ಐಟೆಕ್ ಅನಾಲಿಟಿಕ್ ಸಲೂಷನ್ಸ್

iTech Analytic Solutions

No. 9, 1st Floor,
8th Main, 9th Cross,
SBM Colony, Brindavan Nagar,
Mathikere, Bangalore – 560 054

Email: itechanalytcisolutions@gmail.com
Website: www.itechanalytcisolutions.com
Mobile: 9902058793

Chapter 2

Introduction to Macros

What is a Macro?

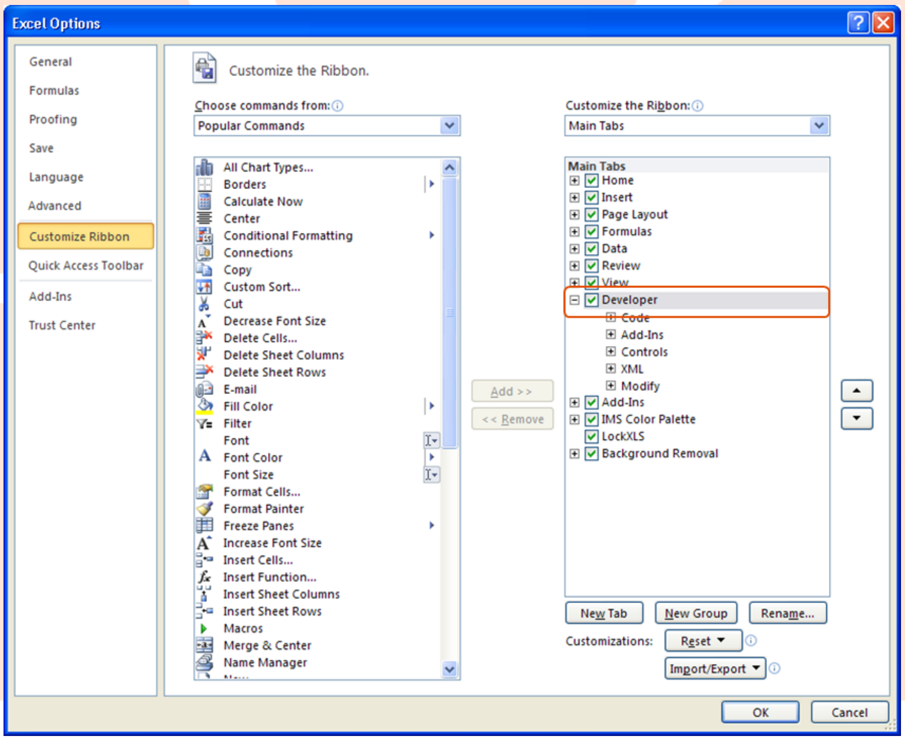
A macro is a way to automate a task that you perform repeatedly or on a regular basis. It is a series of commands and actions that can be stored and run whenever you need to perform the task. You can record or build a macro and then run it to automatically repeat the series of steps or actions.

A macro is a set of commands that can be played back at will to perform a given task. These tasks may be something as simple as inserting your name and address into a document or more complex, such as reading all of the data from a file and entering it into a database.

Tasks performed by macros are typically repetitive in nature and can provide significant time savings. Let the computer work for you instead of typing and clicking the same things over and over again.

Turn on the Developer Tab

- On the File tab, choose Options to open the Excel Options dialog box.
- Click Customize Ribbon on the left side of the dialog box.
- Under Choose commands from on the left side of the dialog box, select Popular Commands.
- Under Customize the ribbon on the right side of the dialog box, select Main tabs, and then select the Developer check box.
- Click OK.



Excel Options

Customize the Ribbon.

Choose commands from: Popular Commands

Customize the Ribbon: Main Tabs

Main Tabs: Home ✓, Insert ✓, Page Layout ✓, Formulas ✓, Data ✓, Review ✓, View ✓, Developer ✓ (Code, Add-Ins, Controls, XML, Modify), Add-Ins ✓, IMS Color Palette ✓, LockXLS ✓, Background Removal ✓

Recording a Simple Macro

Although you can write your own complex macros in the Visual Basic programming language, the easiest method for creating many macros is to use the macro recorder. When you record a macro, Excel stores information about each step you take as you perform a series of commands. You then run the macro to repeat, or play back, the commands.

