

**VB Code For Excel Alignments:**

**1.vb code for clear old value In Existing excel sheet:**

Sub Clear()

Dim Ws As Worksheet //variable declaration

For Each Ws In ActiveWorkbook.Worksheets

Ws.Cells.ClearContents

Next Ws

End Sub

**Input: --🡪 Output:**

|  |  |  |
| --- | --- | --- |
| **name** | **job** | **age** |
| Guru | server | 23 |
| Siva | head | 45 |
| Shankar | salesman | 67 |
| Muthu | Tl | 32 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\* Above code used for remove all the row values from existing sheet.**

**--------------------------------------------------------------------------------------------------------------------------------**

**2.Delete the column in Excel which you need:**

**Sub DeleteRRR()**

**Columns(3).Delete or Range(“C:E,H:J”).Delete**

**End Sub**

**Input: --🡪 Output:**

|  |  |  |
| --- | --- | --- |
| **name** | **job** | **age** |
| Guru | server | 23 |
| Siva | head | 45 |
| Shankar | salesman | 67 |
| Muthu | Tl | 32 |

|  |  |
| --- | --- |
| **name** | **job** |
| Guru | server |
| Siva | head |
| Shankar | salesman |
| Muthu | Tl |

**\*It is used for removing the one column from excel.**

**------------------------------------------------------------------------------------------------------------------------------**

**3.For Each Code In VBA:**

**Sub froeachmodel()**

**For Each Number In Range("A2:A9")**

**If Number > 50 Then**

**Number.Offset(0, 2).Value = "pass"**

**Else**

**Number.Offset(0, 1).Value = "Fail"**

**End If**

**Next Number**

**End Sub**

**-------------------------------------------------------------------------------------------------------------------------**

**4. Color apply:**

**Range("A2:F2").Interior.Color = RGB(218, 238, 243)**

**Range("C21:E21").Interior.ColorIndex = #B66DFF**

**---------------------------------------------------------------------------------------------------------**

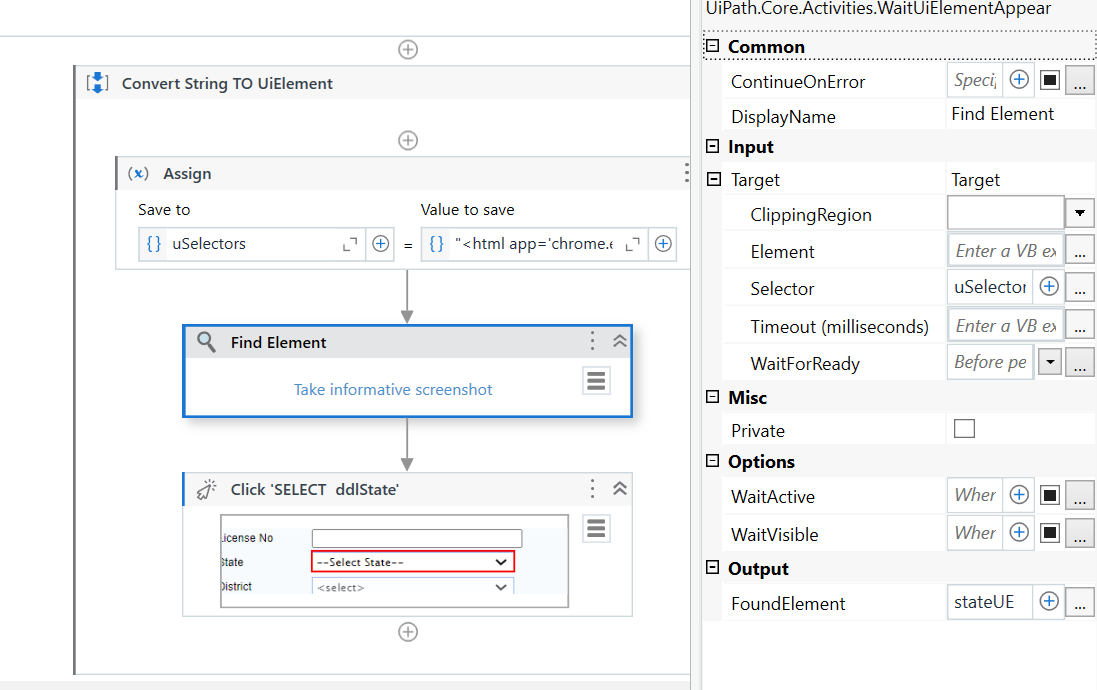
**5.Convert the String To Uielement and Pass The Full Dynamic Selectors:**

