

Chapter 1 : MS Excel Project Explorer in VBA Environment

Using Macros in Excel - A beginner's tutorial for Microsoft Excel to learn basic to advance concepts step by step including open workbooks, format workbooks, save workbooks, email workbooks, formulas setting, pivot tables, pivot charts, print workbooks and worksheets, zoom in and out worksheets, graphics, clip art, tables, header and footer, review, and worksheet translation.

Macros in Excel are written in Visual Basic and it is through running macros that you will use your VB code. You can assign a macro to objects such as Combo boxes or shapes and they can also be run using short cut keys. Broadly speaking there are two reasons to create macros in Excel: To automate repetitive tasks such as formatting data To add functionality to Excel spreadsheets e. Source There are two ways of creating Visual Basic code in Excel: The first is to record a macro. When you perform a task while Excel is recording a macro, for example, turning grid-lines on and off. You can then see the Visual Basic code that Excel uses to perform that task. For a thorough guide on recording macros, click here: I have used Visual Basic code to complete two projects in Excel. The first is a thematic map a chart that shows themes or variations across a geographical area, such as rainfall across the United States This uses shapes that change colour based on the contents of the reference data. Visual Basic code was used to link the reference data to the individual shapes that constitute the overall map. If you would like to learn how to create a thematic map in Excel, I have a hub that goes into creating one in further detail which can be found here: Thematic map created using Visual Basic code in Excel and Excel Source Enabling the Developer tab in Excel and Excel Before we begin to create Visual Basic code and macros, we need to enable the Developer tab. The methods are different depending on what version you are running. If you are using Excel Source For those using Excel Source Working in the Microsoft Visual Basic window in Excel and Excel When working with Visual Basic code, you will be working within modules as shown below. There are two ways of accessing Visual Basic modules: Select the Developer tab and click the View Code button in the Controls group Click on the Macros button in the Code group on the Developer tab and select a macro you want to work on and select Edit Microsoft Visual Basic modules in Excel and Excel Source To create a new module, click on Insert within the Microsoft Visual Basic window and select Module This will open a new blank module To move between modules macros , double click on the a module in the Modules section How to move between Visual Basic modules in Excel and Excel Source Before we begin with writing the code itself; there is one more buttons you should be familiar with on this screen. Source The button highlighted above with the red arrow is the Run button. This allows you to test your code to ensure that there are no errors. Excel will tell you if there is an error in your code and helpfully which line it is on. Basics of writing Visual Basic code in Excel and Excel The first step for all macros is to tell Excel the name of your Macro and that your code is starting. The syntax for this is: You can see from the figure below that the second line in my code is coloured red. Source The red lettering is telling us that there is something missing from that line If when you run it by clicking the Run button, there is still an error in your code, Excel will give you an an error and then highlight the line containing the error in yellow. You can then check that line for the error and fix it. Excel and Excel showing an error exists in the Visual Basic code that was run. While researching this hub, I learnt how to define the active worksheet and also how to change the colour of my shape directly from the help. I had to change the code to suit my needs, but I learnt the basics from the help itself. I found out that I needed to define it because without a defined active worksheet, Excel did not know which A2 I was referring to. The first step is to define a variable we will call it Number to hold the contents of our cell we would like to link to our shape: Dim Number Now we need to tell which Excel which worksheet we want it to work off I have a large number of worksheets, if you only have one you can skip this line. Activate Visual basic is the name of the worksheet I am working off and we have asked Excel to use it as the active worksheet. Now we tell Excel what to put into our variable Number. Value This tells Excel that Number equals the contents of cell A2 and that it is a value Now we come to the heart of our code, telling Excel what to do with the value held in number. If the variable Number which is linked to cell A2 is greater than 50 then do what is on the next line, if not do nothing and go to the next IF statement. To find out what

