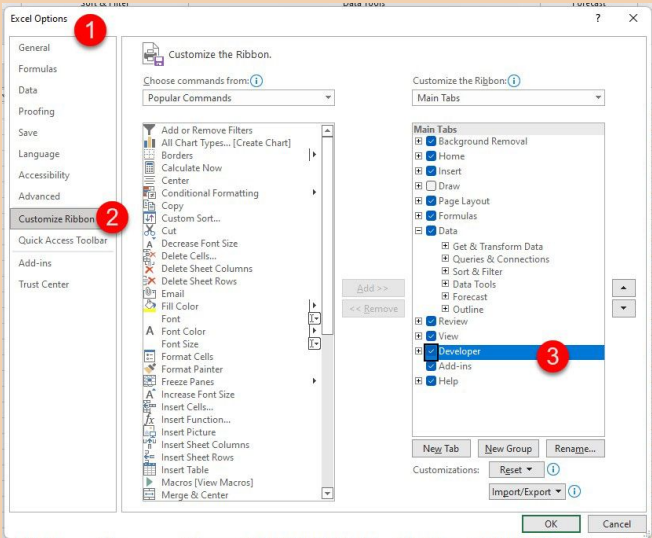


Programatic Manipulation of Excel Spreadsheets prior to Orixa Importation

It is often necessary to import data into Orixa from Excel. The following short document runs through the steps needed to automate the deletion of multiple columns in an Excel sheet. This is particularly important if a work process involves large numbers of Excel Sheets with large numbers of columns to delete.

Steps to Run VBA script in an Excel Spreadsheet



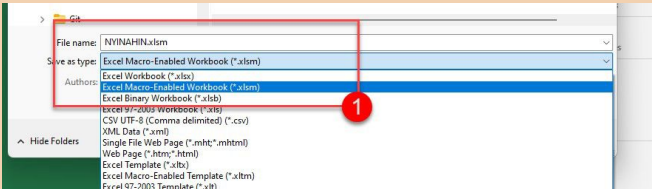
Excel VBA Options "Developer" Tab

You will need to access the "Developer" tab of your Excel Tool-ribbon. If this is not visible, take the following steps:

By default Excel does not show the developer-tools needed to run VBA scripts. To fix this:

1. Go to Excel Options
2. Find "Customize Ribbon".
3. Tick the "Developer" Tab.

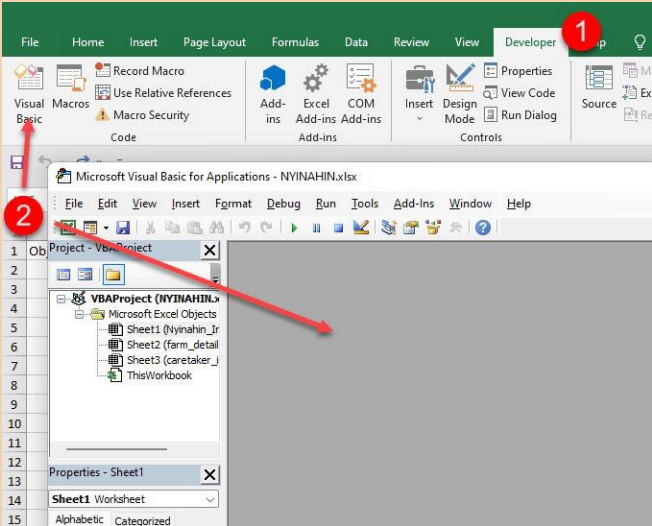
Note that versions of excel differ. If your version of Excel does not show options in this format, please check how to enable Developer Options for your version of Excel.



Excel VBA "Save as Macro Enabled"

In some versions of Excel, the standard Excel Sheet format does not support running macros and VBA scripts. Try to run the scripts as shown. If this fails, take the following step:

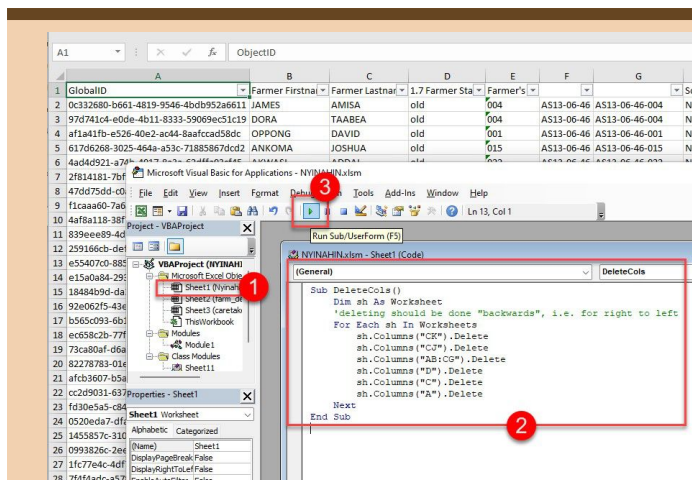
To enable this, first **save as** and pick the "macro-enabled" option (*.xlsm)



Developer Tab Show VBA IDE

Once Developer Options are set to be visible, and the Excel sheet is saved in XLSM format, if that is needed, take the following steps:

1. Find the Developer Tab.
2. Click on the "Visual Basic" button. The Microsoft VBA application will open.



Running a VBA Macro in Excel

Now create and run a VBA script

1. Double-click on a "sheet" in the VBA Project tab.
2. Write a VBA script, or paste text from a previously saved script file.
3. Press the green "run" button to actually run a script.

Example scripts to delete columns and rows from a sheet, and to call "Save As"

The following script can be pasted into the Excel VBA code-viewer, and run to remove columns from a sheet. Once run all columnsshift so the code should never be run twice. Changes are only saved when a spreadsheet is saved, so it is possible to undo errors.

```
Sub DeleteCols()

Dim sh As Worksheet
'deleting should be done "backwards", i.e. for right to left
For Each sh In Worksheets
    sh.Columns("AZ").Delete
    sh.Columns("AM").Delete
    sh.Columns("H:Z").Delete
    sh.Columns("G").Delete
    sh.Columns("D").Delete
    sh.Columns("B").Delete
Next
End Sub
```

The following script can be used to remove the top **row** from an Excel file, and then save it in CSV format. This is useful as a preparation process prior to data-importation in Orix.

```
Sub DeleteColsAndSaveAsCSV()
    Dim sh As Worksheet
    Set wrk = ActiveWorkbook.ActiveSheet
    wrk.Rows("1:10").Delete
    ActiveWorkbook.SaveAs Filename:="C:\FileName-Inspection.csv", FileFormat:=xlCSV
End Sub
```

NOTES ON VBA SCRIPTS

- Many of the formulae and forms of written syntax used in Excel carry across into VBA.
- For example the use of "H:Z" in the above example to signify a range of columns.
- There are very large numbers of VBA scripts available via the internet.
- The complex syntax of Visual Basic is beyond the scope of this short document, but note "sh as Worksheet" creates a variable in the script which can be referred to, and the "sh" variable has "properties" such as "Columns", "Rows" etc., which can be targetted in the script.
- It can be very easy to run VBA scripts which damage a spreadsheet. Always take care to test scripts on a non-critical spreadsheet.

