

How to Create a Custom Ribbon Bar in Microsoft Excel

Purpose and Objectives

This document details the steps to take to create a Custom Ribbon Bar in Microsoft Excel.

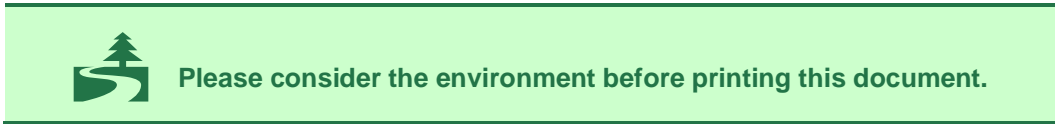







Table of Contents

- 1. Overview of the Office Fluent ribbon2**
 - 1.1 [MS-CUSTOMUI]: Custom UI XML Markup Specification2
- 2. Excel Open XML File Format2**
- 3. imageMSO (image Microsoft Office)2**
 - 3.1 imageMSO Browser Add-in2
- 4. Creating a Custom Ribbon3**
 - 4.1 Notepad ++.....3
 - 4.2 custom_ribbon.xlsm.....3
 - 4.3 Change Extension of custom_ribbon.xlsm to .zip3
 - 4.4 Edit the .rels (Relationships) XML File in Notepad++3
 - 4.5 Create the customUI Folder  and Sub Folders _rels and Images5
 - 4.5.1  customUI5
 - 4.5.2  _rels and  images5
 - 4.6 Copy Required Custom Icons / Images to Folder  images.....5
 - 4.7 Create the Custom .rels (Relationships) XML File6
 - 4.8 Create the Custom Ribbon XML File6
 - 4.9 Rename custom_ribbon.zip to custom_ribbon.xlsm7
 - 4.10 Open custom_ribbon.xlsm7
 - 4.10.1 Custom Ribbon.....7
 - 4.11 Add the VBA Macros / Code.....8
- 5. Trouble Shooting.....9**
- 6. Acronyms and Abbreviations.....9**
- 7. References9**
- 8. Revision Log9**

1. Overview of the Office Fluent ribbon

<https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/overview-of-the-office-fluent-ribbon>

1.1 [MS-CUSTOMUI]: Custom UI XML Markup Specification

https://docs.microsoft.com/en-us/openspecs/office_standards/ms-customui/31f152d6-2a5d-4b50-a867-9dbc6d01aa43

2. Excel Open XML File Format

Starting with the 2007 Microsoft Excel uses the XML-based file formats, such as .xlsx. Excel workbooks are really just a collection of XML files, all zipped together. To create a custom ribbon, we edit some of the XML files and also add to them.

<https://support.microsoft.com/en-us/office/open-xml-formats-and-file-name-extensions-5200d93c-3449-4380-8e11-31ef14555b18>

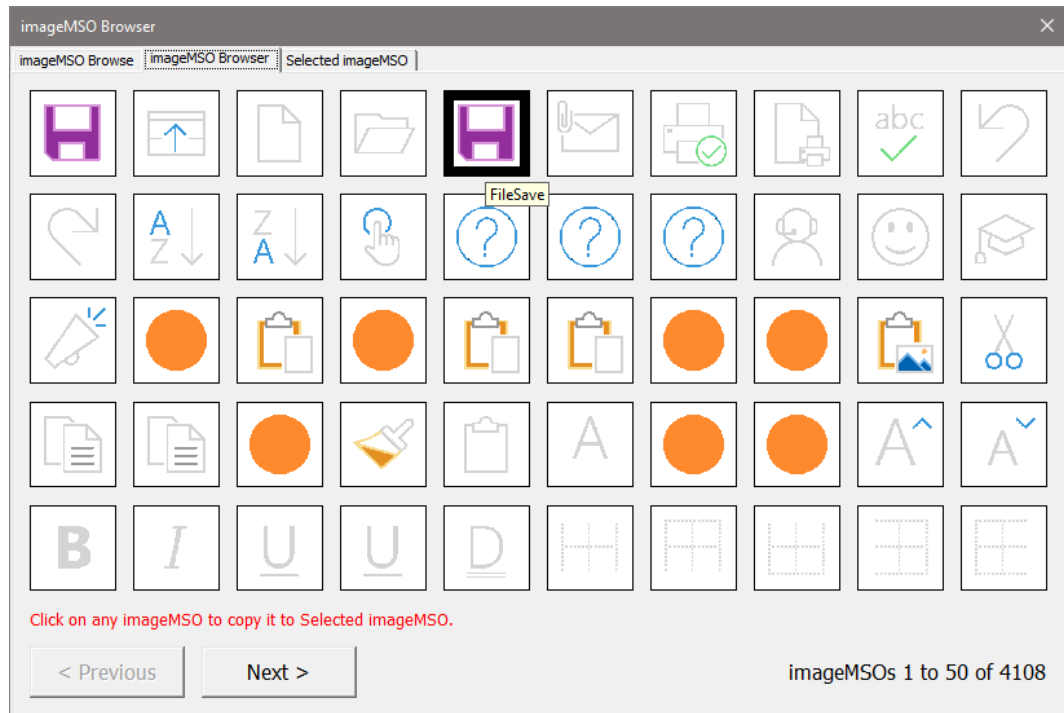
3. imageMSO (image Microsoft Office)