your shape is called click on it and note down its name which will appear in the box next to the formula box. Shape that Visual Basic code will colour based on the contents of a cell linked to the shape in Excel and Excel Assigning a Macro to shapes or objects in Excel and Excel The final step is to assign our new macro to the shape itself. The reason for doing this is to allow you to run the macro by simply clicking on the shape itself. Select your shape or object Choose Assign Macro In the list of Macros, choose the macro you want to assign to the shape and click OK Now that we have completed our code and assigned it to our shape we need to test it. First enter 20 into cell A2 and click your shape to confirm that nothing happens Next, enter 50 into A2 and click the shape again. It should now be filled red Finally, enter into A2 and click the shape one last time. It should now change to a bright green! Illustration of Visual Basic code colouring the shape based on the contents of cell A2 in Excel and Excel Source Conclusion Visual Basic allows you much more flexibility to do things in Excel that are beyond the tools that are hard coded into it. Your Visual Basic code is available via macros which can run via short cut keys or assigned to a shape or object. First, we looked at the enabling the Developer tab and how to access Microsoft Visual Basic from within Excel Then, we examined the basics of writing Visual Basic code as well as how Excel lets us know if we have errors in our code Next, I showed how to use the excellent help available within Microsoft Visual Basic in Excel to help you build your code After that, we went through line by line building code to change the colour of my shape depending on the contents of cell A2 Finally, we looked at how to associate macros with shapes or objects on your worksheet! I wish you success in writing your own code in Visual Basic to fulfil any needs you may have within your worksheets. Please feel free to leave any comments you may have below. Many thanks for reading!

Chapter 2 : Using Macros in Excel

VBA (Visual Basic for Applications) is the programming language of Excel and other Office programs. 1 Create a Macro: With Excel VBA you can automate tasks in Excel by writing so called macros. In this chapter, learn how to create a simple macro.

Less To automate a repetitive task, you can record a macro with the Macro Recorder in Microsoft Excel. Imagine you have dates in random formats and you want to apply a single format to all of them. A macro can do that for you. You can record a macro applying the format you want, and then replay the macro whenever needed. These steps can include typing text or numbers, clicking cells or commands on the ribbon or on menus, formatting cells, rows, or columns, or even importing data from an external source, say, Microsoft Access. Although VBA gives you the ability to automate processes within and between Office applications, it is not necessary to know VBA code or computer programming if the Macro Recorder does what you want. It is important to know that you when you record a macro, the Macro Recorder captures almost every move you make. So if you make a mistake in your sequence, for example, clicking a button that you did not intend to click, the Macro Recorder will record it. The resolution is to re-record the entire sequence, or modify the VBA code itself. The more smoothly you record a sequence, the more efficiently the macro will run when you play it back. Macros and VBA tools can be found on the Developer tab, which is hidden by default, so the first step is to enable it. For more information, see Show the Developer tab. Record a macro There are a few helpful things you should know about macros: When you record a macro for performing a set of tasks in a range in Excel, the macro will only run on the cells within the range. So if you added an extra row to the range, the macro will not run the process on the new row, but only the cells within the range. If you have planned a long process of tasks to record, plan to have smaller relevant macros instead of having one long macro. It is not necessary that only tasks in Excel can be recorded in a macro. For example, you can record a macro where you first update a table in Excel and then open Outlook to email the table to an email address. Follow these steps to record a macro. On the Developer tab, in the Code group, click Record Macro. In the Macro name box, enter a name for the macro. Make the name as descriptive as possible so you can quickly find it if you create more than one macro. The first character of the macro name must be a letter. Subsequent characters can be letters, numbers, or underscore characters. Spaces cannot be used in a macro name; an underscore character works well as a word separator. If you use a macro name that is also a cell reference, you may get an error message that the macro name is not valid. To assign a keyboard shortcut to run the macro, in the Shortcut key box, type any letter both uppercase or lowercase will work that you want to use. In the Store macro in list, select where you want to store the macro. In the Description box, optionally type a brief description of what the macro does. Although the description field is optional, it is recommended you enter one. Also, try to enter a meaningful description with any information that may be useful to you or other users who will be running the macro. If you create a lot of macros, the description can help you quickly identify which macro does what, otherwise you might have to guess. Click OK to start recording. Perform the actions that you want to record. On the Developer tab, in the Code group, click Stop Recording. Working with recorded macros in Excel In the Developer tab, click Macros to view macros associated to a workbook. This opens the Macro dialog box. Macros cannot be undone.

How to open the VBA environment. You can access the VBA environment in Excel by opening the Microsoft Visual Basic for Applications window.. First, be sure that the Developer tab is visible in the toolbar in Excel.

Share on Facebook Advanced users of Microsoft Excel often use macros to speed their ability to perform computing tasks. A macro is a series of instructions given to the spreadsheet software. Simple macros record your keystrokes or mouse clicks, allowing you to replicate them quickly and easily. Creating a macro in Excel automates repetitive tasks, such as complex cell formatting. You can create and save a series of macros to perform basic activities and improve your work efficiency. Create a macro in Excel to simplify complicated tasks. Step Identify a complex series of commands you frequently give Microsoft Excel. For example, you could create a macro for formatting a group of cells with 8-point, blue Times New Roman font. Step Choose the "File" tab. Click "Options" and "Customize Ribbon. Enable macros under the "Macro Settings" area. Step Click "Record Macro" to open a window and create a new macro in Excel. Step Enter a unique name for your macro. Step Select an area to store the macro by changing the "Store Macro In" field. Create a brief description of the macro so that you can remember its purpose later. Click OK to close the window and begin recording. Step Perform the series of functions you want to record. Perform each mouse click, keystroke or other function in the proper order to ensure that the macro will work properly. Step Click "Stop Recording" when you have finished your series of commands. Microsoft Excel automatically saves the macro when you click this button. Step Test the macro by selecting an area upon which you want the macro to operate. Select the macro and press "Run.