**1. Store the selectors in string Variable**

**2.Then pass The String Variable in Selectors Field that is properties Of Find element activity.**

**3.create the output variable of find element.**

**4.it will pass in element field of click activity.**

**6.Transpose the data:**

**Sub Transpose()**

**Sheet1.Activiate**

**Sheet1.UsedRange.Select**

**Selection.Copy**

**Sheet2.Activate**

**End Sub**

**input:**

|  |  |
| --- | --- |
| ZONE | SOUTH |
| COM\_PERC | 0.84 |
| BRANCH | VIJAYAWADAB |
| TARGET | 20 |
| COMPLETION | 16.81 |
| FTD | 0 |

**output:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ZONE | COM\_PERC | BRANCH | TARGET | COMPLETION | FTD |
| SOUTH | 0 | VIZAGB | 3 | 0 | 0 |

**7. Used For Each IN Excel Range:**

**Sub froeachmodel()**

**For Each Number In Range("A2:A9")**

**If Number > 50 Then**

**Number.Offset(0, 2).Value = "pass"**

**Else**

**Number.Offset(0, 1).Value = "Fail"**

**End If**

**Next Number**

**End Sub**

**---------------------------------------------------------------------------------------------------------------------------------------------**

**8.ADD one column and update the values in Vba code:**

**Dim row As DataRow**

**in\_inputDt.Columns.Add("Total")**

**For Each row In in\_inputDt.Rows**

**row("Total")=(Convert.ToInt64(row("price1")))+(Convert.ToInt64(row("price2")))**

**Next row**

**out\_outputDt=in\_inputDt**

**input: Output:**

|  |  |
| --- | --- |
| Price1 | price2 |
| 100 | 10 |
| 200 | 20 |
| 300 | 30 |

|  |  |  |
| --- | --- | --- |
| Price1 | price2 | Total |
| 100 | 10 | 110 |
| 200 | 20 | 220 |
| 300 | 30 | 330 |

VB Code For Learning Purpose ;

Dim row As DataRow

'23 Column Preparation

Dim dt\_referenceTable As DataTable

dt\_referenceTable = io\_dt\_input.Clone

dt\_referenceTable = io\_dt\_input

Dim cancellationWithinSameMonth, cancelAfterGST, cancelBeforeGST, NPzero As String

cancellationWithinSameMonth ="Cancellation within same month"

cancelAfterGST = "Cancellation After GST Period"

cancelBeforeGST = "Cancellation After GST Period"

NPzero = ""

Dim reg, regEn, UnReg, policyWith18And12, policyWith12, policyWith18, invalidColumns As String

reg = "Reg"

regEn = "RegEN"

UnReg = "UnReg"

policyWith18And12 = "Policy With 18%&12%"

policyWith12 ="Policy With 12%"

policyWith18 ="Policy With 18%"

'Required Input Columns

Dim pPolicyNumber, policyRef, version, pPolicyIssueDate, tDateDesc, pProductID, ptPartnerID, ptPartnerRegion, taxCode, btpSGSTAmt, btpUTGSTAmt, btpCGSTAmt, btpIGSTAmt, partnerGSTN, partnerStateCode, branchStateCode, netSurcharge, gstTax12Perc, gstTax18Perc, taxType, gstRegnFlag, gstENFlag As String

pPolicyNumber = "P\_POLICY\_NUMBER"

policyRef = "POLICY\_REF"

version = "VERSION"

pPolicyIssueDate = "P\_POLICY\_ISSUE\_DATE"

tDateDesc= "T\_DATE\_DESC"

pProductID = "P\_PRODUCT\_ID"

ptPartnerID = "PT\_PARTNER\_ID"

ptPartnerRegion = "PT\_PARTNER\_REGION"

taxCode = "TAX\_CODE"

btpSGSTAmt = "BTP\_SGST\_AMT"

btpUTGSTAmt = "BTP\_UTGST\_AMT"

btpCGSTAmt = "BTP\_CGST\_AMT"

btpIGSTAmt = "BTP\_IGST\_AMT"

partnerGSTN = "PARTNER\_GSTN"

partnerStateCode = "PARTNER\_STATE\_CODE"

branchStateCode = "BRANCH\_STATE\_CODE"

netSurcharge = "NETSURCHARGE"

gstTax12Perc = "GST\_TAX\_12\_PERC"

gstTax18Perc = "GST\_TAX\_18\_PERC"

taxType = "TAXTYPE"

gstRegnFlag = "GST\_REGN\_FLAG"

gstENFlag = "GST\_EN\_FLAG"