imageMSO's are built in icons used in the ribbon. They can be browsed using a suitable ImageMSO browser add-in.

3.1 imageMSO Browser Add-in

[Excel Coding - imageMSO Browser](#)

The imageMSO icon FileSave is currently highlighted.



4. Creating a Custom Ribbon Bar

4.1 Notepad ++

It is recommended to use Notepad++ to edit and create the required XML files.

Notepad++ is a free (as in “free speech” and also as in “free beer”) source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GNU General Public License.

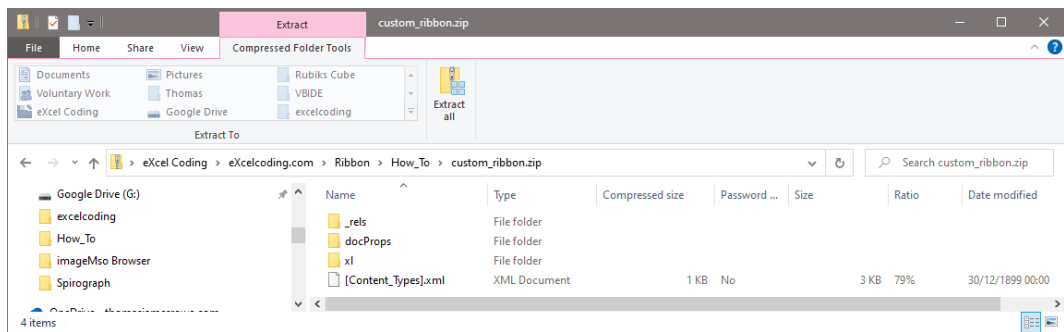
<https://notepad-plus-plus.org/>

4.2 custom_ribbon.xlsm

custom_ribbon.xlsm is the accompanying workbook for which we will be creating the custom ribbon.

4.3 Change Extension of custom_ribbon.xlsm to .zip

Change the extension of custom_ribbon.xlsm to .zip and open the file.



You should see three folders...

-rels

docProps

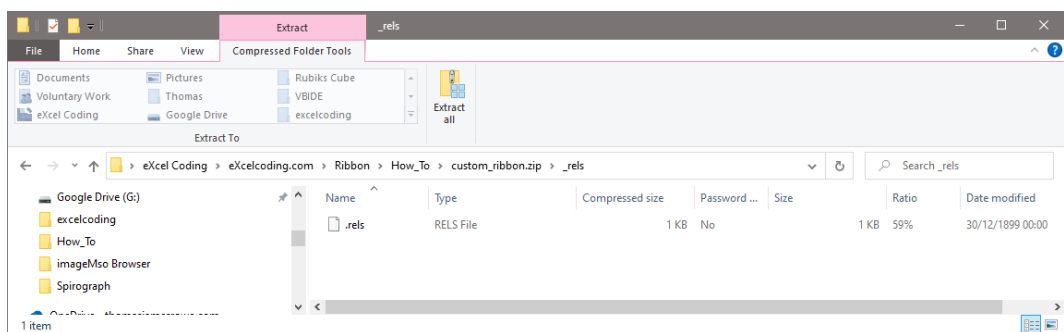
xl

And one XML Document File...

[Content_Types].xml

4.4 Edit the .rels (Relationships) XML File in Notepad++


The .rels file is found in folder _rels.



The default file looks like the following...

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship
Id="rId3"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties" Target="docProps/app.xml"/><Relationship Id="rId2"
Type="http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties" Target="docProps/core.xml"/><Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument" Target="xl/workbook.xml"/></Relationships>
```

The following line needs to be added to point to your custom ribbon .xml file in folder  customUI i.e. customUI/custom_ribbon.xml ...