The macro recorder records every action you complete. Therefore, planning your macro before you begin the recording process is very important so you don't record unnecessary steps.

Record a macro

Follow these steps to record a macro:

1. **Choose Record Macro in the Code group of the Developer tab.**

   The Record Macro dialog box appears.

2. **Type a name for the macro in the Macro Name text box.**

   The first character of the macro name must be a letter, and the name cannot contain spaces or cell references. Macro names are not case-sensitive.

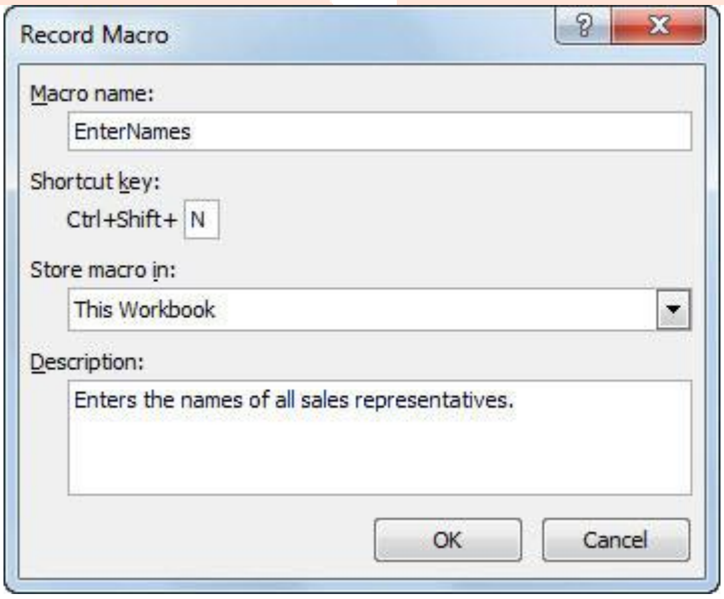
3. **(Optional) Assign a Shortcut Key.**

   If you select a shortcut key already used in Excel, the macro shortcut key overrides the Excel shortcut key while the workbook that contains the macro is open.

4. **From the Store Macro In drop-down list, select where you want to store the macro:**

   - **This Workbook:** Save the macro in the current workbook file.
   - **New Workbook:** Create macros that you can run in any new workbooks created during the current Excel session.
   - **Personal Macro Workbook:** Choose this option if you want the macro to be available whenever you use Excel, regardless of which worksheet you're using.

5. **(Optional) Type a description of the macro in the Description text box.**



Record Macro

Macro name:
EnterNames

Shortcut key:
Ctrl+Shift+ N

Store macro in:
This Workbook

Description:
Enters the names of all sales representatives.

6. **Click OK.**

   The Record Macro option on the Developer tab changes to Stop Recording.

7. **Perform the actions you want to record.**

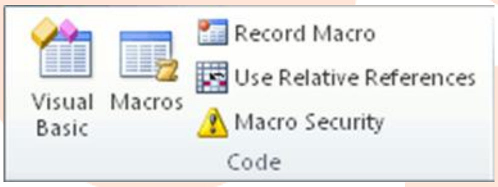
8. **Choose Stop Recording in the Code group of the Developer tab.**
   The macro recorder stops recording keystrokes and the macro is complete.

Run a macro

There are several ways to run a macro in Microsoft Excel. A macro is an action or a set of actions that you can use to automate tasks. Macros are recorded in the Visual Basic for Applications programming language. You can always run a macro by clicking the **Macros** command on the ribbon (**Developer** tab, **Code** group). Depending on how a macro is assigned to run, you might also be able to run it by pressing a CTRL combination shortcut key, by clicking a button on the Quick Access Toolbar or in a custom group on the ribbon. or by clicking an area on an object, graphic, or control. In addition, you can run a macro automatically when you open a workbook.

Note When you set the macro security level in Excel to **Disable all macros without notification**, Excel will run only those macros that are digitally signed or stored in a trusted location, such as the Excel startup folder on your computer. If the macro that you want to run is not digitally signed or located in a trusted location, you can temporarily change the security level that enables all macros.

