**Fast POFS Readme**

**Release: 10/19/2023**

**Auth: David Kao**

**Summary:**

This VBA program automates data processing tasks within an Excel workbook named "Fast Component DO\_NOT\_TOUCH.xlsm." and “Fast Shippable DO\_NOT\_TOUCH.xlsm” It updates, calculates, and saves the workbook, then performs calculations and data analysis on a summary worksheet. The program is scheduled to run daily at 1:00 AM using Windows Task Scheduler, streamlining data management and analysis.

**Windows Task Scheduler Automation for VBA Program**

**Step 1: Set Up Windows Task Scheduler** Before using this VBA program, it's important to configure Windows Task Scheduler to run the program daily at 1:00 AM. Follow these steps:

1. **Open Windows Task Scheduler:** You can typically find Task Scheduler in the Windows Control Panel or by searching for it in the Start menu.
2. **Create a New Task:** In Task Scheduler, click on "Create Basic Task" or "Create Task" to create a new task.
3. **Name and Describe the Task:** Give your task a name and an optional description. For example, you can name it "Daily VBA Automation" and describe it as "Runs the VBA program in Fast Component.xlsm every day at 1:00 AM."
4. **Trigger Configuration:** Set the trigger to run the task "Daily" and specify the time as "1:00 AM." Choose the start date and, if needed, set an end date.
5. **Action Configuration:** Select "Start a program" as the action. Browse and select the Excel executable (Excel.exe) as the program/script, and provide the full path to the Excel workbook containing your VBA code (Fast Component.xlsm) as the argument.
6. **Finish Configuration:** Review your settings, and when satisfied, click "Finish" or "OK" to create the scheduled task.

Now, your VBA program will be automatically executed every day at 1:00 AM using Windows Task Scheduler.

**MainProgram Subroutine** The MainProgram subroutine is the main entry point for your VBA program. It consists of several steps that are executed sequentially.

1. **UpdateAndCalculateWorkbook:** This step calls the UpdateAndCalculateWorkbook subroutine, which updates and calculates the content of the current workbook (Fast Component.xlsm). It also duplicates this workbook with a new filename based on the current date and saves it in a specified directory.
2. **savetovalues:** This step calls the savetovalues subroutine, which converts formulas to their calculated values in all worksheets of the current workbook. This can be useful to freeze the current state of the data.
3. **UpdateSummary:** This step calls the UpdateSummary subroutine, which updates the "Component Parts Summary" worksheet in the current workbook with specific calculations and data from other worksheets. It calculates various values related to parts, such as open sales, open purchases, and future projections, and updates the summary accordingly.
4. **closefile:** This step calls the closefile subroutine, which closes the current workbook (Fast Component.xlsm) while saving any changes made during the program's execution.

**UpdateAndCalculateWorkbook Subroutine** The UpdateAndCalculateWorkbook subroutine performs the following tasks:

1. **Set File Paths:** It defines variables for the source file path (SourceFilePath) and the destination file path (DestinationFilePath). The source file is "ORDER REPORT FY2016.xlsx," and the destination file will be a duplicate of the current workbook with a new name.
2. **Attempt to Open Source File:** It tries to open the source file in read-only mode. If successful, it assigns it to the OrderReport workbook object.
3. **Update Workbook:** It calculates all sheets in the current workbook (Fast Component.xlsm) to update and recalculate the data.
4. **Disable Automatic Calculation:** It disables automatic worksheet calculation to optimize performance during data processing.
5. **Save as a New File:** It saves the current workbook as a new file with a filename containing the current date in "yyyy-mm-dd" format. The new file is saved in a specified directory.
6. **Close Workbooks:** It closes both the current workbook (with changes saved) and the read-only "ORDER REPORT FY2016" workbook (without saving changes) if it was successfully opened.

**savetovalues Subroutine** The savetovalues subroutine converts all formulas in every worksheet of the current workbook to their calculated values. This effectively replaces formulas with the results they produce. After execution, formulas no longer automatically update when input data changes, and the workbook contains static data.

**UpdateSummary Subroutine** The UpdateSummary subroutine updates the "Component Parts Summary" worksheet in the current workbook. It performs the following tasks:

1. **Set Summary Sheet:** It defines the summarySheet object, representing the "Component Parts Summary" worksheet.
2. **Insert New Column:** If the first column in the summary sheet doesn't contain "SS index," it inserts a new column and adds "SS index" as the header.
3. **Set Column Headers:** It sets headers for various columns in the summary sheet, such as "CURRENT FREE STOCK," "OPEN SALES," and "OPEN PURCHASE."
4. **Hide Columns:** It hides unnecessary columns in the summary sheet to improve readability.
5. **Collect Part Data:** It collects data for each part from other worksheets in the workbook, including open sales, open purchases, and projections for future quarters.
6. **Calculate SS Index:** It calculates the SS (Safety Stock) index for each part based on the collected data.
7. **Calculate Inhand Index:** It calculates the Inhand Index for each part based on specific criteria and applies color coding to highlight values.
8. **Apply Color Gradient:** It applies a color gradient to the "SS index" column based on specified criteria to visually represent safety stock levels.

**closefile Subroutine** The closefile subroutine simply closes the current workbook (Fast Component.xlsm) while saving any changes made during program execution.

This VBA code automates various tasks related to updating and analyzing data in the workbook, making it more efficient and user-friendly for your specific needs. Users can run the MainProgram subroutine to execute these tasks in sequence.

\*\* FastShippable shares the same structure without **UpdateSummary** but with minor adjustments.

A person sitting in an office

Description automatically generated

-David 10/19/2023