'Master Header Name

Dim masterStateName, masterPtPartnerRegion, masterStateCode, analysisCode, lookupCode, finalRemark As String

masterStateName = "State name"

masterPtPartnerRegion = "PT\_PARTNER\_REGION"

masterStateCode = "STATE CODE"

analysisCode = "Analysis Code"

lookupCode = "Lookup Code"

finalRemark = "Final Remark"

'Additional Column Names

Dim ExemptedBifurcation, revisedTaxType, revisedPartnerID, revisedPartnerStateCode, revisedPartnerStateName, exemptedTaxType, exempted, remark, addRemark, revisedGSTN, revisedGSTRegnFlag, wrongGST, remarkWorngGST, sezNonSez, tpRelatedRemark, tpProductCodeApplicable, tpPremiumTaxable12, OtherPremiumTaxable18, RevisedMaximusInvoicenumber, SAC, sacName, StatcodeAsperRevisedGSTN, StatcodeAsperGSTNRevisedPartnerIDStateCodeTF, Remarks As String

ExemptedBifurcation ="Exempted Bifurcation"

revisedTaxType = "Revised Tax Type"

revisedPartnerID = "Revised PARTNER\_ID"

revisedPartnerStateCode = "Revised PARTNER\_STATE\_CODE"

revisedPartnerStateName= "Revised PARTNER\_STATE\_NAME"

exemptedTaxType = "Exempted Tax Type"

exempted = "Exempted"

remark = "Remark"

addRemark = "Add Remark"

revisedGSTN = "Revised GSTN"

revisedGSTRegnFlag="Revised GST\_REGN\_FLAG"

wrongGST = "Wrong GST"

remarkWorngGST = "Remark Worng GSt"

sezNonSez = "SEZ/ Non Sez"

tpRelatedRemark = "Tp Related Remark"

tpProductCodeApplicable = "TP Product Code Applicable"

tpPremiumTaxable12 = "TP Premium Taxable @12%"

OtherPremiumTaxable18 = "Other Premium Taxable@18%"

RevisedMaximusInvoicenumber = "Revised Maximus Invoice number"

SAC = "SAC"

sacName= "SAC Name"

StatcodeAsperRevisedGSTN = "Statcode As per Revised GSTN"

StatcodeAsperGSTNRevisedPartnerIDStateCodeTF = "Statcode As per GSTN & Revised Partner ID State Code T /F"

Remarks = "Remarks"

io\_dt\_input.Columns.Add(ExemptedBifurcation)

io\_dt\_input.Columns.Add(revisedTaxType)

io\_dt\_input.Columns.Add(revisedPartnerID)

io\_dt\_input.Columns.Add(revisedPartnerStateCode)

io\_dt\_input.Columns.Add(revisedPartnerStateName)

io\_dt\_input.Columns.Add(exemptedTaxType)

io\_dt\_input.Columns.Add(exempted)

io\_dt\_input.Columns.Add(remark)

io\_dt\_input.Columns.Add(addRemark)

io\_dt\_input.Columns.Add(revisedGSTN)

io\_dt\_input.Columns.Add(revisedGSTRegnFlag)

io\_dt\_input.Columns.Add(wrongGST)

io\_dt\_input.Columns.Add(remarkWorngGST)

io\_dt\_input.Columns.Add(sezNonSez)

io\_dt\_input.Columns.Add(tpRelatedRemark)

io\_dt\_input.Columns.Add(tpProductCodeApplicable)

io\_dt\_input.Columns.Add(tpPremiumTaxable12)

io\_dt\_input.Columns.Add(OtherPremiumTaxable18)

io\_dt\_input.Columns.Add(RevisedMaximusInvoicenumber)

io\_dt\_input.Columns.Add(SAC)

io\_dt\_input.Columns.Add(sacName)

io\_dt\_input.Columns.Add(StatcodeAsperRevisedGSTN)

io\_dt\_input.Columns.Add(StatcodeAsperGSTNRevisedPartnerIDStateCodeTF)

io\_dt\_input.Columns.Add(Remarks)