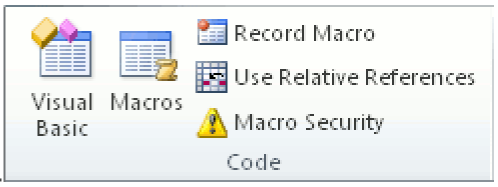
1. Open the workbook that contains the macro.
2. On the **Developer** tab, in the **Code** group, click **Macros**.
3. In the **Macro name** box, click the macro that you want to run



4. Do one of the following:
   - To run a macro in an Excel workbook, click **Run**.
     Tip You can also press CTRL+F8 to run the macro. You can interrupt the execution of the macro by pressing ESC.
   - To run a macro from a Microsoft Visual Basic for Applications (VBA) module, click **Edit**, and then on the**Run** menu, click **Run Sub/UserForm** , or press F5.

Run a macro by pressing a CTRL combination shortcut key

1. If the **Developer** tab is not available, do the following to display it:
   1. Click the **File** tab, click **Options**, and then click the **Customize Ribbon** category.
   2. In the **Main Tabs** list, select the **Developer** check box, and then click **OK**.



2. On the **Developer** tab, in the **Code** group, click **Macros**.
3. In the **Macro name** box, click the macro that you want to assign to a CTRL combination shortcut key.

4. Click **Options**.

   The **Macro Options** dialog box appears.

5. In the **Shortcut key** box, type any lowercase letter or uppercase letter that you want to use with the CTRL key.

   Note The shortcut key will override any equivalent default Excel shortcut key while the workbook that contains the macro is open.

   For a list of CTRL combination shortcut keys that are already assigned in Excel, see the article Excel shortcut and function keys.

6. In the **Description** box, type a description of the macro.
7. Click **OK** to save your changes, and then click **Cancel** to close the **Macro** dialog box.

Run a macro by clicking a button on the Quick Access Toolbar

To add a button to the Quick Access Toolbar that will run a macro, do the following:

1. Click the **File** tab, **Options**, and then click **Quick Access Toolbar**.
2. In the **Choose commands from** list, select **Macros**.
3. In the list, click the macro that you created, and then click **Add**.
4. To change the button image of the macro, select the macro in the box to which it was added, and then click**Modify**.
5. Under **Symbol**, click the button image that you want to use.
6. To change the name of the macro that is displayed when you rest the pointer on the button, in the **Display name** box, type the name that you want to use.
7. Click **OK** to add the macro button to the **Quick Access Toolbar**.
8. On the **Quick Access Toolbar**, click the macro button that you just added.

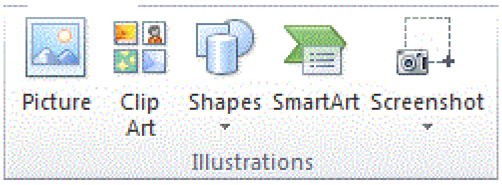
Run a macro by clicking a button in a custom group on the ribbon

By taking advantage of the customizability of the ribbon in Excel 2010, you can create a custom group that appears on a tab in the ribbon, and then assign a macro to a button in that group. For example, you can add a custom group named "My Macros" to the Developer tab, and then add a macro (that appears as a button) to the new group.

Run a macro by clicking an area on a graphic object

You can create a hot spot on a graphic that users can click to run a macro.

1. In the worksheet, insert a graphic object, such as a picture, clip art, shape, or SmartArt.
2. To create a hot spot on the existing object, on the **Insert** tab, in the **Illustrations** group, click **Shapes**, select the shape that you want to use, and then draw that shape on the existing object.

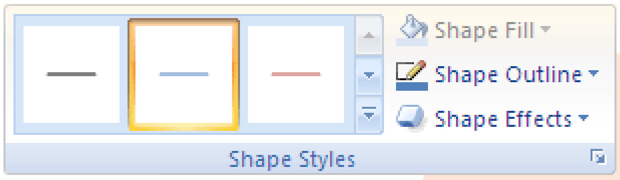


3. Right-click the hot spot that you created, and then click **Assign Macro**.
4. Do one of the following:

- To assign an existing macro to the graphic object, double-click the macro or enter its name in the **Macro name** box.
- To record a new macro to assign to the selected graphic object, click **Record**, type a name for the macro in the **Record Macro** dialog box, and then click **OK** to begin recording your macro. When you finish recording the macro, click **Stop Recording** on the **Developer** tab in the **Code** group.

  Tip You can also click **Stop Recording** on the left side of the status bar.
- To edit an existing macro, click the name of the macro in the **Macro name** box, and then click **Edit**.

