Microsoft Office Training Series

Microsoft Application Series

Excel Advanced Post-Course

- Courses never Cancelled
- 12+ Months Schedule
- 24 Months Online Support
- UK Wide Delivery

STL MicrosoftTraining.net

Learning & Performance Institute
Accredited Learning Provider

ilm Recognised Provider

Microsoft Partner
Certified Silver Partner

CONTENTS

Module One: Using Logical Functions

In this module we will look at defining named ranges and using them in formulas. Identifying and applying Excel's logical functions, IF, AND. OR and creating nested functions by combining logical functions in a formula.

Working with Names & Ranges

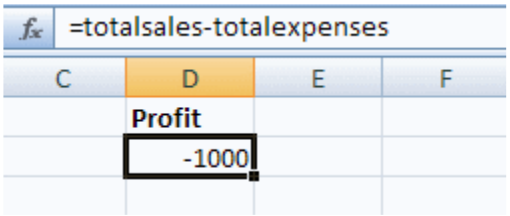
Working with numbers isn't always easy. A complex formula involving several cell ranges can be difficult to understand. Individual cells that contain important data can be hard to find on a large worksheet. Cell references like D5:D22 or A33:C33 are somewhat abstract, and don't really communicate anything about the data they contain.

In Excel, you can create meaningful names for cells or ranges that can be used to overcome these difficulties.

What Are Range Names?

Range names are meaningful character strings that you can assign to individual cells or cell ranges. You can use a range name practically any where you can use a cell or range reference.

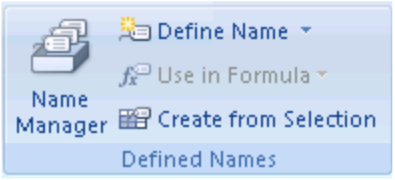
The advantage of using names comes from the fact that a name, like Employees, is more meaningful and less abstract than a reference like C2:C55. Also, named ranges are by default absolute, so if you copy or AutoFill a formula using named ranges, it will maintain its original cell references.



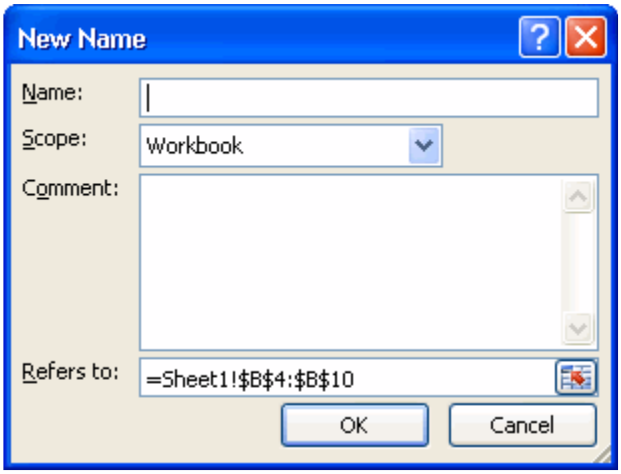
Range names will make formulas much more readable and they will make it easier to find and reference individual cells. When you are first designing your worksheet, you can create formulas using names instead of traditional cell references, and then define the names for the corresponding ranges of data as required. Basically, using range names in your worksheet improves clarity, improves organization, and aids in the overall design.

Defining and Using Range Names

To define a range **name**, select either a cell or cell range and choose the Define Name button from the Defined Names button group on the Formulas Ribbon.

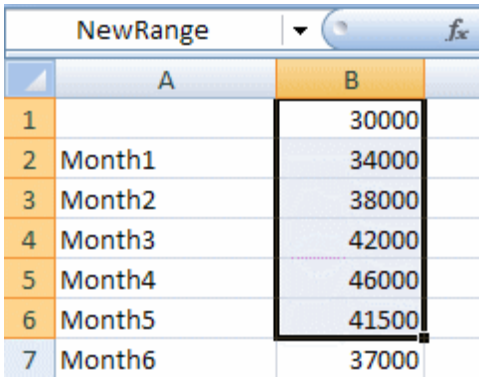


This will display the New Name dialogue box.



