**Notes:**

============================================================

The macro below is included in my portfolio to demonstrate my ability to work with complex VBA automation for business reporting. The script shown here is non-active and uses only dummy references — proprietary or sensitive details have been cleaned or aliased to ensure confidentiality.

**Purpose**

The original version of this macro was designed to produce daily metrics reports for a procurement department, enabling purchasing staff to make timely and informed decisions. Data was extracted directly from SAP, processed in Excel, and distributed in a consistent, automated format. A visual image of the final product is available in my portfolio.

**Core Competencies Demonstrated**

Connecting to an external system: Establishing programmatic access to SAP via scripting.

Data extraction & export: Automating transaction runs and exporting multiple reports (e.g., purchase requisitions, purchase orders, backorders, MRP orders).

Data consolidation: Combining multiple sources into a single reporting workbook.

Data transformation: Applying filters, cleaning values, and structuring the results for readability.

Visualization & preparation: Formatting data into a consistent reporting template.

Report transmission: (in the original version) emailing the report to a distribution list automatically.

**Notes for Reviewers**

This is heavy automation code — intended to reduce manual daily effort for a high-volume procurement environment.

The presented version emphasizes clarity, modularity, and professional standards (indenting, commenting, variable declarations).

Sensitive identifiers (e.g., plant codes, user directories, filenames) have been replaced with aliases.

The purpose of sharing this code is to highlight competency in building end-to-end automation solutions in VBA, not to provide an active production tool.

'============================================================

Sub AES\_Fixer()

'--------------------------

' Variable Declarations

'--------------------------

Dim myStrDate As String

Dim myDate As Date

Dim myStrDate2 As String

Dim myDate2 As Date

Dim YesterdayStrDate As String

Dim Yesterday As Date

Dim TodayStrDate As String

Dim Today As Date

Dim sh As Worksheet

Dim Wb As Workbook

Dim Rng As Range

Dim LastFridayofMonthStrDate As String

Dim LastFridayofMonth As Date

Dim LastDayofYearStrDate As String

Dim LastDayofYear As Date

' SAP Automation Objects

Dim SapGuiAuto As Object

Dim App As Object

Dim Connection As Object

Dim session As Object

Dim WshShell As Object

' Other supporting variables

Dim sourceSheet66 As Workbook

Dim lastrow As Long

'--------------------------

' Date Handling

'--------------------------

myStrDate = DateAdd("d", -2, Date) ' Two days before today

myStrDate2 = Date ' Today’s date

myDate = CDate(myStrDate) ' Convert string to Date

myDate2 = CDate(myStrDate2)

LastDayofYearStrDate = "12/31/" & Year(Date) ' Build last day of current year

LastDayofYear = CDate(LastDayofYearStrDate)

YesterdayStrDate = DateAdd("d", -1, Date) ' Yesterday’s date

TodayStrDate = Date ' Today’s date

Yesterday = CDate(YesterdayStrDate)

Today = CDate(TodayStrDate)

'--------------------------

' Open Report Template

'--------------------------

Set Wb = Workbooks("Report\_Template.xlsm")

'--------------------------

' Connect to SAP

'--------------------------

Set SapGuiAuto = GetObject("SAPGUI")

Set App = SapGuiAuto.GetScriptingEngine

Set Connection = App.Children(0)

Set session = Connection.Children(0)

Set WshShell = CreateObject("WScript.Shell")

