Option Explicit

'' DECLARE GLOBAL VARIABLES ''

Dim i As Long, lastMainRow As Long, lastHistRow As Long, lastParamRow As Long

Dim j As Integer, k As Integer

Dim ParamWB As Workbook

Dim ParamWS As Worksheet

Dim wsMain As Worksheet, wsHist As Worksheet

Dim rngSearch As Range, rngFound As Range

Dim foundArr() As Variant

Const firstMainRow As Integer = 9

Const PKEY\_COL As Integer = 1

Const BOOK\_COL As Integer = 3

Const REF\_1\_COL As Integer = 8

Const REF\_2\_COL As Integer = 9

Const CUST\_COL As Integer = 10

Const SCHEME\_NO\_COL As Integer = 11

Const CONTRACT\_DATE\_COL As Integer = 12

Const CCY1\_COL As Integer = 28

Const AMT1\_COL As Integer = 29

Const FIXED\_IR1\_COL As Integer = 35

Const CCY2\_COL As Integer = 39

Const AMT2\_COL As Integer = 40

Const FIXED\_IR2\_COL As Integer = 46

Const CF\_DATES\_COL As Integer = 48

Const OBS\_REF\_COL As Integer = 49

Const TOTAL\_COL\_NO As Integer = 15

Sub ParamFile()

'' (A) DECLARE LOCAL VARIABLES ''

Dim mainArr() As Variant, mainSubArr() As Variant, slice As Variant, colArr As Variant, dateRangeArr(0 To 2) As Variant

Dim filepath As String

Dim validateWB As Workbook

Dim validateWSArr As Variant, validateWS As Worksheet

Dim firstValidateRow As Integer, lastValidateRow As Integer

Dim rowSeq As Integer

'' (B) ASSIGN LOCAL VARIABLES ''

Set wsMain = ThisWorkbook.Sheets("Main")

Set wsHist = ThisWorkbook.Sheets("Historical")

lastMainRow = wsMain.Cells(Rows.Count, 1).End(xlUp).row

lastHistRow = wsHist.Cells(Rows.Count, 1).End(xlUp).row

Set rngSearch = wsHist.Range("A:A")

colArr = Array(PKEY\_COL, BOOK\_COL, REF\_1\_COL, REF\_2\_COL, CUST\_COL, \_

SCHEME\_NO\_COL, CONTRACT\_DATE\_COL, CCY1\_COL, AMT1\_COL, FIXED\_IR1\_COL, \_

CCY2\_COL, AMT2\_COL, FIXED\_IR2\_COL, CF\_DATES\_COL, OBS\_REF\_COL)

dateRangeArr(0) = wsMain.Cells(2, 5)

dateRangeArr(1) = wsMain.Cells(3, 5)

dateRangeArr(2) = wsMain.Cells(4, 5)

Application.DisplayAlerts = False

' Open TOD's Validation File from designated filepath

filepath = "H:\Derivatives Section\Banking New Ref\*.xls"

Set validateWB = Workbooks.Open(filepath)

validateWSArr = Array(3, 2, 1, 5, 4, 7, 6, 8)

'' (C) PERFORM MANDATORY CHECKS AGAINST TOD'S VALIDATION FILE ''

' The purpose of the mandatory checks is to identify:

' (1) New deals - Outcome: Updated cells would be highlighted in yellow

' (2) Original deals that have been amended by AOTD Banking team - Outcome: Updated cells would be highlighted in yellow

' (3) Incorrect GBase reference numbers used by AOTD Banking team - Outcome: MsgBox will be generated

' If lastMainRow - firstMainRow = 0 Then

' ' Assign mainArr as 1D array to contain specific swap details

' ReDim Preserve mainArr(TOTAL\_COL\_NO - 1)

'

' ' Create mainArr containing specific swap details

' For j = LBound(mainArr) To UBound(mainArr)

' mainArr(j) = wsMain.Cells(firstMainRow, colArr(j))

' Next j

'

' ElseIf lastMainRow - firstMainRow > 0 Then

' ' Assign mainArr as 2D array to contain specific swap details

' ReDim Preserve mainArr(lastMainRow - firstMainRow, TOTAL\_COL\_NO - 1)

' i = firstMainRow

'

' ' Create mainArr containing specific swap details

' For j = LBound(mainArr, 1) To UBound(mainArr, 1)

' For k = LBound(mainArr, 2) To UBound(mainArr, 2)

' mainArr(j, k) = wsMain.Cells(i, colArr(k))

' Next k

' i = i + 1

' Next j

'

' Else

' MsgBox "There are no swap deals for the stipulated period of T-2 and T+0!"

