Monthly Sales Reporting Data Preparation Template

Urban Creations

Import Sales data from Sales Document templates for one or more companies in one go. *(Note: Browse and find the files - file name starts with "Sales_Document_xxx")*

Import Sales

Generate the Regional & Manager Reports for each region. *(Create a pivot based Excel file for Regional managers and a hardcoded file for Top managers)*

Create Report

Europe ▾

Create CSV file of ALL data for the current month to be uploaded to the central system for year-to-date reporting (e.g. SAP)

Create CSV

Before working on the next month's data, clear values and start fresh.
Admin - opens up the hidden tabs in case problems arise.
(note: password = me)

Admin

Clear Values & Tasks

Tasks Completed

```
Sub Import_Sales()

    '1. First import data 1-to-1 to NewData tab
    '2. Use Find function to find the matching header from Summary tab in NewData tab
    '3. Copy and paste the required column from NewData in Summary tab

    Dim shSales As Worksheet
    Dim FileToOpen As Variant
    Dim FileCnt As Byte, c As Byte
    Dim LastRow As Long, LastTempRow As Long
    Dim SelectedBook As Workbook
    Dim GetHeader As Range
    Const StartRowTemp = 4  ' if start row on company templates change, update this line
    Const CompanyRng As String = "C2"
    Dim CompanyImported() As String
    Call Entry_Point

    On Error GoTo Handle:

    'Pick the files to import - allow multiselect
    FileToOpen = Application.GetOpenFilename(Filefilter:="Excel Files (*.xlsx), *.xlsx", Title:="Select Workbook to Import", MultiSelect:=True)

        If IsArray(FileToOpen) Then
            'how many files were selected

            For FileCnt = 1 To UBound(FileToOpen)

                'clean up the data on temporary sheet that were already imported
                shNewDat.Cells.Clear
                'locate last empty row in summary tab
                LastRow = shAll.Cells(Rows.Count, 1).End(xlUp).Row + 1
                Set SelectedBook = Workbooks.Open(FileName:=FileToOpen(FileCnt))

                'in case this sheet doesn't exist, error handle takes care of it by informing the user
                SelectedBook.Worksheets("Sales").Cells.Copy

                shNewDat.Range("A1").PasteSpecial xlPasteValuesAndNumberFormats

                 SelectedBook.Close

                'locate last row in the temp sheet
                LastTempRow = shNewDat.Cells(Rows.Count, 2).End(xlUp).Row

                'assign company name
                'calculate the number of rows that should be copied over and put the company code there
                shAll.Range("A" & LastRow, "A" & LastRow + (LastTempRow - StartRowTemp) - 1).Value = shNewDat.Range(CompanyRng).Value
              '  Debug.Print LastRow + LastTempRow - StartRowTemp - 1
                c = 2  'columns needed for the summary sheet
                'use find function to match headers, since the template could have a different order of columns than the summary tab
                Do While shAll.Cells(1, c) <> ""
                    Set GetHeader = shNewDat.Rows(StartRowTemp).Find(what:=shAll.Cells(1, c).Value, LookIn:=xlValues, MatchCase:=False, lookat:=xlWhole)
                    If Not GetHeader Is Nothing Then
                        shNewDat.Range(shNewDat.Cells(StartRowTemp + 1, GetHeader.Column), shNewDat.Cells(LastTempRow, GetHeader.Column)).Copy
                        shAll.Cells(LastRow, c).PasteSpecial
                    End If
                    c = c + 1 'check and copy next column
               Loop
                ReDim Preserve CompanyImported(1 To FileCnt)
                CompanyImported(FileCnt) = shNewDat.Range("C2").Value
            Next FileCnt

            MsgBox "Data Imported Sucessfully", vbInformation, "Good Job"

        End If
```

```
Option Explicit
Dim myDate As Long
Dim myRegion As String
Dim ReplaceFile As VbMsgBoxResult
Dim Overwrite(1) As Boolean

Sub Create_Reports()
    Dim tbTemp As ListObject
    Dim RowsTb As Long, LastRow As Long

    Call Entry_Point
    Overwrite(0) = True
    Overwrite(1) = True
    On Error GoTo Handle
    'create an object variable for the table inside Region Temp tab