5. Click **OK**.
6. In the worksheet, select the hot spot. This displays the **Drawing** Tools, adding a **Format** tab.
7. On the **Format** tab, in the **Shape Styles** group, click the arrow next to **Shape Fill**, and then click **No Fill**.



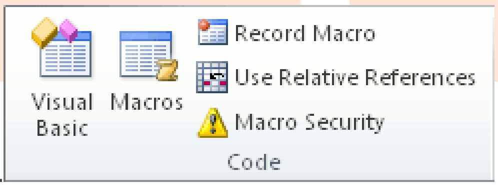
8. Click the arrow next to **Shape Outline**, and then click **No Outline**.

Configure a macro to run automatically upon opening a workbook

If you record a macro and save it with the name "Auto_Open," the macro will run whenever you open the workbook that contains the macro. Another way to automatically run a macro when you open a workbook is to write a VBA procedure in the **Open** event of the workbook by using the Visual Basic Editor. The **Open** event is a built-in workbook event that runs its macro code every time you open the workbook.

Create an Auto_Open macro

1. If the **Developer** tab is not available, do the following to display it:
   1. Click the **File** tab, and then click **Options**.
   2. In the **Customize Ribbon** category, in the **Main Tabs** list, select the **Developer** check box, and then click**OK**.
2. To set the security level temporarily to enable all macros, do the following:
   1. On the **Developer** tab, in the **Code** group, click **Macro Security**



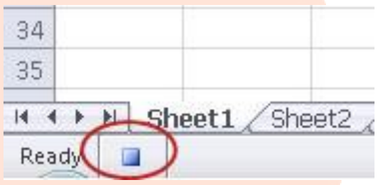
   2. In the **Macro Settings** category, under **Macro Settings**, click **Enable all macros (not recommended, potentially dangerous code can run)**, and then click **OK**.

   Note To help prevent potentially dangerous code from running, we recommend that you return to any one of the settings that disable all macros after you finish working with macros.

1. If you want to save the macro with a particular workbook, open that workbook first.
2. On the **Developer** tab, in the **Code** group, click **Record Macro**.
3. In the **Macro name** box, type **Auto_Open**.
4. In the **Store macro in** list, select the workbook where you want to store the macro.

   Tip If you want a macro to be available whenever you use Excel, select **Personal Macro Workbook**. When you select **Personal Macro Workbook**, Excel creates a hidden personal macro workbook (Personal.xlsb), if it does not already exist, and saves the macro in this workbook. In Windows Vista, this workbook is saved in the C:\Users\user name\AppData\Local\Microsoft\Excel\XLStart folder. If you can't find it there, it may have been saved in the Roaming subfolder instead of Local. In Microsoft Windows XP, this workbook is saved in the C:\Documents and Settings*user name*\Application Data\Microsoft\Excel\XLStart folder. Workbooks in the XLStart folder are opened automatically whenever Excel starts. If you want a macro in the personal macro workbook to be run automatically in another workbook, you must also save that workbook in the XLStart folder so that both workbooks are opened when Excel starts.
5. Click **OK**, and then perform the actions that you want to record.
6. On the **Developer** tab, in the **Code** group, click **Stop Recording** ■.

   Tip You can also click **Stop Recording** on the left side of the status bar.



Notes

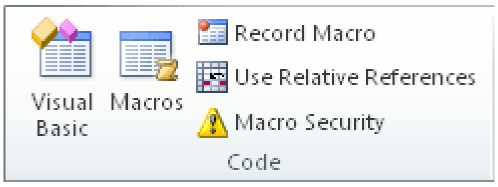
- If you chose to save the macro in **This Workbook** or **New Workbook** in step 6, save or move the workbook into one of the XLStart folders.
- Recording an Auto_Open macro has the following limitations:
- If the workbook where you save the Auto_Open macro already contains a VBA procedure in its **Open** event, the VBA procedure for the **Open** event will override all actions in the Auto_Open macro.
- An Auto_Open macro is ignored when a workbook is opened programmatically by using the **Open** method.
- An Auto_Open macro runs before any other workbooks open. Therefore, if you record actions that you want Excel to perform on the default Book1 workbook or on a workbook that is loaded from the XLStart folder, the Auto_Open macro will fail when you restart Excel, because the macro runs before the default and startup workbooks open.