Dim Count As Integer

Count = 1

For Each row In io\_dt\_input.Rows

Console.WriteLine("Row" + CStr(Count))

' Revised PARTNER\_ID column derivation

'(1)\*\*\*\*\*\*\*\*\*\*PT\_PARTNER\_ID Column Started

invalidColumns = ""

Try

If Not String.IsNullOrEmpty(row(ptPartnerID).tostring) Then

row(revisedPartnerID) = "'"+row(ptPartnerID).tostring

Else

row(revisedPartnerID) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," +revisedPartnerID+", " + errMsg.Message.ToString

End Try

'(1)\*\*\*\*\*\*\*\*\*\*PT\_PARTNER\_ID Column \_\_\_\_END

'Revised PARTNER\_STATE\_CODE column Derivation

'If "PARTNER\_STATE\_CODE" column is not empty, then the value need to be copied the same.

Try

If Not String.IsNullOrEmpty(row(partnerStateCode).ToString) Then

row(revisedPartnerStateCode)= row(partnerStateCode).tostring

Else

'If "TAXTYPE" column is not empty, then the value need to be copied the same.

If row(taxType).tostring = "SGST+CGST" Then

row(revisedPartnerStateCode) = in\_currentStateCode

Else

'If "TAX\_CODE" column is not empty, then the value need to be copied the same.

If Not String.IsNullOrEmpty(row(taxCode).tostring) And row(taxCode).tostring.Length > 3 Then

row(revisedPartnerStateCode) = row(taxCode).tostring.Substring(0,2)

Else

''If "PT\_PARTNER\_REGION" column is not empty, then the value need to be copied the same.

If Not String.IsNullOrEmpty(row(ptPartnerRegion).tostring) Then

' "PT\_PARTNER\_REGION" column value need to verify with BRANCH master. If matched then extract state code column value from BRANCH MASTER.

If in\_dt\_BranchMasterStateCode.AsEnumerable().Where(Function(DR) DR(masterStateName).ToString.Trim.ToUpper.Equals(row(masterPtPartnerRegion).ToString.ToUpper.Trim)).Count > 0 Then

row(revisedPartnerStateCode) = in\_dt\_BranchMasterStateCode.AsEnumerable().Where(Function(DR) DR(masterStateName).ToString.Trim.ToUpper.Equals(row(masterPtPartnerRegion).ToString.ToUpper.Trim)).Select(Function(DR1) DR1(masterStateCode).ToString).ToList(0).ToString

Else

'Have to verify current state is maharastra then YES value is "27" else "24"

If in\_currentStateName.ToUpper.Trim.Contains("MAH") Then

row(revisedPartnerStateCode) = "27"

Else

row(revisedPartnerStateCode) = "24"

End If

End If

Else

'Have to verify current state is maharastra then YES value is "27" else "24"

If in\_currentStateName.ToUpper.Trim.Contains("MAH") Then

row(revisedPartnerStateCode) = "27"

Else

row(revisedPartnerStateCode) = "24"

End If

End If

End If

End If

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + revisedPartnerStateCode+", " + errMsg.Message.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(2)Revised PARTNER\_STATE\_CODE Column Done

Try

If IsNumeric(row(revisedPartnerStateCode).tostring) Then

If in\_dt\_BranchMasterStateCode.AsEnumerable().Where(Function(DR) CInt( If (IsNumeric(DR(masterStateCode).ToString),DR(masterStateCode).ToString,"0")) = Convert.ToInt16(row(revisedPartnerStateCode).ToString)).Count >= 1 Then

row(revisedPartnerStateName) = in\_dt\_BranchMasterStateCode.AsEnumerable().Where(Function(DR) CInt( If (IsNumeric(DR(masterStateCode).ToString),DR(masterStateCode).ToString,"0")) = CInt(row(revisedPartnerStateCode).ToString)).Select(Function(DR1) DR1(masterStateName)).ToList(0).ToString

Else

row(revisedPartnerStateName) = ""

End If

Else

row(revisedPartnerStateName) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns + ","+ revisedPartnerStateName + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(3) Revised PARTNER STATE Name Column Done

'\*\*\*\*\*\*\*\*\*\*(4) Exempted Tax Type Name Column Started

Try

If row(taxType).ToString.ToUpper.Trim.Equals(exempted.ToUpper.Trim) Then

If IsNumeric(row(partnerStateCode).ToString) And IsNumeric(row(branchStateCode).ToString) Then