'

' End If

' ReDim Preserve mainSubArr(lastMainRow - firstMainRow)

' mainSubArr(0) = wsMain.Cells(rowSeq, CONTRACT\_DATE\_COL)

' mainSubArr(1) = wsMain.Cells(rowSeq, SCHEME\_NO\_COL)

' mainSubArr(2) = wsMain.Cells(rowSeq, REF\_1\_COL) & CStr(wsMain.Cells(rowSeq, BOOK\_COL)) & CStr(wsMain.Cells(rowSeq, REF\_2\_COL))

' mainSubArr(3) = wsMain.Cells(rowSeq, CUST\_COL)

' mainSubArr(4) = wsMain.Cells(rowSeq, OBS\_REF\_COL)

j = 0

For rowSeq = firstMainRow To lastMainRow

If lastMainRow - firstMainRow = 0 Then

' Create 1D mainSubArr containing specific swap details

ReDim Preserve mainSubArr(4)

mainSubArr(0) = wsMain.Cells(rowSeq, CONTRACT\_DATE\_COL)

mainSubArr(1) = wsMain.Cells(rowSeq, SCHEME\_NO\_COL)

mainSubArr(2) = wsMain.Cells(rowSeq, REF\_1\_COL) & CStr(wsMain.Cells(rowSeq, BOOK\_COL)) & CStr(wsMain.Cells(rowSeq, REF\_2\_COL))

mainSubArr(3) = wsMain.Cells(rowSeq, CUST\_COL)

mainSubArr(4) = wsMain.Cells(rowSeq, OBS\_REF\_COL)

ElseIf lastMainRow - firstMainRow > 0 Then

' Create 2D mainSubArr containing specific swap details

ReDim Preserve mainSubArr(lastMainRow - firstMainRow, 1)

i = firstMainRow

For j = LBound(mainSubArr, 1) To UBound(mainSubArr, 1)

For k = LBound(mainSubArr, 2) To UBound(mainSubArr, 2)

mainSubArr(j, 0) = wsMain.Cells(rowSeq, CONTRACT\_DATE\_COL)

mainSubArr(j, 1) = wsMain.Cells(rowSeq, SCHEME\_NO\_COL)

' Perform 1st level check against TOD's Validation File, i.e. validateWSArr(0), to validate Scheme Number and Swap Reference Number

validateWS = validateWB.Worksheets(validateWSArr(0))

FirstCheck dateRangeArr, wsMain, mainSubArr, validateWS

Next rowSeq

If UBound(mainArr, 1) >= 0 And UBound(foundArr, 1) <> -1 Then

Else

MsgBox "The number of transactions in New Ref file is more than the number of transactions extracted from OBS for the same T-2 to T+0 period. Please check!"

End If

''https://stackoverflow.com/questions/30066775/comparing-2-dimension-arrays-in-vb-excel

' '' (D) MATCH PKEY BETWEEN WSMAIN AND WSHIST ''

' '''PLEASE WORK ON AMENDING THIS

' While firstMainRow <= lastMainRow

' Set rngFound = rngSearch.Find(wsMain.Cells(firstMainRow, wsMain.Cells.Find(What:="PKEY", LookAt:=xlWhole).Column), After:=rngSearch.Cells(rngSearch.Cells.Count), \_

' LookIn:=xlValues, LookAt:=xlWhole, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:=False, SearchFormat:=False)

'

' ' To ignore transaction if there is a match, i.e. swap deal has been booked into GBase

' If Not rngFound Is Nothing Then

' ' Do nothing

' Else

' ' Store row number in mainArr()

' mainArr(j) = firstMainRow

' j = j + 1

' ReDim Preserve mainArr(j)

' End If

' firsttMainRow = firstMainRow + 1

' Wend

'

' Application.DisplayAlerts = False

'

' '' (E) OPEN PARAMETER FILE AND SAVE SWAP DEALS ''

' If UBound(mainArr) <> 0 Then

' Set ParamWB = Workbooks.Open(Filename:=ThisWorkbook.Path & "\" & "GBASE\_Parameters.xlsx", Format:=xlDelimited, Local:=True)

' Set ParamWS = ParamWB.Sheets("Parameters")

' lastParamRow = ParamWS.Cells(Rows.Count, 1).End(xlUp).row

'

' For j = LBound(mainArr) To UBound(mainArr) - 1

' wsMain.Range(wsMain.Cells(mainArr(j), wsMain.Cells.Find(What:="PKEY", LookAt:=xlWhole).Column), \_