Chapter 4 : Visual Basic for Excel and Excel - A Beginner's Guide | HubPages

Learn how to use the macro recorder in Excel to create simple vba macros without coding. To download the course materials, sample Excel files used in the videos and the handouts for the.

How to create and edit macros is easy and makes you an Excel pro. Macros may seem very intimidating at first glance, but by working through this tutorial and practicing in Microsoft Excel, you will be a pro in no time. You could use the audio recorder or the To Do List function and depending on your phone make, it may look like this: Saving the above To Do List on your phone and using it every time you have a similar event in the above case a party will save you from having to recreate a To Do List every time or even to manually remember the actions to carry out. Similarly, a macro in Excel is just a list of recorded actions that are saved and can then be used to repeat a specified set of instructions. You can record macros for just about any set of actions that can be carried out in Excel. In our example, we will run a macro that has been created to save an Excel worksheet in PDF format. To run a macro that is already created within a workbook: Click on the Developer tab on the Ribbon. If the Developer tab is not visible on your Ribbon, you will have to enable it. To enable the Developer tab, click on File " Options. In the Excel Options dialog box, select Customize the Ribbon. In the right-hand panel, underneath the heading Customize the Ribbon, tick the box for Developer and press OK. In the Code group, click on the Macros button circled in yellow in the screenshot below: This will launch the Macro dialog box listing all the macros that are available in the open workbook. In the screenshot example below, there is one macro named: To run this macro, select it in the list by clicking on it. Next click on the Run button to execute the macro. When clicking the macro pictured in the screenshot above, the worksheet will automatically be saved as a PDF to a specified folder on your PC. It replaces having to carry out the manual actions of saving a worksheet as a PDF into a specified folder. This is one example of a macro. Macros can be used for formatting, formulas, functions, saving worksheets, creating worksheets and just about any other set of actions you perform in Excel. In the Code group, click on Record Macro. This will launch the Record Macro dialog box: Give your macro a name in the Macro Name box circled in yellow in the screenshot below. Ensure the name is easy to understand and tied to the functionality of the macro. Assign a shortcut key to the macro. Be careful to not use shortcut keys that you frequently use for other functions. Select where you wish to store the macro. If you only wish to run the macro on the current workbook, select: In order to help other people using the worksheet or even to act as a reminder for yourself at a later stage, type a description for the macro in the Description box. Take a deep breath before pressing the OK key. Once you press this, you will have to carry out the actions for the macro, as detailed in the script you created in the first step, without stopping. This indicates the macro is busy recording. Carry out all the actions you would normally carry out to perform a task to save as a PDF in our example. Once you have completed the actions, press the Stop Recording button. If your macro was successful, you will now be able to run it by following the instructions listed in the section above, Running a Macro. Well, there is one final action that brings it all together: You could create a whole section of the Ribbon containing only the functions you created. Imagine showing your friends or colleagues these customized functions and watching them search in vain to find it on the Ribbon of their own copy of Excel. To assign a macro to a button: Open the worksheet containing the macro you wish to create a custom button for. Click on the File tab to launch the Backstage view and then select Options to launch the Excel Options dialog box. In the Excel Options dialog box, click on Customize the Ribbon. In the right-hand side panel, click on New Tab circled in yellow in the screenshot below: Give the tab a logical name, for example, My Functions, by selecting the new tab you created and clicking on Rename at the bottom of the panel. You can also right-click the new tab you have created and select Rename from the menu list. Under the tab you created a new group will automatically be added. Rename this new group by giving it a name that fits the type of macros you will be adding to it. Next, you need to add the macro you created to the Tab and Group. In the Choose Commands From box, select Macros see screenshot below: Scroll through the Macros list and select the macro you created. In the Tabs list on the right-hand side panel, ensure the group under the tab you created, is selected. In our example this is the My Function tab and the Saving Functions