If CInt(row(partnerStateCode).ToString) = CInt(row(branchStateCode).ToString) Then

row(exemptedTaxType) = "CGST+SGST"

Else

row(exempted) = "IGST"

End If

Else

row(exemptedTaxType) = ""

End If

Else

row(exemptedTaxType) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns + ","+ exemptedTaxType + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(4) Exempted Tax Type Name Column Ended

'\*\*\*\*\*\*\*\*\*\*(5) Exempted Column Started

Try

If Not String.IsNullOrEmpty(row(pProductID).tostring) Then

If in\_dt\_exemptedTPproductCode.AsEnumerable().Where(Function(DR) DR(analysisCode).ToString.Trim.Equals(row(pProductID).tostring)).Count > 0 Then

row(exempted) = in\_dt\_exemptedTPproductCode.AsEnumerable().Where(Function(DR) DR(analysisCode).ToString.Trim.Equals(row(pProductID).tostring)).Select(Function(DR1) DR1(lookupCode).ToString).ToList(0).ToString

Else

row(exempted) = ""

End If

Else

row(exempted) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + exempted + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(5) Exempted Column End

'\*\*\*\*\*\*\*\*\*\*(6) Reamrk Column Start

Try

If dt\_referenceTable.AsEnumerable().Where(Function(DR) DR(pPolicyNumber).ToString.ToUpper.Trim.Equals(row(pPolicyNumber).ToString.ToUpper.Trim)).Count >= 2 Then

If String.Join(",",dt\_referenceTable.AsEnumerable().Where(Function(DR) DR(pPolicyNumber).ToString.ToUpper.Trim.Equals(row(pPolicyNumber).ToString.ToUpper.Trim)).Select(Function(DR1) DR1(version).ToString).ToList()).ToUpper.Contains("ER") Then

Dim policyIssueDate, transactionDate As String

policyIssueDate = row(pPolicyIssueDate).tostring

transactionDate = row(tDateDesc).tostring

If IsDate(policyIssueDate) And IsDate(transactionDate) Then

If CInt(Date.ParseExact(policyIssueDate,"dd-MM-yyyy",System.Globalization.CultureInfo.InvariantCulture).ToString("MM")) = CInt(Date.ParseExact(transactionDate,"dd-MM-yyyy",System.Globalization.CultureInfo.InvariantCulture).ToString("MM")) Then

row(remark) = cancellationWithinSameMonth

Else

If row(version).tostring.ToUpper.Trim.Contains("ER")

If CInt(Date.ParseExact(policyIssueDate,"dd-MM-yyyy",System.Globalization.CultureInfo.InvariantCulture).ToString("yyyy")) >= 2017 Then

row(remark) = cancelAfterGST

Else

row(remark) = cancelBeforeGST

End If

Else

row(remark) = row(taxType).tostring

End If

End If

Else

row(remark) = row(taxType).tostring

End If

Else

row(remark) = row(taxType).tostring

End If

Else

If dt\_referenceTable.AsEnumerable().Where(Function(DR) DR(pPolicyNumber).ToString.ToUpper.Trim.Equals(row(pPolicyNumber).ToString.ToUpper.Trim)).Count = 1 Then

If row(version).tostring.ToUpper.Trim.Contains("ER") Then

If CInt(Date.ParseExact(row(pPolicyIssueDate).tostring,"dd-MM-yyyy",System.Globalization.CultureInfo.InvariantCulture).ToString("yyyy")) >= 2017 Then

row(remark) = cancelAfterGST

Else

row(remark) = cancelBeforeGST

End If

Else

row(remark) = row(taxType).tostring

End If

Else

row(remark) = row(taxType).tostring

End If

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + remark + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(6) Reamrk Column End

'\*\*\*\*\*\*\*\*\*\*(7) Add Reamrk Column Start

Try

If row(remark).tostring.ToString.ToUpper.Trim.Equals(cancellationWithinSameMonth) Or row(remark).tostring.ToString.ToUpper.Trim.Equals(NPzero) Then

row(addRemark) = "Not Consider"

Else

row(addRemark) = "Consider"

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +","+ addRemark + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(7) Add Reamrk Column End