```
<!-- Change the customUI/file_name.xml to suit your application -->

<Relationship Id="customUIXML"
Type="http://schemas.microsoft.com/office/2007/relationships/ui/extensibility"
Target="customUI/custom_ribbon.xml"/>
```

So that the file looks like this...

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">

<!-- Change the customUI/file_name.xml to suit your application -->

<Relationship Id="customUIXML"
Type="http://schemas.microsoft.com/office/2007/relationships/ui/extensibility"
Target="customUI/custom_ribbon.xml"/>

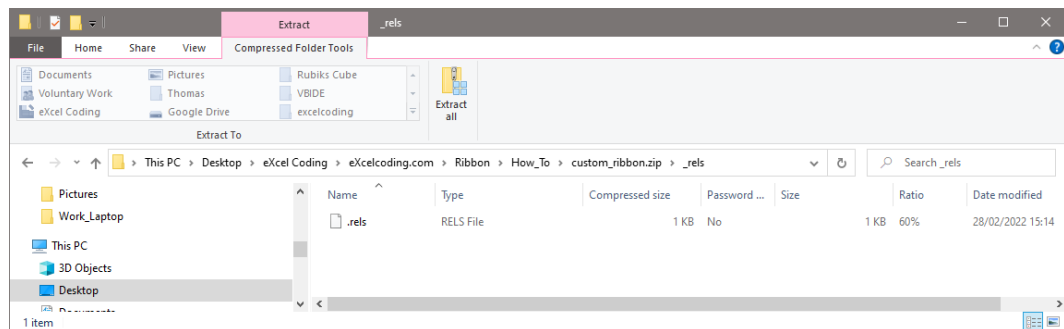
<Relationship Id="rId3"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties" Target="docProps/app.xml"/>

<Relationship Id="rId2"
Type="http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties" Target="docProps/core.xml"/>


<Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument" Target="xl/workbook.xml"/>

</Relationships>
```

Once the file has been edited replace the default .rels file with the edited .rels file.

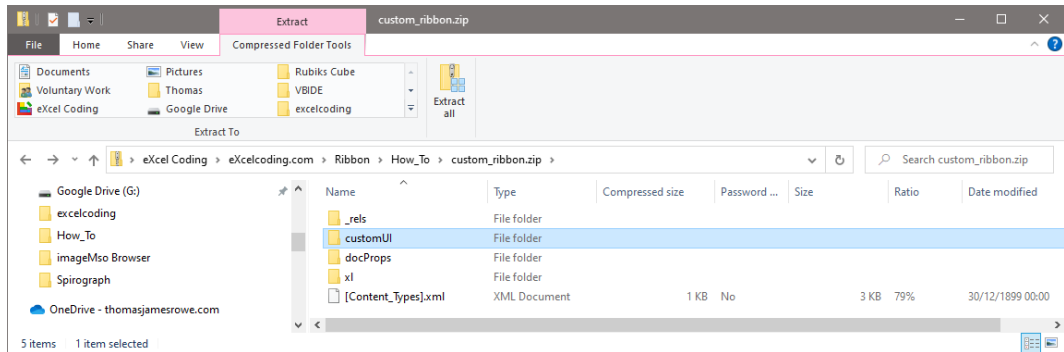


4.5 Create the customUI Folder and Sub Folders `_rels` and `Images`

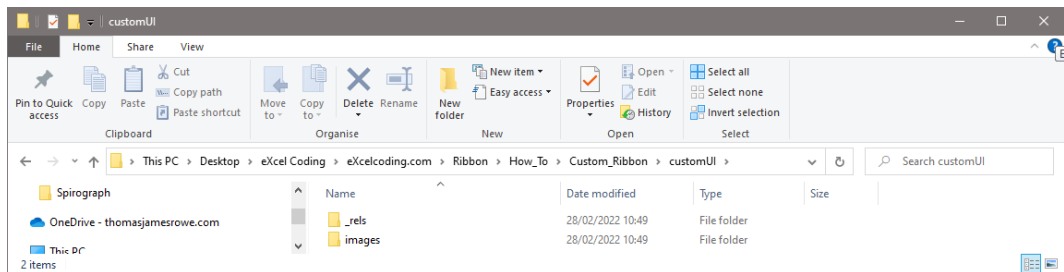
Create the customUI folder  and Sub Folders `_rels` and `Images`. This is where the XML file for the custom ribbon, the custom `.rels` (Relationships) file and any custom images will be stored.

Note: Create the folders on your C:/Drive and then copy to `custom_ribbon.zip`.