In the New Name box, you will see the reference to the cell or range you selected in the bottom text field. This is the reference that will be associated with the name you choose. To name your range, type a name in the top text field and click OK. (If you wish, you can also add a comment to be associated with your new named range.) The Scope refers to the parts of the workbook where your named range will be valid.

Another way to name a cell or range is to select it, type the name in the name box on the formula bar, and press Enter.



In this example, the cells B1 to B6 were selected and the name NewRange was entered into the name box. (If you click the down pointing arrow just to the right of the name box, you will display a list of range names used in the spreadsheet.)

Excel will not accept spaces between words in the names you choose. For example, "newrange" or "newRange" would be acceptable, but "New Range" would not. Once you have defined your named ranges, you can use them in formulas and functions just as you would a regular cell or range reference.

As an example, if you named a range of figures Sales, and you named another range of data Expenses, you could calculate the total sales or total expenses by entering the function =SUM(sales) or =SUM(expenses) respectively.

Using names for ranges and cells in this way makes your formulas and functions much clearer. When you want to enter a range in a formula or function, it is much easier to remember and type the name of the range, rather than specific cell references.

For example, = (TotalSales-TotalExpenses) is a more meaningful formula than = (B2-C2), Similarly, =AVERAGE(Height) is more meaningful than =AVERAGE(B2:B100).

Using Logical Functions

Excel's logical functions are:

AND

OR

IF

NOT

FALSE

TRUE

IFERROR

These logical functions are important when doing advanced work in Excel because they can help you control the behavior of your worksheets based on specific logical conditions.

This module will focus on the use of the logical functions IF, AND and OR.

The IF function

Excel's IF function can often prove to be very useful. You can use this function to branch to different values or actions depending on a specified condition. The structure of an If function is as follows: IF (logical test, value if true, value if false)

IF functions are called conditional functions because the value that the function returns will depend on whether or not a specific condition is satisfied. As an example, consider the following function: IF (A1=10, 5, 1)

This function states that if cell A1 has a value of 10 the cell that contains the function will have the value of 5. But if A1 doesn't have a value of 10, the cell that contains the function will have a value of 1. In other words, the function reads: if A1 equals 10 then return the number 5, else, return the number 1. Let's say that this next IF function is entered into cell B2: IF (A1<=100, A1*.5, C3*2)

This function states that if the contents of cell A1 is less than or equal to 100, the value in cell B2 will be the value in A1 multiplied by .5; else, the value in B2 will be the value of cell C3 multiplied by 2.

You can insert an IF function by invoking the Insert Function dialogue and looking under the Logical category, or by typing it directly into the formula bar.

The logic of the IF function can be a little confusing until you get used to it. The best way to get comfortable with IF functions, is to practice using them.

