I'm not a bot

To split Excel sheets into multiple files, there are two methods. The manual approach involves copying and pasting each sheet individually, while the VBA macro method uses code to automate the process. Method 1: Manual Approach Method 1.1: Copy and Paste Options Select the entire sheet, copy it (Ctrl + C), open a new worksheet, paste it (Ctrl + V), and save the workbook. Repeat this process for each sheet. This method is time-consuming and inefficient. Method 1.2: Move or Copy Feature Right-click on a sheet, select "Move or Copy," choose "Create a Copy" and select a destination folder. This method is slightly faster than copy-pasting. Method 2: VBA Macro Before using the code, create a folder to store output files and save the main Excel file in that folder. Method 2.1: Splitting into Multiple Excel Files Save the main workbook as "Split Excel Sheet.xlsx" in the new folder, then open the original workbook, go to Developer tab, select Visual Basic, insert a module, copy code, and run it. This splits all sheets into individual Excel files. Method 2.2: Splitting into Multiple PDF Files Copy the following code, insert a module, paste the code, and run it. This splits all sheets into multiple PDF files. Sub Split_Sheet_Specific_Word() Dim File_Path As String Dim Find_Term As String Find_Term = "22" File_Path = Application.ActiveWorkbook.Path Application.ScreenUpdating = False Application.DisplayAlerts = False For Each Sheet In ThisWorkbook.Sheets If InStr(1, Sheet.Name, Find_Term, vbBinaryCompare) 0 Then Sheet.Copy Application.ActiveWorkbook.SaveAs Filename:=File_Path & "\" & Sheet.Name & ".xlsx" Application.ActiveWorkbook.Close False End If Next Application.DisplayAlerts = True Application.ScreenUpdating = True End Sub Alternatively, you can download the practice workbook from here and use this code with some modifications to include a folder chooser and CSV delimiter option. Function Get_Folder_Name(Msg As String) As String Dim bInfo As BROWSEINFO, path As String, r As Long Dim X As Long, pos As Integer bInfo.pidlRoot = 0& If IsMissing(Msg) Then bInfo.lpszTitle = "Select a folder." Else bInfo.lpszTitle = Msg End If bInfo.ulFlags = &H1 X = SHBrowseForFolder(bInfo) Parse_the_result_to_get_folder_name(X) End Function These codes allow you to split Excel sheets containing specific words or phrases into multiple files. They also include a folder chooser and CSV delimiter option for more flexibility. Looking forward to meeting everyone at the upcoming conference and discussing our strategies. Given text has been rewritten using "ADD SPELLING ERRORS (SE)" method. Looking forward to the meeting tomorrow, where we'll discuss our strategy for splitting each Excel sheet into a separate file and saving them separately is an exciting project. If you have an Excel workbook with many worksheets, there isn't a straightforward way to split each of these sheets into separate Excel files and save them individually. This could be necessary when your sheets are for different months or regions or products/clients, and you want to quickly get a separate workbook for each sheet as either an Excel file or PDF. While manually splitting sheets into separate workbooks and saving them is possible but inefficient and error-prone, we'll provide a simple VBA code that can help you split all the worksheets into their own separate files in just a few seconds. To save worksheets as separate Excel files, follow these steps. First, break down the main Excel file into individual sheets and store them in their own files. This operation is quick if you have fewer worksheets but may take some time for larger collections. Each saved file will retain the original sheet name from the main workbook. Given that a VBA code has been embedded within the Excel workbook, save it with the .XLSM format to preserve and ensure the macro's functionality. This step is crucial as the Application.ScreenUpdating = False and Application.DisplayAlerts = False settings used in the code aim to conceal the processes occurring behind the scenes. After splitting and saving the worksheets, revert these settings back to TRUE. It is highly advisable to create a backup copy of the main Excel file beforehand to prevent data loss due to any potential issues or system crashes. For those preferring PDF format instead of Excel files, modify the code as follows: 'Code Created by Sumit Bansal from trumpexcel.com Sub SplitEachWorksheet() Dim FPath As String FPath = Application.ActiveWorkbook.Path Application.ScreenUpdating = False Application.DisplayAlerts = False For Each ws In ThisWorkbook.Sheets ws.Copy Application.ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, Filename:=FPath & "\" & ws.Name & ".xlsx" Application.ActiveWorkbook.Close False Next Application.DisplayAlerts = True Application.ScreenUpdating = True End Sub Before using this code, ensure the following: 1. Create a folder where you want all resulting files to be saved. 2. Save the main Excel file containing the desired worksheets in that folder. This code splits each sheet into a PDF and saves it within the same folder as the main Excel file. For users with multiple sheets and a specific criterion for splitting, consider the following modification: 'Code Created by Sumit Bansal from trumpexcel.com Sub SplitEachWorksheet() Dim FPath As String Dim TexttoFind As String TexttoFind = "2020" FPath = Application.ActiveWorkbook.Path Application.ScreenUpdating = False Application.DisplayAlerts = False For Each ws In ThisWorkbook.Sheets If InStr(1, ws.Name, TexttoFind, vbBinaryCompare) 0 Then ws.Copy End If Next Application.DisplayAlerts = True Application.ScreenUpdating = True End Sub This code identifies worksheets containing the specified text ("2020" in this example) and saves them as separate Excel files. Filename:=FPath & "\" & ws.Name & ".xlsx" Application.ActiveWorkbook.Close False End If Next Application.DisplayAlerts = True Application.ScreenUpdating = True End Sub In the provided VBA code, a variable named TexttoFind is initially assigned the value "2020". The code then uses a For Next loop to iterate through each worksheet in the active workbook. For each worksheet, it employs the INSTR function to check if the name contains the string "2020". If found, the position of this text is returned; otherwise, 0 is displayed. The IF Then condition is utilized here, where if the specified sheet meets the criteria (i.e., containing the string "2020"), the worksheet name will be split and saved as a separate file. Conversely, if the conditions are not met, nothing occurs, thus avoiding unnecessary actions. It's worth noting that there are alternative methods to accomplish this task, such as combining multiple Excel files into one workbook with various techniques, including: 1. Method 1: Applying Move or Copy Operation 2. Method 2: Combining Multiple Excel Files Using Paste Link Feature 3. Method 3: Utilizing Power Query for Consolidation 4. Method 4: Employing Excel VBA Code Looking for a way to merge multiple workbooks into one? This tutorial can guide you on how to achieve this by using Visual Basic Macro in Excel. The process involves opening multiple files at once, copying the sheets from each file and pasting them into a new workbook.

How to convert multiple excel sheets to pdf. How to convert multiple excel sheets to one pdf files. How to convert multiple excel sheets to single sheet. Excel multiple sheets into one. How do i convert multiple excel sheets to csv. How do i convert multiple excel sheets to pdf. How to convert multiple excel sheets to multiple pdf files. Can you convert multiple excel sheets to pdf. How to convert multiple sheets in excel to one sheet. How to convert multiple excel sheets to multiple pdf files vba. Combine data from multiple sheets.