

## Adding a Macro Ribbon to Excel 2007

---

### Software

The Custom UI Editor Tool can be downloaded from:

Information <http://openxmldeveloper.org/articles/customuieditor.aspx>

Download <http://openxmldeveloper.org/attachment/808.ashx>

### Setting up the Macros

Your macros need to be modified as follows:

#### Normal Macro:

```
Sub TestMacro()  
    ...  
End Sub
```

#### Modified for Ribbon:

```
Sub TestMacro (control As IRibbonControl)  
    ...  
End Sub
```

They will no longer appear as macros. If you need to have both, you can try something like this:

```
Sub TestMacro()  
    ...  
End Sub  
  
Sub TestMacroRibbon (control As IRibbonControl)  
    TestMacro  
End Sub
```

Since you can't have two procedures with the same name, you will need a new name for either the macro or the ribbon version. The ribbon version can then simply call the macro version.

## Installing the Ribbon

The ribbon is installed by attaching XML data to the Excel file.

1. Run the UI Editor and open the Excel Document

You cannot have it open in Excel if you plan to save the XML

2. Copy and Paste the sample XML text.

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
<ribbon>
  <tabs>
    <tab id="sample" label="Sample">
      <group id="changeValues" label="Change Values">
        <button id="idNegate" label="Change Sign" onAction="negate" />