group. Click on the Add button. The macro name will now appear in the selected group. Finally you can customize the macro by creating a button icon for it. Select the Macro you created and click on Rename. You can then select an icon from the icon gallery to add to your Macro. Click OK to accept all the changes. Depending on the name and icon you chose, the customized Ribbon with your macro will look something like this: Clicking the button you created for the Macro will now launch the macro as if it is a built-in function in Excel. The final ingredient in the world of macros is the type of file that is capable of containing a macro and macro security settings that could prevent your macro from functioning. To save a workbook that contains a macro: Click on File and then select Save As. Select where you wish to save the workbook and give it a file name. Press OK to save the workbook. An Excel workbook that contains macros display as an icon with an exclamation mark: When opening a worksheet that contains macros, Excel will by default display a Security Warning at the top of the worksheet and prompt you to enable macros. If the file you are opening is a workbook you created or originates from a safe source, it is okay to click on Enable Content and utilize the worksheet as usual. It is good practice to keep a tight control over the number of macros and their functions in a worksheet. To delete a macro that is no longer in use: Open the worksheet that contains the macro. In the Code group, click on the Macros button. This will open the Macros dialog box. In the Macros dialog box, click on the macro you wish to delete and click the delete button. Once deleted, you cannot reinstate a macro and would have to recreate it. To launch the Macros dialog box, use the shortcut combination: Now you have done the tutorial

Chapter 5 : VBA Excel Macros

Java Project For Beginners Step By Step Using NetBeans And MySQL Database In One Video [With Code] - Duration: 1BestCsharp blog , views.

What is a macro? A macro is a piece of programming code that runs in Excel environment and helps automate routine tasks. Some of the customers pay through the bank and at the end of the day, you are required to download the data from the bank and format it in a format that meets your business requirements. You can import the data into Excel and format. The following day you will be required to perform the same ritual. It will soon become boring and tedious. Macros solve such problems by automating such routine tasks. You can use a macro to record the steps of Importing the data Formatting it to meet your business reporting requirements. It is a programming language that Excel uses to record your steps as you perform routine tasks. You do not need to be a programmer or a very technical person to enjoy the benefits of macros in Excel. Excel has features that automatically generated the source code for you. Read the article on Vba for more details. Macro Basics Macros are one of the developer features. By default, the tab for developers is not displayed in excel. You will need to display it via customize report Macros can be used to compromise your system by attackers. By default, they are disabled in excel. Always fill in the description of the macro when creating one. This will help you and others to understand what the macro is doing. Step by step example of recording macros in Excel We will work with the scenario described in the importance of macros excel. We will work with the following CSV file. You can download the above file here We will create a macro enabled template that will import the above data and format it to meet our business reporting requirements. Enable the developer option as shown below and pin it into your main ribbon in Excel. Create a folder in drive C named Bank Receipts Paste the receipts. Step two will be there by default Enter the description as shown in the above diagram Click on "OK" tab Put the cursor in cell A1 Click on the DATA tab Click on From Text button on the Get External data ribbon bar You will get the following dialogue window Go to the local drive where you have stored the CSV file Select the CSV file You will get the following wizard Click on Next button after following the above steps Follow the above steps and click on next button Click on Finish button Your workbook should now look as follows Make the columns bold, add the grand total and use the SUM function to get the total amount. Now that we have finished our routine work, we can click on stop recording macro button as shown in the image below Before we save our work book, we will need to delete the imported data. We will do this to create a template that we will be copying every time we have new receipts and want to run the ImportBankReceipts macro. Highlight all the imported data Right click on the highlighted data Click on Delete Click on save as button Save the workbook in a macro enabled format as shown below Make a copy of the newly saved template Open it.

Chapter 6 : How To Create A Simple Macro In Excel

By Greg Harvey. After you create an Excel macro, either by using the macro recorder or by creating it in Visual Basic for Applications (VBA), you run the macro to have it carry out the commands and keystrokes that are saved as part of the macro.

Chapter 7 : Free Excel Tutorial at GCFGlobal

With Excel VBA you can automate tasks in Excel by writing so called macros. In this chapter, learn how to create a simple macro which will be executed after clicking on a command button. First, turn on the Developer tab. To turn on the Developer tab, execute the following steps. 1. Right click.

Chapter 8 : Free Excel Tutorial - Using Macros - Excel

Open Excel. The process for enabling macros is the same for Excel , , and There is a slight difference for Excel for Mac, which will be detailed below.

Chapter 9 : Enable VBA/Macro's in Excel - Ingmar Verheij

VBA is the acronym for Visual Basic for Applications. It is a programming language that Excel uses to record your steps as you perform routine tasks. You do not need to be a programmer or a very technical person to enjoy the benefits of macros in Excel.