Application.DisplayAlerts = False

'============================================================

' BACKORDER DATA EXTRACTION

'============================================================

session.findById("wnd[0]").maximize ' Maximize SAP window

session.findById("wnd[0]/tbar[0]/okcd").Text = "YMMR\_PO\_EXP" ' Enter transaction code

session.findById("wnd[0]/tbar[0]/btn[0]").press ' Execute transaction

session.findById("wnd[0]/usr/ctxtSO\_EKGRP-LOW").Text = "" ' Clear Purchasing Group field

session.findById("wnd[0]/tbar[0]/btn[0]").press ' Run with cleared group

If session.findById("wnd[0]/usr/ctxtP\_WERKS").Text = "PlantCode01" Then

'--------------------------

' PlantCode01 specific logic

'--------------------------

session.findById("wnd[0]/usr/ctxtP\_EKORG").Text = "PlantCode01"

session.findById("wnd[0]/usr/ctxtP\_DATE").Text = "12/31/2025"

session.findById("wnd[0]/usr/ctxtP\_DATE").SetFocus

session.findById("wnd[0]/usr/ctxtP\_DATE").caretPosition = 10

' Open BSTYP filter window

session.findById("wnd[0]/usr/btn%\_SO\_BSTYP\_%\_APP\_%-VALU\_PUSH").press

session.findById("wnd[1]/tbar[0]/btn[16]").press ' Select All

session.findById("wnd[1]/tbar[0]/btn[8]").press ' Continue

' Open BSART filter window

session.findById("wnd[0]/usr/btn%\_SO\_BSART\_%\_APP\_%-VALU\_PUSH").press

session.findById("wnd[1]/tbar[0]/btn[16]").press ' Select All

session.findById("wnd[1]/tbar[0]/btn[8]").press ' Continue

' Apply filter for BSTYP (set F and L)

session.findById("wnd[0]/usr/btn%\_SO\_BSTYP\_%\_APP\_%-VALU\_PUSH").press

session.findById("wnd[1]/usr/tabsTAB\_STRIP/tabpSIVA/ssubSCREEN\_HEADER:SAPLALDB:3010/tblSAPLALDBSINGLE/ctxtRSCSEL\_255-SLOW\_I[1,0]").Text = "F"

session.findById("wnd[1]/usr/tabsTAB\_STRIP/tabpSIVA/ssubSCREEN\_HEADER:SAPLALDB:3010/tblSAPLALDBSINGLE/ctxtRSCSEL\_255-SLOW\_I[1,1]").Text = "L"

session.findById("wnd[1]/tbar[0]/btn[8]").press ' Apply filter

session.findById("wnd[0]/tbar[1]/btn[8]").press ' Execute

session.findById("wnd[0]/tbar[1]/btn[33]").press ' Choose Layout

session.findById("wnd[1]").Close ' Close layout selection

' Save results to file

session.findById("wnd[0]/tbar[1]/btn[32]").press ' Export

session.findById("wnd[1]/usr/tabsG\_TS\_ALV/tabpALV\_M\_R1/...").selectedRows = "0-7"

session.findById("wnd[1]/usr/tabsG\_TS\_ALV/.../btnAPP\_WL\_SING").press

session.findById("wnd[1]/tbar[0]/btn[0]").press

session.findById("wnd[0]/tbar[1]/btn[45]").press ' Export to local

session.findById("wnd[1]/usr/ctxtDY\_PATH").Text = "C:\SAP\_Exports"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").Text = "BackOrder.XLS"

session.findById("wnd[1]/tbar[0]/btn[11]").press ' Confirm export

Else

'--------------------------

' Logic for other Plants

'--------------------------

session.findById("wnd[0]/usr/ctxtP\_WERKS").Text = "PlantCode01"

session.findById("wnd[0]/usr/ctxtP\_EKORG").Text = "PlantCode01"

session.findById("wnd[0]/usr/ctxtP\_DATE").Text = "12/31/2022"

' (Similar steps as above for filters and export...)