  If you encounter these limitations, instead of recording an Auto_Open macro, you must create a VBA procedure for the **Open** event as described in the next section of this article.
- If you want Excel to start without running an Auto_Open macro, hold down the SHIFT key when you start Excel.

Create a VBA procedure for the Open event of a workbook

The following example uses the **Open** event to run a macro when you open the workbook.

1. If the **Developer** tab is not available, do the following to display it:
1. Click the **File** tab, and then click **Options**.
2. In the **Customize Ribbon** category, in the **Main Tabs** list, select the **Developer** check box, and then click**OK**.
2. To set the security level temporarily to enable all macros, do the following:
1. On the **Developer** tab, in the **Code** group, click **Macro Security**.



2. In the **Macro Settings** category, under **Macro Settings**, click **Enable all macros (not recommended, potentially dangerous code can run)**, and then click **OK**.

   Note To help prevent potentially dangerous code from running, we recommend that you return to any one of the settings that disable all macros after you finish working with macros.
1. Save and close all open workbooks.
2. Open the workbook where you want to add the macro, or create a new workbook.
3. On the **Developer** tab, in the **Code** group, click **Visual Basic**.
4. In the Project Explorer window, right-click the **ThisWorkbook** object, and then click **View Code**.

   Tip If the Project Explorer window is not visible, on the **View** menu, click **Project Explorer**.
5. In the **Object** list above the Code window, select **Workbook**.

   This automatically creates an empty procedure for the **Open** event, such as this:

```
Private Sub Workbook_Open()

End Sub
```

6. Add the following lines of code to the procedure:

```
Private Sub Workbook_Open()
MsgBox Date
Worksheets("Sheet1").Range("A1").Value = Date
End Sub
```

7. Switch to Excel and save the workbook as a macro-enabled workbook (.xlsm).
8. Close and reopen the workbook. When you open the file again, Excel runs the Workbook_Open procedure, which displays today's date in a message box.
9. Click **OK** in the message box.

   Note that cell A1 on Sheet1 also contains the date as a result of running the Workbook_Open procedure

Edit and Delete a Macro in Microsoft Excel 2010

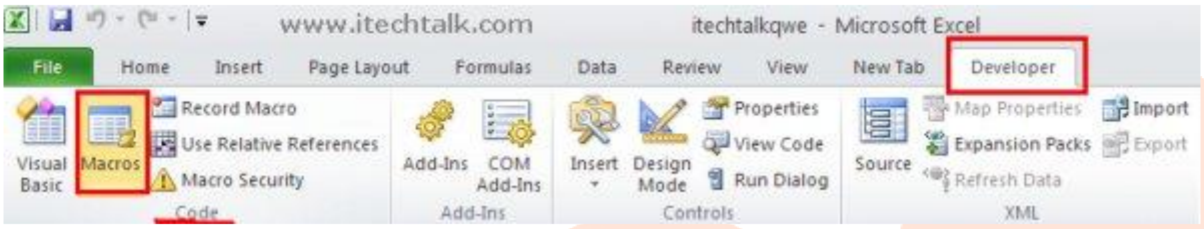
Follow these simple steps mentioned below to edit and delete a Macro. That already exists.

1) Firstly, make the Developer Tab visible on the ribbon.

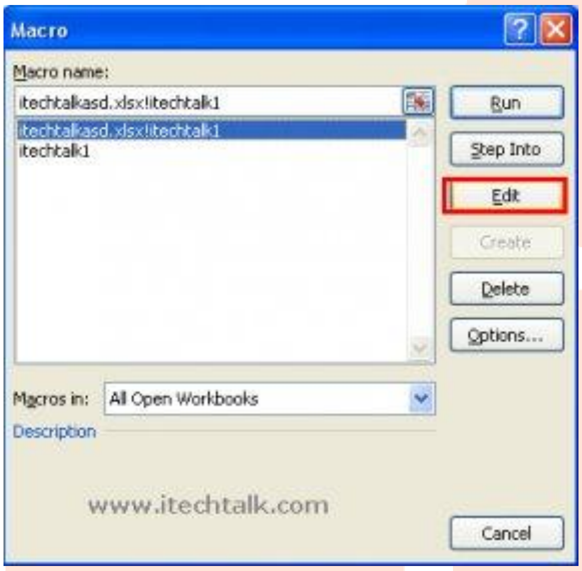
2) Enable all the Macros in the Excel application.

3) Open the workbook that contains Macros that you wish to edit or delete.