    Set tbTemp = shRegion.ListObjects("TableTemp")
    RowsTb = tbTemp.DataBodyRange.Rows.Count
    'Remove existing data in regional table except for line 1 if there is more than 1 line of data
    If RowsTb > 1 Then tbTemp.DataBodyRange.Rows("2:" & RowsTb).Delete
    LastRow = shAll.Range("A" & Rows.Count).End(xlUp).Row
    myRegion = shStart.AxRegion.Value
    myDate = Format(shAll.Range("C2").Value, "YYYYMM")

    'find the region for each company for the summary tab
    shAll.Range("I2").FormulaR1C1 = "=VLOOKUP(RC[-8],MCompany,4,FALSE)"
    shAll.Range("I2").AutoFill Destination:=shAll.Range("I2:I" & LastRow)
    Application.Calculate

    With shAll.Range("A1:I" & LastRow)
        .AutoFilter Field:=9, Criteria1:=myRegion
        .Offset(1).Resize(.Rows.Count - 1, .Columns.Count - 1).SpecialCells(xlCellTypeVisible).Copy
    End With

    'paste the values
    shRegion.Range("A2").PasteSpecial
     shAll.Range("A1").AutoFilter
     shAll.Columns("I").Delete
'    Debug.Print Timer - Start
    shRegion.Range("U1").Value = myDate
    Application.Calculate
    shRegion.PivotTables("PTRegion").PivotCache.Refresh
    Call Region_Report
    Call Manager_Report
    If Overwrite(0) = False And Overwrite(1) = False Then
        MsgBox "No filese were created.", , "Operation Cancelled"
    Else
        MsgBox "File(s) created in the same directory as this workbook.", , "Well Done"

    End If
    Call Exit_Point
    Exit Sub

Handle:
            If Err.Number = 1004 Then
            MsgBox "It seems the data for this region has not been uploaded yet! Please upload and try again."
            shAll.Range("A1").AutoFilter
            shAll.Columns("I").Delete
            Exit Sub
        Else
            MsgBox "Looks like no data has been imported yet.", vbInformation
            Exit Sub
        End If
    Call Exit_Point
```

```vba
Private Sub Region_Report()
    Dim NewRbook As Workbook
    Dim newRpath As String
    Dim PT As PivotTable
    newRpath = ThisWorkbook.Path & "\" & "RM_" & myDate & myRegion & ".xlsx"
    If Len(Dir(newRpath)) > 0 Then 'file exists
        ReplaceFile = MsgBox("Regional File Exists. Do you want to overwrite?", vbOKCancel, "Overwrite?")
        If ReplaceFile = vbCancel Then
            Overwrite(0) = False
            Exit Sub
        End If
    End If
    Set NewRbook = Workbooks.Add
    shRegion.Copy Before:=NewRbook.Sheets(1)
    'remove common filter for slicer
    NewRbook.SlicerCaches("Slicer_Company_Name").PivotTables.RemovePivotTable _
    (ActiveSheet.PivotTables("PTArticle"))
    'change pivot cache
    NewRbook.ActiveSheet.PivotTables("PTRegion").ChangePivotCache _
    NewRbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:="TableTemp")

    For Each PT In NewRbook.ActiveSheet.PivotTables
        If PT.Name <> "PTRegion" Then
            PT.ChangePivotCache ("PTRegion")
        End If
    Next PT

    'add back the common slicer
    NewRbook.SlicerCaches("Slicer_Company_Name").PivotTables.AddPivotTable _
    (ActiveSheet.PivotTables("PTArticle"))
    'some cleaning up
    With ActiveSheet
        .PivotTables("PTRegion").SaveData = True
        .Columns("AJ:AZ").Delete
        .Range("TableTemp").CurrentRegion.Columns.Group
        .Outline.ShowLevels RowLevels:=0, ColumnLevels:=1
        .Range("A1").CurrentRegion.Copy
        .Range("A1").PasteSpecial (xlPasteValues)

    End With

    NewRbook.SaveAs FileName:=newRpath
    NewRbook.Close
    TaskRow = shStart.Range("B" & shStart.Rows.Count).End(xlUp).Row + 1
    shStart.Range("B" & TaskRow).Value = "Regional Report for " & myRegion & myDate & " - created on: " & Now


End Sub
```

```vba
Private Sub Manager_Report()

    Dim newBook As Workbook
    Dim newPath As String

    newPath = ThisWorkbook.Path & "\" & "M_" & myDate & myRegion & ".xlsx"
    If Len(Dir(newPath)) > 0 Then 'file exists
        ReplaceFile = MsgBox("Manager File Exists. Do you want to overwrite?", vbOKCancel, "Overwrite?")
        If ReplaceFile = vbCancel Then
            Overwrite(1) = False
            Exit Sub
        End If
    End If
    Set newBook = Workbooks.Add
     ActiveWindow.DisplayGridlines = False
```