End If

' End of conditional SAP navigation block

'============================================================

' PURCHASE REQUISITION (PReq) DATA

'============================================================

session.findById("wnd[0]/tbar[0]/okcd").Text = "ME5A" ' Enter transaction ME5A

session.findById("wnd[0]/tbar[0]/btn[0]").press ' Execute

' Configure selection screen

session.findById("wnd[0]/usr/ctxtP\_LSTUB").Text = "ALV" ' Use ALV output list

session.findById("wnd[0]/usr/ctxtS\_WERKS-LOW").Text = "PlantCode01" ' Plant code filter

session.findById("wnd[0]/tbar[1]/btn[8]").press ' Run the report

'------------------------------------------------------------

' Apply Filters in ALV output

'------------------------------------------------------------

' Filter: UOM (Unit of Measure) not equal to blank

session.findById("wnd[0]/tbar[1]/btn[28]").press ' Filter button

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "UOM"

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "<>"

session.findById("wnd[1]/tbar[0]/btn[0]").press ' Apply filter

' Filter: Release Date <= Today

session.findById("wnd[0]/tbar[1]/btn[28]").press

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "REL\_DATE"

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "<=" & Today

session.findById("wnd[1]/tbar[0]/btn[0]").press

' Exclusion: Exclude PlantCode02

session.findById("wnd[0]/tbar[1]/btn[28]").press

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "WERKS"

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "<>PlantCode02"

session.findById("wnd[1]/tbar[0]/btn[0]").press

'------------------------------------------------------------

' Export PReq results

'------------------------------------------------------------

session.findById("wnd[0]/tbar[1]/btn[45]").press ' Export

session.findById("wnd[1]/usr/ctxtDY\_PATH").Text = "C:\SAP\_Exports"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").Text = "PReqData.XLS"

session.findById("wnd[1]/tbar[0]/btn[11]").press ' Confirm export

'============================================================

' PURCHASE ORDER (PO) DATA

'============================================================

session.findById("wnd[0]/tbar[0]/okcd").Text = "ME2N" ' Enter ME2N

session.findById("wnd[0]/tbar[0]/btn[0]").press

' Configure selection screen for PO data

session.findById("wnd[0]/usr/ctxtS\_WERKS-LOW").Text = "PlantCode01" ' Plant filter

session.findById("wnd[0]/usr/ctxtS\_EKGRP-LOW").Text = "" ' Purchasing Group

session.findById("wnd[0]/usr/ctxtS\_BEDAT-LOW").Text = myDate ' From date

session.findById("wnd[0]/usr/ctxtS\_BEDAT-HIGH").Text = myDate2 ' To date

session.findById("wnd[0]/tbar[1]/btn[8]").press ' Run report

'------------------------------------------------------------

' Apply PO Filters

'------------------------------------------------------------

' Exclude plant PlantCode02

session.findById("wnd[0]/tbar[1]/btn[28]").press

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "WERKS"

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "<>PlantCode02"

session.findById("wnd[1]/tbar[0]/btn[0]").press

'------------------------------------------------------------

' Export PO results

'------------------------------------------------------------

session.findById("wnd[0]/tbar[1]/btn[45]").press ' Export

session.findById("wnd[1]/usr/ctxtDY\_PATH").Text = "C:\SAP\_Exports"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").Text = "POData.XLS"

session.findById("wnd[1]/tbar[0]/btn[11]").press ' Confirm export

'============================================================

' MRP ORDERS (Planned Orders / Production Orders)

'============================================================

session.findById("wnd[0]/tbar[0]/okcd").Text = "ZMM\_MRP\_ORDREP" ' Enter transaction ZMM\_MRP\_ORDREP

session.findById("wnd[0]/tbar[0]/btn[0]").press ' Execute

' Configure selection screen

session.findById("wnd[0]/usr/ctxtS\_WERKS-LOW").Text = "PlantCode01" ' Plant filter

session.findById("wnd[0]/usr/ctxtS\_DISPO-LOW").Text = "" ' Clear MRP controller

session.findById("wnd[0]/usr/ctxtS\_BDATU-LOW").Text = myDate ' Date From

session.findById("wnd[0]/usr/ctxtS\_BDATU-HIGH").Text = myDate2 ' Date To

session.findById("wnd[0]/tbar[1]/btn[8]").press ' Run report

'------------------------------------------------------------

' Apply Filters

'------------------------------------------------------------

' Exclude PlantCode02

session.findById("wnd[0]/tbar[1]/btn[28]").press

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "WERKS"