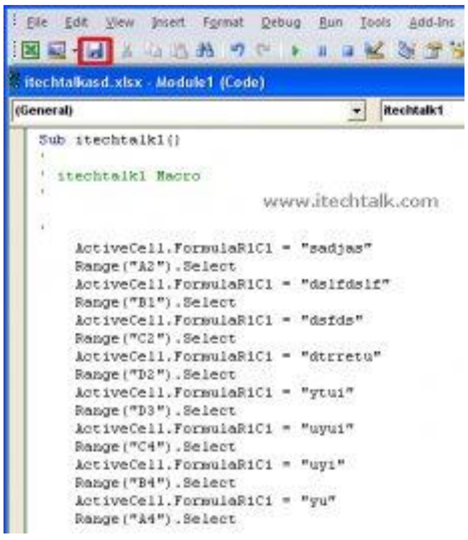
4) To edit a Macro, click on the Developer tab and then under the Developer tab, under the Code category, click on the "Macros" icon.



5) Now, in the Macro dialogue box thus opened, select the Macro that you want to edit and then click on the 'Edit' button.



6) Once you click on the Edit button, the VBA editor opens up in a separate window with the code of the Macro displayed in it.

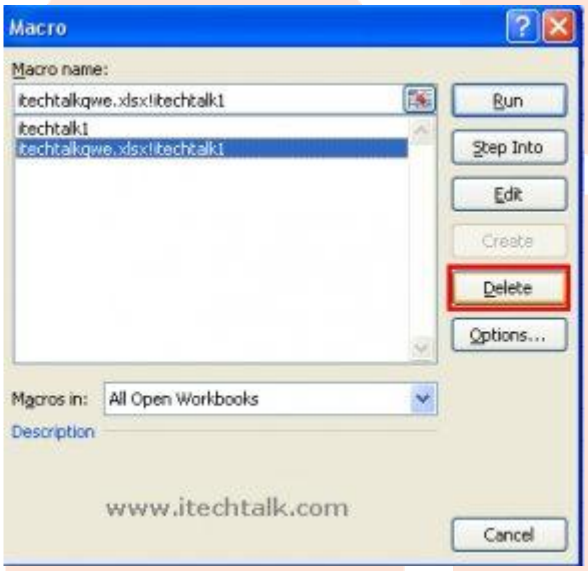

File Edit View Insert Format Debug Run Tools Add-Ins
itechtalkasd.xlsx - Module1 (Code)
(General)
itechtalk1
www.itechtalk.com

```
Sub itechtalk1()
'
' itechtalk1 Macro
'

'
    ActiveCell.FormulaR1C1 = "sadjas"
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "dslfdslf"
    Range("B1").Select
    ActiveCell.FormulaR1C1 = "dsfds"
    Range("C2").Select
    ActiveCell.FormulaR1C1 = "dtrretu"
    Range("D2").Select
    ActiveCell.FormulaR1C1 = "ytui"
    Range("D3").Select
    ActiveCell.FormulaR1C1 = "uyui"
    Range("C4").Select
    ActiveCell.FormulaR1C1 = "uyi"
    Range("B4").Select
    ActiveCell.FormulaR1C1 = "yu"
    Range("A4").Select
```

7) Now, you can edit the Macro in the VBA editor and once finished editing, save the Macro by clicking on the Save button on the Menu bar of the VBA editor.

8) If you wish to delete the Macro, in the Macro dialogue box, select the Macro that you wish to delete and then click on the 'Delete' button.



Macro name: itechtalkqwe.xlsx!itechtalk1

itechtalk1
itechtalkqwe.xlsx!itechtalk1

Macros in: All Open Workbooks

So, this is how you can edit and delete Macros in Microsoft excel 2010 application.

Setting Macro Security

In Microsoft Excel, you can change the macro security settings to control which macros run and under what circumstances when you open a workbook. For example, you might allow macros to run based on whether they are digitally signed by a trusted developer (a person who writes programming code).

Macro security settings and their effects

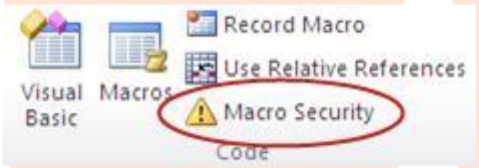
The following list summarizes the various macro security settings. Under all settings, if antivirus software that works with Microsoft Office 2010 is installed and the workbook contains macros, the workbook is scanned for known viruses before it is opened.

