from outlook to the SharePoint folder. The sub loops through each item inside of the report folder, and if it is an email it extracts the subject, sender, and the date received. Since New Holland sends their report as text inside of the email body, it must be processed differently. To start, the date of the email is checked, to make sure it is from the correct day. Then, for New Holland another sub is called to extract the text and create an excel report. For all of the other reports, once the date of the email is verified, the program gets the attachments and makes sure that the attachment is an excel file. If the email is from city, the file name must be altered as each brewery sends a report with the same name. This is accomplished by simply adding a 1, 2, or 3.

goodEmail

This function takes in the date received of the email and if is from New Holland. New Holland sends their report in the evening so the correct date will be one day back, whereas the other reports are sent in the morning so today’s date is checked.

exportToExcel

This sub takes in the New Holland email and the path to the SharePoint folder. It starts by creating a new excel workbook. It then creates an array for the rows and columns of the table inside of the New Holland email. Each row is put into the array by splitting them on the carriage return (vbcrlf). The program then loops through each of these rows and splits each cell with a tab. Each cell is then output to the excel file. There is usually blank space at the end of the new Holland email, so if there are empty rows, the program will break out of the for loop. Finally, the excel file is saved to the SharePoint folder.

CreateTable

This function starts by creating a new excel workbook and opening the PERSONAL excel workbook which contains the excel macro. It then gets today’s date and calls the macro stored in the PERSONAL workbook. The line “xlWb.Application.Run "PERSONAL.XLSB!DailyInventory"” may need to be altered on different machines. If you go to Excel, Developer tab, and then click on Macros, you can find the name that the macro is stored as in the PERSONAL workbook. The file is then saved with today’s date in the filename and the function will return true. If an error occurs, it will be logged and the function will return false.

moveOld

This function loops through the emails in the reports folder, and if they are not from today will move them to an old folder so they can be referenced if needed.

Application\_Startup

A sub with this name placed into the “ThisOutlookSession” Object will always be called when Outlook opens. As of right now, the report is run after the user opens outlook.

Application\_Reminder

Checks what reminder is called and exits if it is anything but the “Run Inventory” reminder. The main driver for the report is then called.

olRemind\_BeforeReminderShow

This sub will automatically dismiss the reminder created so the user does not have it pop up every day.

CreateAppointment

Creates an appointment that will trigger the report to be run. The line tDate = Now () + 2 / 1440 controls how long the report takes to run after opening Outlook. The 2 signifies two minutes. The buffer is given so that Outlook can sync with the server.