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "<>PlantCode02"

session.findById("wnd[1]/tbar[0]/btn[0]").press

'------------------------------------------------------------

' Export MRP Orders

'------------------------------------------------------------

session.findById("wnd[0]/tbar[1]/btn[45]").press ' Export

session.findById("wnd[1]/usr/ctxtDY\_PATH").Text = "C:\SAP\_Exports"

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").Text = "MRPOrders.XLS"

session.findById("wnd[1]/tbar[0]/btn[11]").press ' Confirm export

'============================================================

' LAUNCH EXTERNAL VBS SCRIPT

'============================================================

' Script is used to automate additional data pulls or formatting

Shell "wscript.exe ""C:\SAP\_Exports\AES\_WBS\_VBS.vbs"""

'============================================================

' COPY WBS LIST INTO EXCEL

'============================================================

' Open supporting workbook containing Work Breakdown Structure (WBS) data

Set sourceSheet66 = Workbooks.Open("C:\SAP\_Exports\WBS\_List.xlsx")

' Find last row of data in column A

lastrow = sourceSheet66.Sheets("Sheet1").Cells(Rows.Count, 1).End(xlUp).Row

' Copy the WBS values

sourceSheet66.Sheets("Sheet1").Range("A1:A" & lastrow).Copy

' Paste will occur into the active workbook (Report\_Template.xlsm) by user’s logic

' Close source workbook after copying (optional: keep open if needed)

sourceSheet66.Close SaveChanges:=False

'============================================================

' GL FILTERS OR ADDITIONAL SAP REPORTS (IF ANY)

'============================================================

' Example: If your process includes filtering by GL account,

' follow the same filter steps here.

' session.findById("wnd[0]/tbar[1]/btn[28]").press

' session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "GL\_ACCOUNT"

' session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = "<>123456"

' session.findById("wnd[1]/tbar[0]/btn[0]").press

' Export logic can follow the same pattern:

' session.findById("wnd[0]/tbar[1]/btn[45]").press

' session.findById("wnd[1]/usr/ctxtDY\_PATH").Text = "C:\SAP\_Exports"

' session.findById("wnd[1]/usr/ctxtDY\_FILENAME").Text = "GLReport.XLS"

' session.findById("wnd[1]/tbar[0]/btn[11]").press

'============================================================

' CLEAN-UP / WRAP UP

'============================================================

' Any final workbook manipulations can go here, e.g.:

' - Pasting copied WBS data into Report\_Template

' - Formatting sheets

' - Saving consolidated workbook

' Example:

' Wb.Sheets("WBS").Range("A1").PasteSpecial Paste:=xlPasteValues

' Finally, save the main report workbook if required

' Wb.Save

End Sub

'============================================================

' HELPER FUNCTIONS - MODULARIZED SAP ACTIONS

'============================================================

Private Sub RunTransaction(transactionCode As String)

' Generic way to run SAP transactions

session.findById("wnd[0]/tbar[0]/okcd").Text = transactionCode

session.findById("wnd[0]/tbar[0]/btn[0]").press

End Sub

Private Sub ExportToExcel(exportPath As String, fileName As String)

' Standardized SAP export routine

session.findById("wnd[1]/usr/ctxtDY\_PATH").Text = exportPath

session.findById("wnd[1]/usr/ctxtDY\_FILENAME").Text = fileName

session.findById("wnd[1]/tbar[0]/btn[11]").press

End Sub

Private Sub ApplyFilter(filterField As String, filterValue As String)

' Abstracted filter logic

' Example usage:

' ApplyFilter "WERKS", "<>PlantCode02"

session.findById("wnd[0]/tbar[1]/btn[28]").press

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = filterField

session.findById("wnd[1]/usr/tabsG\_SELONETABSTRIP/...").Text = filterValue

session.findById("wnd[1]/tbar[0]/btn[0]").press

End Sub