- **Disable all macros without notification** Click this option if you don't trust macros. All macros in documents and security alerts about macros are disabled. If there are documents that contain unsigned macros that you do trust, you can put those documents into a trusted location. Documents in trusted locations are allowed to run without being checked by the Trust Center security system.
- **Disable all macros with notification** This is the default setting. Click this option if you want macros to be disabled, but you want to get security alerts if there are macros present. This way, you can choose when to enable those macros on a case by case basis.
- **Disable all macros except digitally signed macros** This setting is the same as the **Disable all macros with notification** option, except that if the macro is digitally signed by a trusted publisher, the macro can run if you have already trusted the publisher. If you have not trusted the publisher, you are notified. That way, you can choose to enable those signed macros or trust the publisher. All unsigned macros are disabled without notification.
- **Enable all macros (not recommended, potentially dangerous code can run)** Click this option to allow all macros to run. Using this setting makes your computer vulnerable to potentially malicious code and is not recommended.
- **Trust access to the VBA project object model** This setting is for developers and is used to deliberately lock out or allow programmatic access to the VBA object model from any Automation client. In other words, it provides a security option for code that is written to automate an Office program and programmatically manipulate the Microsoft Visual Basic for Applications (VBA) environment and object model. This is a per user and per application setting, and denies access by default. This security option makes it more difficult for unauthorized programs to build "self-replicating" code that can harm end-user systems. For any Automation client to be able to access the VBA object model programmatically, the user running the code must explicitly grant access. To turn on access, select the check box.

Change macro security settings

You can change macro security settings in the Trust Center, unless a system administrator in your organization has changed the default settings to prevent you from changing the settings.

1. On the **Developer** tab, in the **Code** group, click **Macro Security**.



Note If the **Developer** tab is not available, do the following to display it:

   a) Click the **File** tab, click **Options**, and then click the **Customize Ribbon** category.
   b) In the **Main Tabs** list, select the **Developer** check box, and then click **OK**.
   c) Click any other tab to return to your file.

2. In the **Macro Settings** category, under **Macro Settings**, click the option that you want. For detailed information about these settings, see the section Macro security settings and their effects, earlier in this article.

   Note Any changes that you make in the **Macro Settings** category in Excel apply only to Excel and do not affect any other Microsoft Office program.

Tip You can also access the Trust Center in the **Options** dialog box.

1. Click the **File** tab, click **Options**, and then click the **Trust Center** category.
2. Click **Trust Center Settings**, and then click the **Macro Settings** category

Saving a Document as Macro Enabled

When working with macros in Excel 2010, you save and open workbooks in a new macro-enabled workbook format (.xlsm) that provides added security. When you create a macro, you must use a macro-enabled format to save your workbook or the macro won't be saved.

If you open a macro-enabled workbook, a Security Warning message states that the workbook contains macros. This protects you from possible harm — some macros might contain viruses or other hazards. You can choose to enable the content if the workbook is from a trusted source.

Follow these steps to save a macro-enabled workbook:

Save a macro-enabled workbook

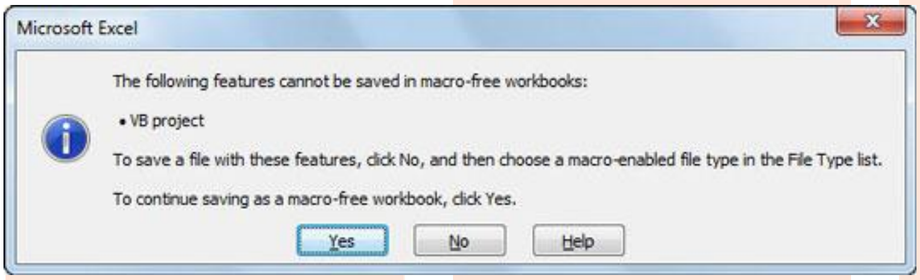
1. **Click the File tab and then choose Save As.**

   The Save As dialog box appears.
2. **Enter a name and select a location for your workbook.**
3. **Click the Save as Type drop-down arrow.**

   A list of file types appears.
4. **Select Excel Macro-Enabled Workbook.**

   Excel adds the .xlsm extension to the filename.
5. **Click Save.**

   If you create a macro in a workbook and neglect to save the workbook as a macro-enabled workbook, you see a warning message telling you that the macro will not be retained.