' wsMain.Cells(mainArr(j), wsMain.Cells.Find(What:="DATES\_PAY\_F03", LookAt:=xlWhole).Column)).Copy

' ParamWS.Range("A" & lastParamRow + 1).PasteSpecial xlPasteValues

' ParamWS.Range("A" & lastParamRow + 1).PasteSpecial xlPasteFormats

' lastParamRow = lastParamRow + 1

' Next j

'

' '' (F) SAVE AND CLOSE PARAMETER FILE ''

' ParamWB.Save

' ParamWB.Close

' End If

'

Application.DisplayAlerts = True

End Sub

Sub UpdateHist()

'' (A) ASSIGN LOCAL VARIABLES ''

Set wsHist = ThisWorkbook.Sheets("Historical")

lastHistRow = wsHist.Cells(Rows.Count, 1).End(xlUp).row

Application.DisplayAlerts = False

'' (B) OPEN PARAMETER FILE AND SAVE SWAP DEALS IN WSHIST AFTER GBASE UPDATE ''

If lastParamRow >= 2 Then

Set ParamWB = Workbooks.Open(Filename:=ThisWorkbook.Path & "\" & "GBASE\_Parameters.xlsx", Format:=xlDelimited, Local:=True)

Set ParamWS = ParamWB.Sheets("Parameters")

lastParamRow = ParamWS.Cells(Rows.Count, 1).End(xlUp).row

ParamWS.Range("A2:AV" & lastParamRow + 1).Copy

wsHist.Range("A" & lastHistRow + 1).PasteSpecial xlPasteValues

wsHist.Range("A" & lastHistRow + 1).PasteSpecial xlPasteFormats

'' (C) CLEAR PARAMWS ''

ParamWS.Range("A2:AV" & lastParamRow + 1).ClearContents

'' (D) SAVE AND CLOSE PARAMETER FILE ''

ParamWB.Save

ParamWB.Close

End If

Application.DisplayAlerts = True

wsMain.Select

End Sub

Sub FirstCheck(dateArr, wsMain, mainArr, validateWS)

'' (A) DECLARE LOCAL VARIABLES ''

Dim dateSearchRange As Range

Dim lastRow As Integer, row As Integer, datePosition As Integer

'' (B) ASSIGN LOCAL VARIABLES ''

lastRow = validateWS.Cells(Rows.Count, 1).End(xlUp).row

'' (C) DETERMINE WHETHER DATE HAS BEEN RECORDED IN TOD'S VALIDATION FILE ''

For datePosition = LBound(dateArr) To UBound(dateArr)

With validateWS.Range("A:A")

Set dateSearchRange = .Find(What:=Format(dateArr(datePosition), "dd-mmm-yy"), \_

After:=.Cells(.Cells.Count), \_

LookIn:=xlValues, \_

LookAt:=xlWhole, \_

SearchOrder:=xlByRows, \_

SearchDirection:=xlNext, \_

MatchCase:=False)

If Not dateSearchRange Is Nothing Then

' If common date is found, i.e. recorded previously in TOD's file, append these details:

' (i) Contract Date

' (ii) Scheme Number

' (iii) Swap Reference Number

' (iv) Counterparty

' (v) OBS Reference Number

' Assign dimension of foundArr

ReDim Preserve foundArr(5)

For row = dateSearchRange.row To lastRow

' Append details in foundArr

foundArr(0) = row

foundArr(1) = .Cells(row, 1)

If InStr(.Cells(row, 2), "INTSWP") <> 0 Then

foundArr(2) = "SPCHF" & CStr(Format(.Cells(row, 3), "000000"))

Else

foundArr(2) = CStr(.Cells(row, 2)) & CStr(Format(.Cells(row, 3), "000000"))

End If

foundArr(3) = .Cells(row, 4)

foundArr(4) = .Cells(row, 5)

foundArr(5) = .Cells(row, 7)

' Validate each row details in both mainSubArr and foundArr

If mainSubArr(2) = foundArr(2) Then

'If IsError(Application.WorksheetFunction.Match(mainSubArr(1), )

End If

Next row

Next k

Next j

End If

' Exit the first For Loop

Exit For

Else

' If no common date is found, i.e. transactions are newly created, return empty array

foundArr = Array()

End If

End With

Next datePosition

End Sub

Sub PrelimCheck(txnArr, datesArr)

lastMainRow = wsMain.Cells(Rows.Count, 1).End(xlUp).row

ReDim Preserve wsMainTxn(lastMainRow - 2, 4)

End Sub

‘https://stackoverflow.com/questions/30066775/comparing-2-dimension-arrays-in-vb-excel