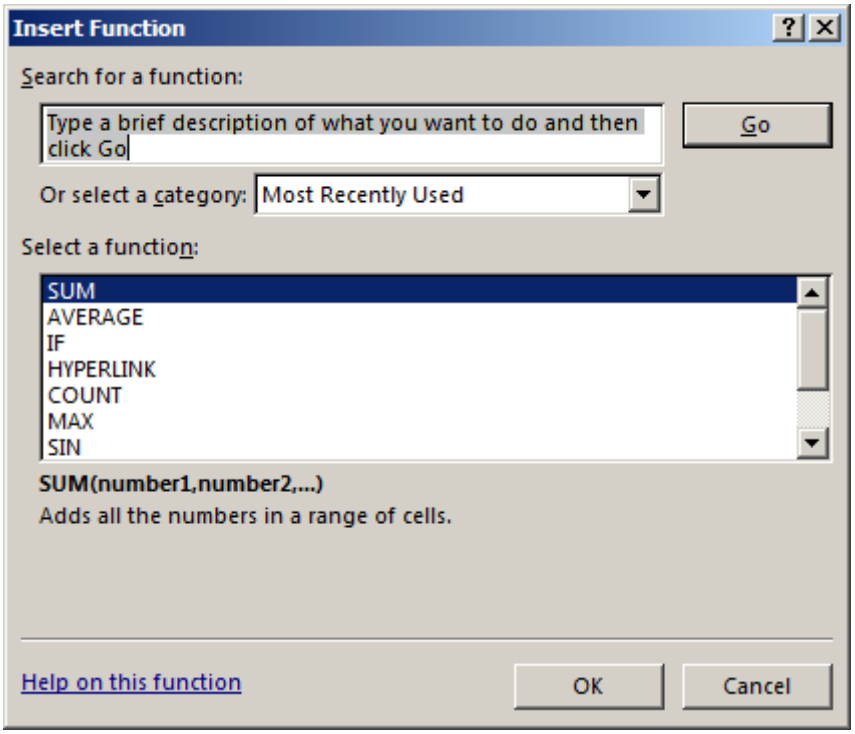
Using the Function Library

Excel 2010 contains an extensive library of functions that you can call upon to help you solve problems. These tools are available in the Function Library button group, on the Formulas ribbon.

The first and largest button is Insert Function. This button will open a dialog allowing you to search for and insert hundreds of functions.

You can also click the small fx button next to the formula bar to display the Insert Function box.

Clicking the Insert Function button activates the Insert Function dialogue box and provides access to the large range of functions available in Excel.



Once the Insert Function dialogue box is open:

- Select the function you wish to use from the available list and click OK or
- Type the name of the function you wish to use in the Search for a function area, press

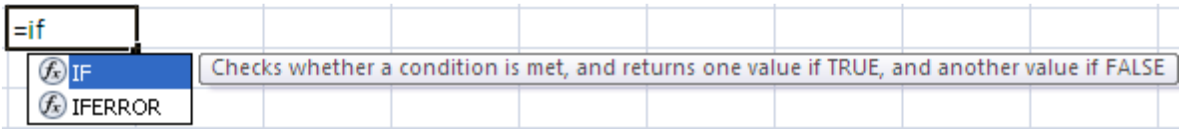
Enter, select the function when it appears in the list and click OK.

Manually entering a function

If you know which function you wish to use, you can enter a function into a worksheet by inputting it manually (i.e. by typing the function directly into a cell).

When you do this in Excel 2010, a screen tip will appear with the possible functions that correspond with the letters of the function name you have entered.

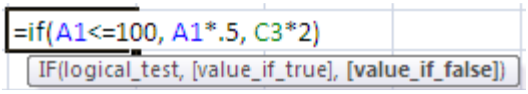
For example, if you type =IF into a cell the following appears.



If there is more than one function listed, you can toggle between them by using the arrow up or arrow down keys.

Once you have selected the function you wish to use, continue by typing in a left bracket, which will cause the function syntax to display in another screen tip.

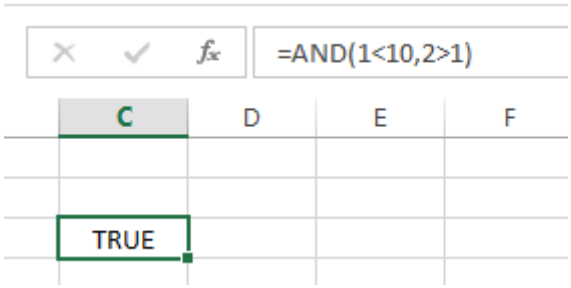
Continue to enter all the arguments required for the function to give you a result, ensuring you type a comma between each argument.



Press Enter to display the result.

The AND Function

The AND function will return true if all of its arguments are true. If one argument is false, the AND function will return false. For example, the logical statement 1<10 is true and the statement 2>1 is also true. As a result, the compound statement 1<10 AND 2>1 is true, because both of the statements that are being joined by AND are true. The following image demonstrates this in Excel.



Notice that the arguments joined by AND are placed in parenthesis and separated by commas. The Excel function =AND(1<10, 2>1) means the exact same thing as saying 1<10 AND 2>1 in plain English. Because 1 is less than 10 and 2 is greater than 1, the Excel AND function returns the logical value True.

You can have as many as 255 logical arguments to an Excel AND function. You can enter these arguments directly, or use cell references as the situation requires. (All of the arguments should be separated by commas.)

The following table (often referred to as a truth table) may help you understand the AND function. In the last column of the truth table, you will see the value returned by the function, according to the corresponding values of the function arguments.