'\*\*\*\*\*\*\*\*\*\*(8) Revised GSTN Column Start

Try

row(revisedGSTN) = row(gstRegnFlag).tostring

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + revisedGSTN + ", "+ errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(8) Revised GSTN Column End

'\*\*\*\*\*\*\*\*\*\*(9) Revised GST Regn Flag Column Start

Try

If row(revisedGSTN).ToString.ToUpper.Trim.Equals(reg.ToUpper.Trim) Then

If row(version).ToString.ToUpper.Trim.Contains("EN") Then

row(revisedGSTN) = regEn

row(revisedGSTRegnFlag) = row(partnerGSTN).Tostring

Else

row(revisedGSTRegnFlag) = row(partnerGSTN).Tostring

End If

Else

If row(revisedGSTN).ToString.ToUpper.Trim.Equals(UnReg.ToUpper.Trim) Then

row(revisedGSTRegnFlag) = ""

End If

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + revisedGSTRegnFlag + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(9) Revised GST Regn Flag Column End

'\*\*\*\*\*\*\*\*\*\*(10) Wrong GST and (11) Remark Worng GSt Column Start

Try

If row(revisedGSTN).ToString.ToUpper.Trim.Equals(reg.ToUpper.Trim) Or row(revisedGSTN).ToString.ToUpper.Trim.Equals(regEn.ToUpper.Trim) Then

If in\_dt\_partnerInvalidGSTN.AsEnumerable().Where(Function(DR) DR(partnerGSTN).ToString.Trim.Equals(row(revisedGSTRegnFlag).ToString)).Count > 0 Then

row(wrongGST) = in\_dt\_partnerInvalidGSTN.AsEnumerable().Where(Function(DR) DR(partnerGSTN).ToString.Trim.Equals(row(revisedGSTRegnFlag).ToString)).Select(Function(DR1) DR1(remark).ToString).ToList(0).ToString

row(remarkWorngGST) = in\_dt\_partnerInvalidGSTN.AsEnumerable().Where(Function(DR) DR(partnerGSTN).ToString.Trim.Equals(row(revisedGSTRegnFlag).ToString)).Select(Function(DR1) DR1(finalRemark).ToString).ToList(0).ToString

Else

row(revisedGSTN) = UnReg

End If

Else

row(wrongGST) = ""

row(remarkWorngGST) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + wrongGST+","+ remarkWorngGST + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(10) Wrong GST and (11) Remark Worng GST Column End

'\*\*\*\*\*\*\*\*\*\*(12) Column Start "SEZ/ Non Sez"

Try

If row(revisedGSTN).ToString.ToUpper.Trim.Equals(reg.ToUpper.Trim) Or row(revisedGSTN).ToString.ToUpper.Trim.Equals(regEn.ToUpper.Trim) Then

If in\_dt\_sezNonSez.AsEnumerable().Where(Function(DR) DR(partnerGSTN).ToString.Trim.Equals(row(revisedGSTRegnFlag).ToString.Trim)).Count > 0 Then

row.Item(sezNonSez) = in\_dt\_sezNonSez.AsEnumerable().Where(Function(DR) DR(partnerGSTN).ToString.Trim.Equals(row(revisedGSTRegnFlag).ToString.Trim)).Select(Function(DR1) DR1(finalRemark).ToString).ToList(0).ToString

If row(taxType).ToString.ToUpper.Trim.Equals("SGST+CGST") Then

row(revisedGSTN) = UnReg

End If

Else

row.Item(sezNonSez) = ""

End If

Else

row.Item(sezNonSez) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + sezNonSez + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(12) Column END "SEZ/ Non Sez"

'\*\*\*\*\*\*\*\*\*\*(13) Column Start "Tp Related Remark"

Try

If in\_dt\_ExemptedTPBasicTP.AsEnumerable().Where(Function(DR) DR("Product Code").ToString.Equals(row(pProductID).ToString)).Count > 0

row(tpRelatedRemark) = row(pProductID).ToString

Else

row(tpRelatedRemark) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +","+ tpRelatedRemark + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(13) Column End "Tp Related Remark"

'\*\*\*\*\*\*\*\*\*\*(14) Column Start "TP Premium Taxable @12%"

Try

Dim CGSTAmt, SGSTAmt, UTGSTAmt, IGSTAmt As Integer

If IsNumeric(row(btpCGSTAmt).tostring) Then

CGSTAmt = CInt(row(btpCGSTAmt))