Microsoft Excel

The following features cannot be saved in macro-free workbooks:

• VB project

To save a file with these features, click No, and then choose a macro-enabled file type in the File Type list.

To continue saving as a macro-free workbook, click Yes.

Open a macro-enabled workbook

To open a macro-enabled workbook, follow these steps:

1. **Open a workbook containing a macro in the same way you open any other workbook.**

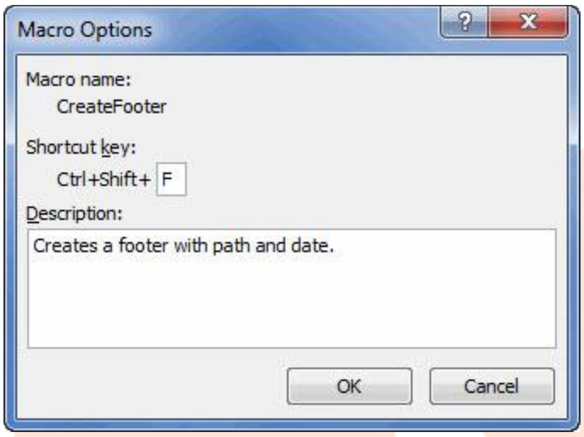
   The workbook opens as usual, but a Security Warning message appears below the Ribbon.
2. **If you know where the macro originated, click the Enable Content button.**

   The Security Warning message closes, and you're free to use the macros in that workbook

Assigning a Macro to an Object

Do you find yourself forgetting keyboard shortcuts for your Excel macros? When first recording a macro, you will have an opportunity in the Record Macro dialog box to specify a shortcut key for the macro. However, you may eventually forget the shortcut, or if you didn't assign a shortcut key when you first recorded the macro, you may now like to assign one – or perhaps you have even decided to change the shortcut key to something different.

To view your macro shortcut keys, on the View tab of the Ribbon, click the Macros button, select your macro, and choose Options…. You will see the Macro Options dialog box:


Macro Options

Macro name:
CreateFooter

Shortcut key:
Ctrl+Shift+ F

Description:
Creates a footer with path and date.

This dialog box shows the shortcut key.

Here you can view the macro shortcut and description, assign a shortcut, or modify the existing shortcut keys. When revising shortcuts, it doesn't matter whether you saved the macro in this workbook or in the Personal Macro workbook. Just make sure that you do not use an existing Excel keyboard shortcut (such as CTRL B, which bolds a cell). In addition, use a single letter with the CTRL or Shift keys.

Note: The CTRL + SHIFT + single letter keyboard combination may work best for a macro shortcut due to the many Excel shortcuts which already use the CTRL + single letter combinations.

Adding a Macro Button to the Quick Access Toolbar

Rather than creating a keyboard shortcut, you can add a macro button to the Quick Access Toolbar. (In Microsoft Office 2010, it is possible to add a new group to the Ribbon so you may prefer to add a Macro group that will include all of your macros).

To add a macro to the Quick Access Toolbar:

1. Open the Excel Options dialog box and click the Quick Access Toolbar link in the left pane. (If you want to add the macro to the Ribbon, choose Customize Ribbon. You will have the option to add a new group to the Ribbon to which you can add your macros).
2. In the "Choose Commands From" drop-down menu, select Macros. Excel macros, including your custom macros, will display.

3. Select the macros that you want to add to the Quick Access Toolbar, and click Add >>. Note that there is a <Separator> at the top of the list, which you can add to separate the macros from other commands on the Quick Access Toolbar).

4. Then click the Modify… button at the bottom of the right pane if you want to rename the macro or if you want to select a different icon.

   Once you click OK, the macros will appear on the Quick Access Toolbar and will always be available while you are working on that same computer.

Adding a Macro Button to the Worksheet

You can also add a macro button to the worksheet by following these steps:

1. On the Developer tab of the Ribbon, in the Controls group, click the Insert button. Choose **Button (Form Control)**.
2. Press and drag somewhere in the worksheet to create a button.
3. The Assign Macro dialog box appears, which displays all of your macros. Select the desired macro and click OK.
4. The button appears on the worksheet. You can change the text of the button while it is selected or at a later time, you can right-click the button and choose Edit Text. You can also format the button by right-clicking the button and choosing Format Control….