In Excel, I want to do an update process of one table based on the values of another table. I do not want to use XLUTHER or any other method that uses copy and paste, as it would require manual filtering. I am open to using VBA to do this process.

To modify your `GeneralizedUpdate` VBA code so that it searches for the target tables across **all worksheets** in the workbook (instead of just on "Sheet1"), you need to loop through all the sheets in the workbook and look for the target tables on each one.

Here's how to adjust your code:

Updated VBA Code:

vba

```vba
Sub GeneralizedUpdate()
    Dim ws As Worksheet
    Dim tbl2 As ListObject
    Dim row2 As ListRow
    Dim targetTable As ListObject
    Dim IDcol As Range
    Dim matchID As Range
    Dim targetColumn As Range
    Dim targetColumnName As String
    Dim targetTableName As String
    Dim ID As Variant
    Dim newValue As Variant
    Dim foundTable As Boolean

    ' Define the worksheet and Table2 (the mapping table on Sheet1)
    Set tbl2 = ThisWorkbook.Sheets("Sheet1").ListObjects("Table2") ' Adjust to where Table2 is located

    ' Loop through each row in Table2
    For Each row2 In tbl2.ListRows
        ' Extract the necessary values from Table2
        targetTableName = row2.Range(1, 2).Value   ' Get the table name from Table2 (TargetTable)
        targetColumnName = row2.Range(1, 3).Value  ' Get the column name from Table2 (TargetColumn)
        ID = row2.Range(1, 1).Value               ' Get the ID for lookup
        newValue = row2.Range(1, 4).Value         ' Get the new value to update

        foundTable = False ' Reset the flag for finding the table

        ' Loop through all sheets in the workbook to find the target table
        For Each ws In ThisWorkbook.Sheets
            On Error Resume Next
            Set targetTable = ws.ListObjects(targetTableName)
            On Error GoTo 0

            If Not targetTable Is Nothing Then
                foundTable = True ' Table found in this worksheet

                ' Find the ID column in the target table
                Set IDcol = targetTable.ListColumns("ID").DataBodyRange

                ' Find the matching row in the target table by ID
                Set matchID = IDcol.Find(What:=ID, LookAt:=xlWhole)

                If Not matchID Is Nothing Then
                    ' Get the column to update in the target table
                    On Error Resume Next
                    Set targetColumn = targetTable.ListColumns(targetColumnName).DataBodyRange
                    On Error GoTo 0

                    If Not targetColumn Is Nothing Then
                        ' Find the correct row to update based on the matching ID
                        Dim rowNumber As Long
                        rowNumber = matchID.Row - targetTable.HeaderRowRange.Row

                        ' Update the cell in the target column
                        targetTable.DataBodyRange(rowNumber, targetTable.ListColumns(targetColumnName).Index).Value = newValue
                    Else
                        MsgBox "Target column " & targetColumnName & " not found in table " & targetTableName & " on sheet " & w
                    End If
                Else
                    MsgBox "ID " & ID & " not found in table " & targetTableName & " on sheet " & ws.Name
                End If

                ' Exit the sheet loop once the target table is found and processed
                Exit For
            End If
        Next ws

        If Not foundTable Then
            MsgBox "Table " & targetTableName & " not found in any sheet."
        End If
    Next row2
    MsgBox "Generalized update complete"
End Sub
```

Key Changes:

1. **Loop through all worksheets**:
    2. The code now loops through every worksheet in `ThisWorkbook.Sheets` to search for the target table.
3. **Check each worksheet for the target table**:
    4. For each worksheet, the code tries to locate the table by its name using `ws.ListObjects(targetTableName)`.
5. **Flagging when the table is found**:
    6. A `foundTable` flag is used to determine if the table was located across any of the worksheets. If the table isn't found in any sheet, the macro will notify you.
7. **Exit the loop once the target table is found**:
    8. Once the correct table is found in any worksheet, the loop breaks to avoid unnecessary checks on the remaining worksheets.
9. **Error handling**:
    10. The `On Error Resume Next` and `On Error GoTo 0` structure handles cases where a table might not be found or other lookup errors occur, with proper messages to notify the user.

Example Workflow:

If **Table2** references tables on different worksheets, such as **Table1** in "Sheet2" and **Table3** in "Sheet3", the code will dynamically search across all sheets and update the relevant rows and columns accordingly.

This will make the code more flexible and adaptable for workbooks with tables spread across multiple sheets. Let me know if you need further clarifications or adjustments!

From ChatGPT, designed with FancyGPT