4.5.1 customUI

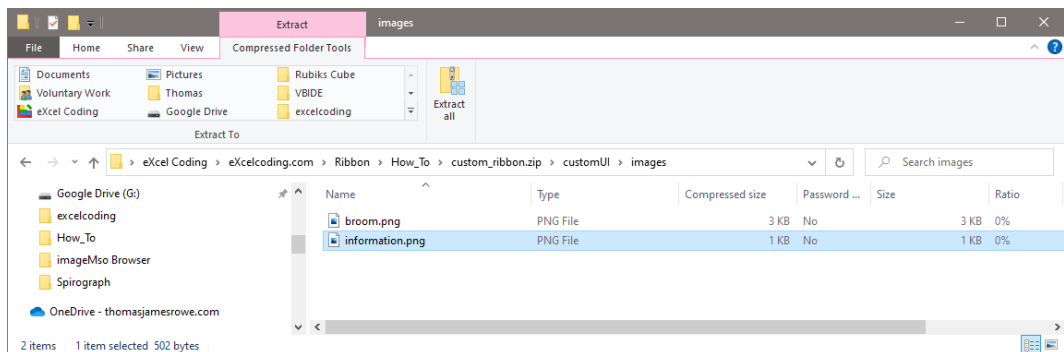


4.5.2 `_rels` and `images`



4.6 Copy Required Custom Icons / Images to Folder `images`

Copy required custom icons / images to folder  `images`, below we have copied images `broom.png` and `information.png`.



Note: icons / images should be 32 x 32 pixels.

4.7 Create the Custom .rels (Relationships) XML File

Create the custom .rels (Relationship) file adding the image Relationships

```
<Relationship Id="broom"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="images/broom.png"/>
```

as follows...

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">

<!-- Add any required custom images for the application in the following format --
>

<Relationship Id="broom"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="images/broom.png"/>

<Relationship Id="information"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
Target="images/information.png"/>

</Relationships>
```

Save the file as

custom_ribbon.xml.rels

Note: It is important to end the file name with .xml.rels otherwise the file can't be located.

4.8 Create the Custom Ribbon XML File

Create the custom ribbon XML file as follows...

```
<customUI onLoad="CustomRibbonRibbonOnLoad"
xmlns="http://schemas.microsoft.com/office/2006/01/customui">

<ribbon startFromScratch="false">

<tabs>

<tab id="custom_ribbon" label="Custom Ribbon">

<group id="Group01" label="Custom Ribbon Group 1">

<button id="myButtonG0101" label="Custom Ribbon Form" imageMso="GroupMacro"
size="large" onAction="rbnRunCustomRibbonForm" supertip="Run Custom Ribbon Form."
/>

</group>

<group id="Group02" label="Custom Ribbon Group 2">

<button id="myButtonG0201" label="Broom" image="broom" size="large"
onAction="rbnRunCustomRibbonBroom" supertip="Run Custom Ribbon Broom." />

</group>

<group id="Group03" label="Custom Ribbon Group 3">

<button id="myButtonG0301" label="Custom Ribbon Information"
image="information" size="large" onAction="rbnCustomRibbonInformation"
supertip="Custom ribbon information." />

<button id="myButtonG0302" label="Unload Custom Ribbon"
imageMso="BroadcastEnd" size="large" onAction="rbnUnloadCustomRibbon"
supertip="Unload Custom Ribbon." />

</group>

</tab>

</tabs>

</ribbon>


</customUI>
```

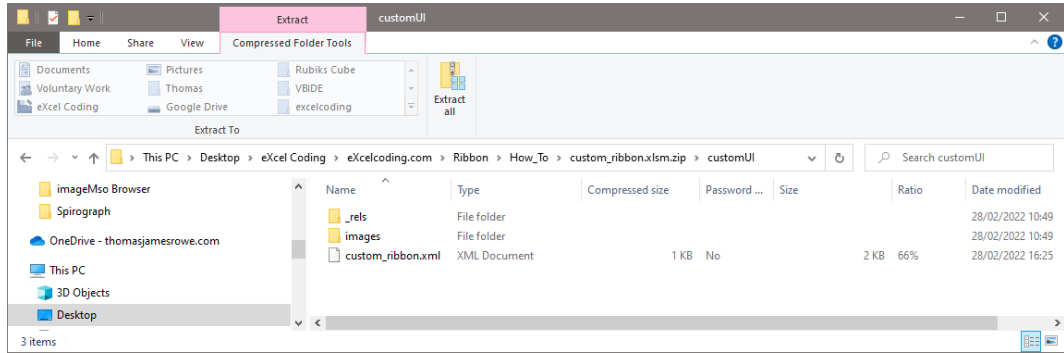
Note: The built in imageMso icons are defined as imageMso =, custom icons are defined as image =.

Save the file as custom_ribbon.xml i.e. the file name that was used in the .rels (Relationships) file.