Else

CGSTAmt = 0

End If

If IsNumeric(row(btpSGSTAmt).tostring) Then

SGSTAmt = CInt(row(btpSGSTAmt))

Else

SGSTAmt = 0

End If

If IsNumeric(row(btpUTGSTAmt).tostring) Then

UTGSTAmt = CInt(row(btpUTGSTAmt))

Else

UTGSTAmt = 0

End If

If IsNumeric(row(btpIGSTAmt).tostring) Then

IGSTAmt = CInt(row(btpIGSTAmt))

Else

IGSTAmt = 0

End If

'Dim tempTP As String

'((btpCGSTAmt+btpSGSTAmt+btpUTGSTAmt+btpIGSTAmt) \* 100/12).ToString

row(tpPremiumTaxable12) = CStr(Math.Round(Convert.ToDouble((CGSTAmt + SGSTAmt + UTGSTAmt + IGSTAmt )\*100/12)))

'row(tpPremiumTaxable12) = CStr(Math.Round((btpCGSTAmt+btpSGSTAmt+btpIGSTAmt+btpUTGSTAmt)\* 100/12 ))

'row(tpPremiumTaxable12) = CStr(Math.Round(CDbl(Convert.ToDouble(btpCGSTAmt)+Convert.ToDouble(btpSGSTAmt)+Convert.ToDouble(btpIGSTAmt)+Convert.ToDouble(btpUTGSTAmt) \* 100/12)))

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + tpPremiumTaxable12 + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(14) Column End "TP Premium Taxable @12%"

'\*\*\*\*\*\*\*\*\*\*(15) Column Start "Other Premium Taxable@18%"

Try

If IsNumeric(row("NET\_SURCHARGE"))

row(OtherPremiumTaxable18)= CInt(row("NET\_SURCHARGE")) - CInt(row(tpPremiumTaxable12))

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + OtherPremiumTaxable18 + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(15) Column End "Other Premium Taxable@18%"

'\*\*\*\*\*\*\*\*\*\*(16) Column Start "TP Product Code Applicable"

Try

If CInt(row(tpPremiumTaxable12).tostring) > 0 And CInt(row(OtherPremiumTaxable18).tostring) > 0 Then

row(tpProductCodeApplicable) = policyWith18And12

Else

If CInt(row(tpPremiumTaxable12).tostring) > 0 And CInt(row(OtherPremiumTaxable18).tostring) = 0 Then

row(tpProductCodeApplicable) = policyWith12

Else

If CInt(row(tpPremiumTaxable12).tostring) = 0 And CInt(row(OtherPremiumTaxable18).tostring) > 0 Then

row(tpProductCodeApplicable) = policyWith18

Else

If CInt(row(tpPremiumTaxable12).tostring) = 0 And CInt(row(OtherPremiumTaxable18).tostring) = 0 Then

row(tpProductCodeApplicable) = policyWith18

End If

End If

End If

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + tpProductCodeApplicable + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(16) Column End "TP Product Code Applicable"

'\*\*\*\*\*\*\*\*\*\*(17) Column Start "SAC"

Try

If in\_dt\_mappingMaster.AsEnumerable().Where(Function(DR) DR("Product Code").ToString.Trim.Equals(row(pProductID).ToString.Trim)).Count > 0 Then

row(SAC) = in\_dt\_mappingMaster.AsEnumerable().Where(Function(DR) DR("Product Code").ToString.Trim.Equals(row(pProductID).ToString.Trim)).Select(Function (DR1) DR1("SAC").ToString).ToList(0).ToString

Else

row(SAC) =""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + SAC + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(17) Column End "SAC"

'\*\*\*\*\*\*\*\*\*\*(18) Column Start "SAC Name"

Try

If in\_dt\_mappingMaster.AsEnumerable().Where(Function(DR) DR("Product Code").ToString.Trim.Equals(row(pProductID).ToString.Trim)).Count > 0 Then

row(sacName) = in\_dt\_mappingMaster.AsEnumerable().Where(Function(DR) DR("Product Code").ToString.Trim.Equals(row(pProductID).ToString.Trim)).Select(Function (DR1) DR1("Insurance and pension services (excluding reinsurance services)").ToString).ToList(0).ToString

Else

row(sacName) =""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + sacName + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(18) Column End "SAC Name"

'\*\*\*\*\*\*\*\*\*\*(19) Column Start "Revised GST Regn Flag"