        <button id="idAddGST" label="Add GST" onAction="addGST" />
        <button id="idRemoveGST" label="Remove GST" onAction="removeGST" />
        <separator id="changeValues1"/>
        <button id="idIncrement" label="Increment" onAction="increment" />
        <button id="idDecrement" label="Decrement" onAction="decrement" />
      </group>
      <group id="changeCase" label="Change Case">
        <button id="idUpperCase" label="UPPER Case" onAction="upperCase" />
        <button id="idLowerCase" label="lower Case" onAction="lowerCase" />
        <button id="idTitleCase" label="Title Case" onAction="titleCase" />
      </group>
      <group id="format" label="Format">
        <button id="idDateFormat" label="date" onAction="DateFormat"/>
      </group>
    </tab>
  </tabs>
</ribbon>
</customUI>
```

3. Fill in the following:

- (a) `tab id="..."` Unique id of the tab  
`tab label="..."` User-friendly label for the tab
- (b) `group id="..."` Unique id of the group  
`group label="..."` User-friendly label for the group
- (c) `button id="..."` Unique id of the button;  
you can use `idAction`  
`button label="..."` User-friendly label for the button  
`button onAction="..."` The name of the procedure as per step 2.
- (d) `separator id="..."` Unique id of the separator;  
you can use `groupN`

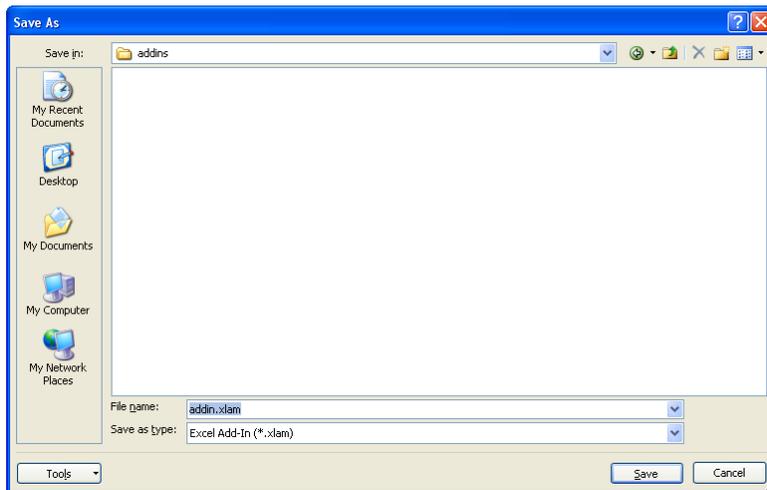
4. Note the XML syntax:

- (a) `<button />` and `<separator />` elements are empty.
- (b) all tags and attributes are case-sensitive
- (c) id attribute must be unique

# Add-Ins

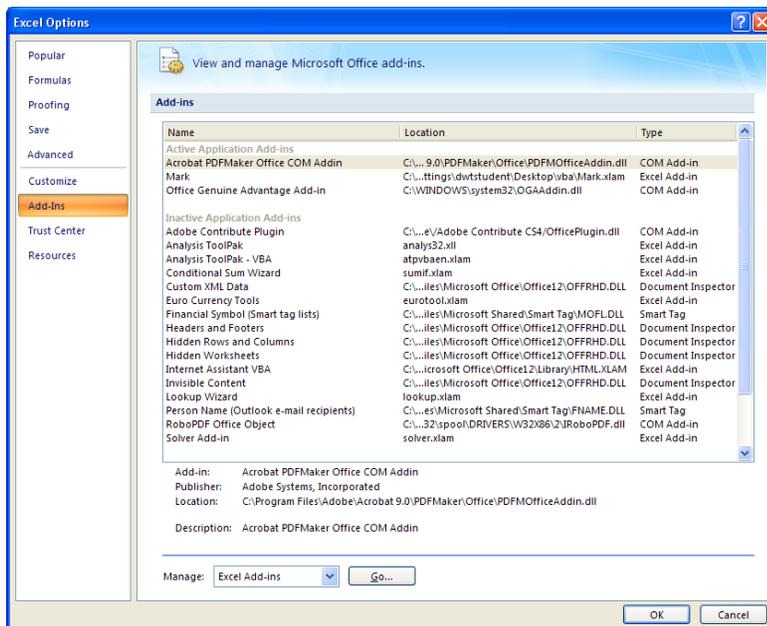
## Saving and Add-In

Save the File as an Add-In, in your own specified folder:



## Using an Add-In

In Excel Options, choose Add-Ins

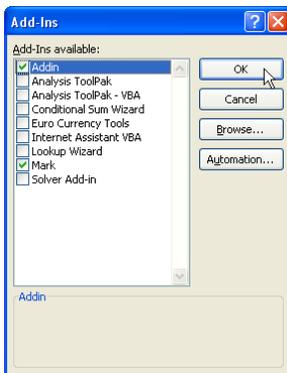
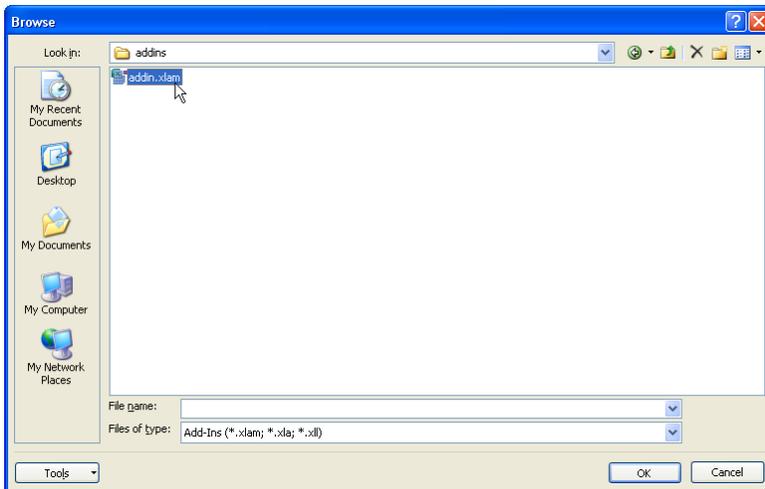
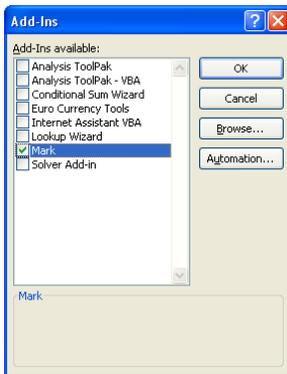


## Excel VBA Notes

---

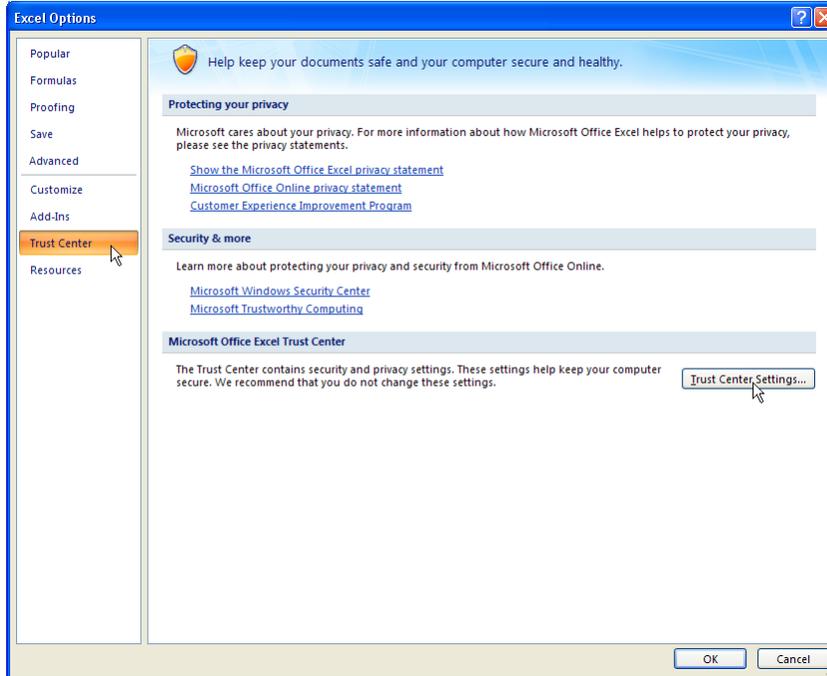
At the bottom of the screen, choose:

Manage Excel Add-Ins: Go...

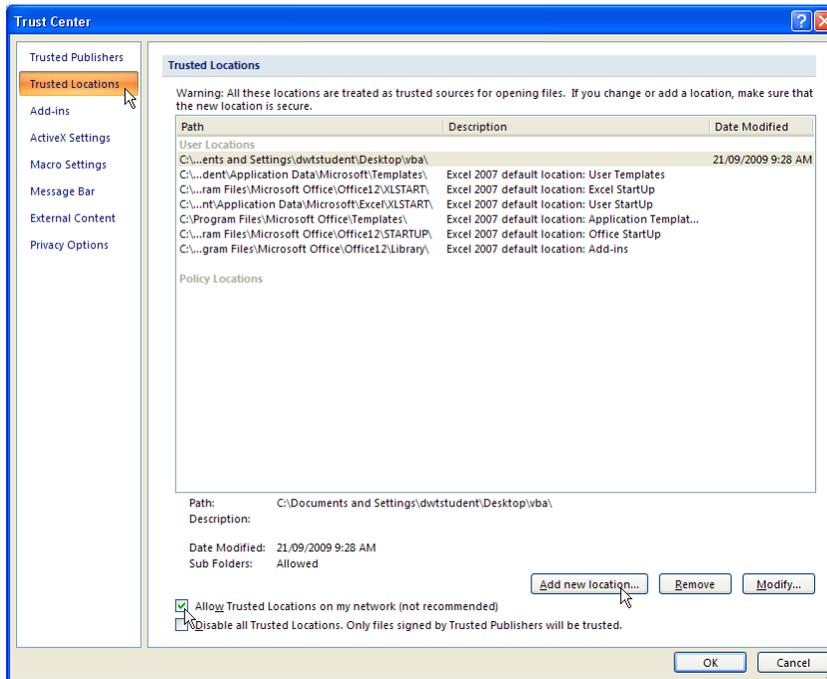


# Trusting a Location

Excel Options : Trust Center

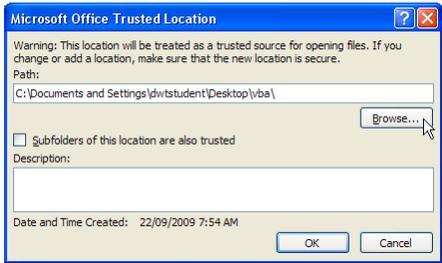


Trust Center Settings ...

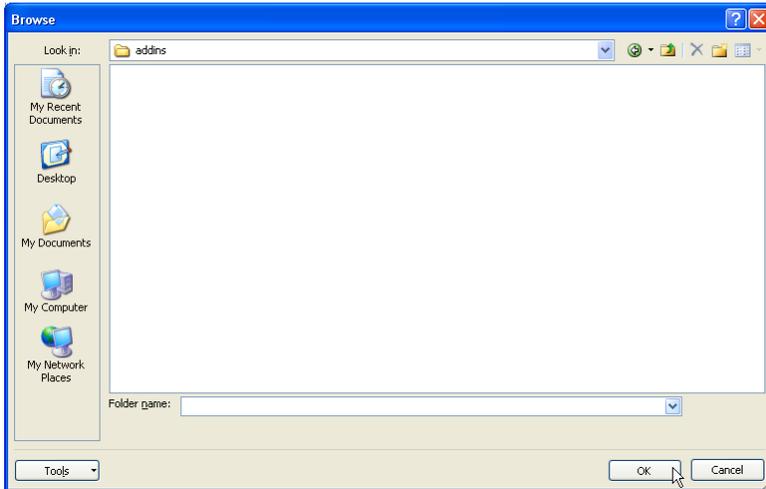


Add New Location...

Browse ...



Select your Folder



# Trusting a Publisher

---

Open a Signed File

You will see this message:



Trust all from publisher

# Digital Certificates

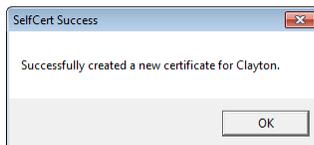
---

## Creating a Digital Certificate

Microsoft Office: Microsoft Office Tools : Digital Certificate for VBA Projects



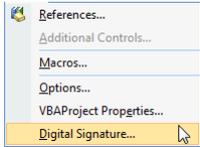
Enter any name that you wish.



## Applying a Digital Certificate

### In VBA

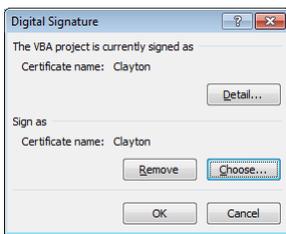
Tools : Digital Signature



Choose ...



Choose Your certificate



OK