```
<!-- Change the customUI/file_name.xml to suit your application -->
```

```
<Relationship Id="customUIXML"
Type="http://schemas.microsoft.com/office/2007/relationships/ui/extensibility"
Target="customUI/custom_ribbon.xml"/>
```

Copy the file to folder  customUI in custom_ribbon.zip.



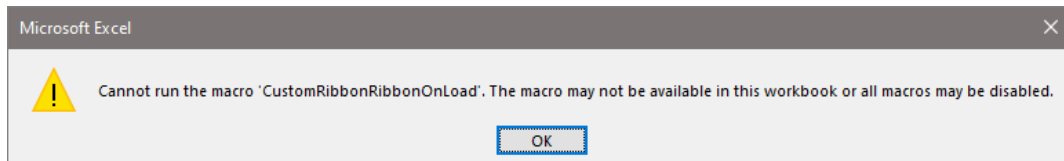
4.9 Rename custom_ribbon.zip to custom_ribbon.xlsm

Rename custom_ribbon.zip to custom_ribbon.xlsm.

4.10 Open custom_ribbon.xlsm

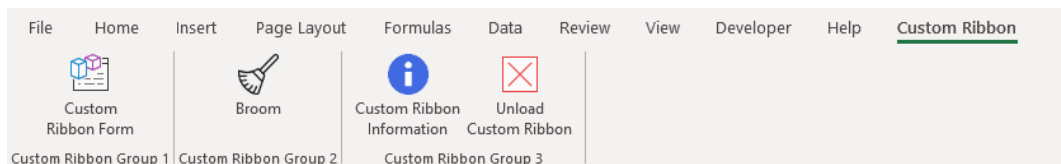
Open custom_ribbon.xlsm

Note: You will get the following error, this is because the macro CustomRibbonRibbonOnLoad has not yet been created.



4.10.1 Custom Ribbon Bar

You should see the following **Custom Ribbon** bar...



4.11 Add the VBA Macros / Code

Insert a Module modCustomRibbon and add the VBA macros / code for the following ribbon onload and onAction's...

```
onLoad="CustomRibbonRibbonOnLoad"
```

```
onAction="rbnRunCustomRibbonForm"
```

```
onAction="rbnRunCustomRibbonBroom"
```

```
onAction="rbnCustomRibbonInformation"
```

```
onAction="rbnUnloadCustomRibbon"
```

In declarations...

```
' Ribbon Bar.  
Public objRibbon As IRibbonUI
```

Add the following Subs.

```
Public Sub CustomRibbonRibbonOnLoad(objIRibbonUI As IRibbonUI)  
Set objRibbon = objIRibbonUI  
' Add any code that is required to be run once the custom ribbon has been loaded.  
End Sub
```

```
Public Sub rbnCustomRibbonBroom(Optional objRibbonControl As IRibbonControl)  
strMsg = "You clicked the broom icon in the custom ribbon bar"  
lngStyle = vbOKOnly + vbInformation  
strTitle = "Broom Icon Clicked"  
Beep  
intResponse = MsgBox(strMsg, lngStyle, strTitle)  
End Sub
```

```
Public Sub rbnCustomRibbonInformation(Optional objRibbonControl As IRibbonControl)  
strMsg = "Title : Custom Ribbon" & vbCrLf & _  
        "Version : Version 1.0 [28/02/2022]"  
lngStyle = vbOKOnly + vbInformation  
strTitle = "Add-In Information"  
Beep  
intResponse = MsgBox(strMsg, lngStyle, strTitle)  
End Sub
```

```
Public Sub rbnRunCustomRibbonForm(Optional objRibbonControl As IRibbonControl)  
frmVBACustomRibbon.Show  
End Sub
```



```
Public Sub rbnUnloadCustomRibbon(Optional objRibbonControl As IRibbonControl)
    strMsg = "Do you want to unload Custom Ribbon"
    lngStyle = vbYesNo + vbDefaultButton2 + vbQuestion
    strTitle = "Unload Custom Ribbon ?"
    Beep
    intResponse = MsgBox(strMsg, lngStyle, strTitle)
    If intResponse = vbYes Then
        ThisWorkbook.Close SaveChanges:=False
    End If
End Sub
```

5. Trouble Shooting

If your Custom Ribbon Bar is not displayed it is usually due to a mistake in editing or creating the necessary XML files, or the XML is invalid. You can validate your XML files at.

<https://www.xmlvalidation.com/>

6. Acronyms and Abbreviations

GNU	General Public License
imageMSO	image Microsoft Office
XML	Extensible Markup Language

7. References

8. Revision Log

Date	Amendments	Updated by
19/04/2022	Initial release.	T.J.Rowe