Try

If Not String.IsNullOrEmpty(row(revisedGSTRegnFlag).tostring) Then

row(StatcodeAsperRevisedGSTN) = row(revisedGSTRegnFlag).tostring.substring(0,2)

Else

row(StatcodeAsperRevisedGSTN) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + StatcodeAsperRevisedGSTN + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(19) Column End "Revised GST Regn Flag"

'\*\*\*\*\*\*\*\*\*\*(20) Column Start "Statcode As per GSTN & Revised Partner ID State Code T /F"

Try

If IsNumeric(row(revisedPartnerStateCode).tostring) And IsNumeric(row(StatcodeAsperRevisedGSTN).tostring) Then

If CInt(row(revisedPartnerStateCode).tostring) = CInt(row(StatcodeAsperRevisedGSTN).tostring) Then

row(StatcodeAsperGSTNRevisedPartnerIDStateCodeTF) = "TRUE"

Else

row(StatcodeAsperGSTNRevisedPartnerIDStateCodeTF) = "FALSE"

row(revisedGSTN) = UnReg

End If

Else

row(StatcodeAsperGSTNRevisedPartnerIDStateCodeTF) = ""

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + StatcodeAsperGSTNRevisedPartnerIDStateCodeTF + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(20) Column End "Statcode As per GSTN & Revised Partner ID State Code T /F"

'\*\*\*\*\*\*\*\*\*\*(21) Column Start "Exempted Bifurcation Value Extraction"

Try

If row(taxType).tostring.ToUpper.Trim.Equals(exempted.ToUpper.Trim) Then

If row("Leader").tostring.ToUpper.Trim.Equals("NL") Then

row(ExemptedBifurcation) = "Co-Insurance inwards"

Else

If Not String.IsNullOrEmpty(row(exempted).tostring) Then

row(ExemptedBifurcation) = "Product Exempt"

Else

If row(sezNonSez).tostring.ToUpper.Trim.Equals("SEZ") Then

row(ExemptedBifurcation) = "SEZ"

Else

If IsNumeric(row("NET\_PREMIUM").tostring) Then

If CInt(row("NET\_PREMIUM").tostring) > -5 And CInt(row("NET\_PREMIUM").tostring) < 5 Then

row(ExemptedBifurcation) = "Premium less than Rs.5"

Else

If row("CN\_TAXTYPE").tostring.ToUpper.Trim.Equals("CN W/O GST") Then

row(ExemptedBifurcation) = row("CN\_TAXTYPE").tostring

Else

If row(remark).tostring.ToUpper.Trim.Equals(cancellationWithinSameMonth) Then

row(ExemptedBifurcation) = cancellationWithinSameMonth

Else

row(ExemptedBifurcation) = "Refer IT"

End If

End If

End If

Else

If row("CN\_TAXTYPE").tostring.ToUpper.Trim.Equals("CN W/O GST") Then

row(ExemptedBifurcation) = row("CN\_TAXTYPE").tostring

Else

If row(remark).tostring.ToUpper.Trim.Equals(cancellationWithinSameMonth) Then

row(ExemptedBifurcation) = cancellationWithinSameMonth

Else

row(ExemptedBifurcation) = "Refer IT"

End If

End If

End If

End If

End If

End If

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + ExemptedBifurcation + ", "+errMsg.ToString

End Try

'\*\*\*\*\*\*\*\*\*\*(21) Column End "Exempted Bifurcation Value Extraction"

'\*\*\*\*\*\*\*\*\*\*(22) Column Start "Revised Tax Type"

Try

If row(taxType).tostring.ToUpper.Trim.Equals(exempted.tostring.ToUpper.Trim) Then

If row(ExemptedBifurcation).tostring.ToUpper.Trim.Equals("CN W/O GST") Then

If CInt(row(revisedPartnerStateCode).tostring) = CInt(in\_currentStateCode) Then

row(revisedTaxType) = "CN W/O GST-CGST+SGST"

Else

row(revisedTaxType) = "CN W/O GST-IGST"

End If

Else

row(revisedTaxType) = exempted

End If

Else

row(revisedTaxType) = row(taxType).tostring

End If

Catch errMsg As Exception

invalidColumns = invalidColumns +"," + revisedTaxType + ", "+errMsg.ToString

End Try

row(Remarks) = invalidColumns

Count = Count +1

Next row