A	B	AND (a, b)
True	False	False
False	True	False
True	True	True
False	False	False

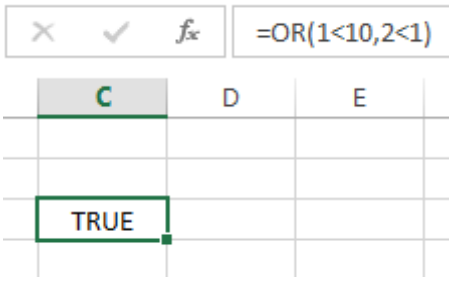
The OR Function

The logical OR function will return true if one or more of the arguments to the function are true. For instance, the compound statement 1<10 OR 2<1 is true, since 1 is less than 10. It does not matter that the statement 2<1 is false, because you have a choice of either 1<10 OR 2<1 to make the compound statement true. If one or both of the arguments is true, the overall OR statement will also be true.

In Excel, you compose an OR function like this:

=OR(argument1, argument2, argument3, argument4,…….argument255)

If one or more of the function arguments is true, the OR function will return true.



Notice the function and its comma separated arguments in the formula bar.

Here is a truth table for the OR function.

A	B	OR (a, b)
True	False	True
False	True	True
True	True	True
False	False	False

Working with Nested Functions

In Excel, you can actually place (or nest) a function within a function.

=AVERAGE(SUM(B2:B6),SUM(C2:C6),SUM(AVERAGE(B2:B6),AVERAGE(C2:C6),

AVERAGE(D2:D6)))

Notice that this function has 3 average functions nested within a sum function, which is in turn nested in another average function. This may seem confusing, but if you carefully step through the function from right to left, it becomes clear.

The average of range D2:D6, the average of C2:C6, and the average of B2:B6, are summed together. This sum is then averaged with the sum of C2:C6 and the sum of B2:B6 for a final result.

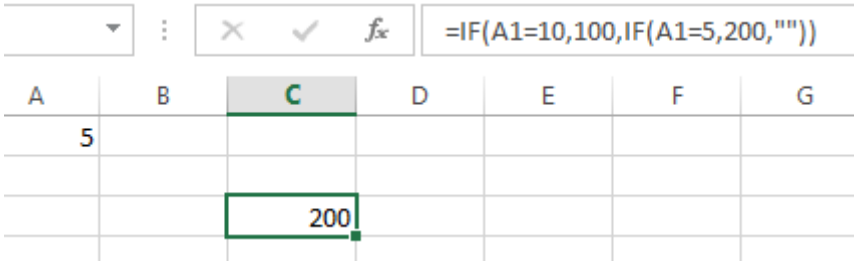
In terms of nested functions, nesting averages within sums and sums within averages is probably not that practical; however, nested IF functions can be extremely useful for a wide variety of situations.

Nested IF functions

The IF function is ideal for making choices based on logical tests. Furthermore, you can nest IF functions one inside another.

=IF(A1=10,100,IF(A1=5,200,””))

In the case of this IF function, if the value in cell A1 is 10 the function will return 100. If the value in A1 is not 10, the function will test if the value in A1 is 5. If the value in A1 is 5, the function will return 200. If it is not 5 (and also not 100) the function will display a blank cell.



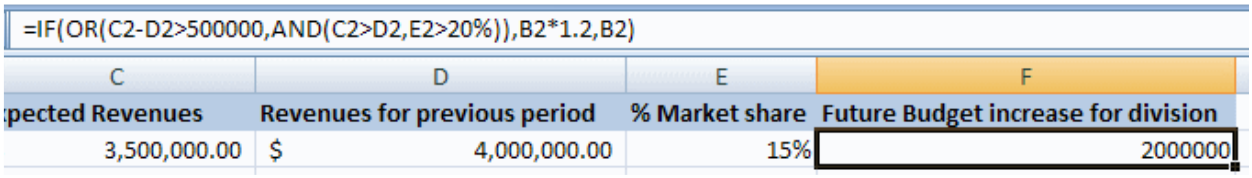
When you nest logical functions, you must make sure that the number of closing parenthesis matches the number of opening parenthesis used in the function. If you count the parenthesis in the function from the image above, you will see two opening, and two closing parenthesis.

Applying Logical Functions

	A	B	C	D	E	F
1	Division	Budget	Expected Revenues	Revenues for previous period	% Market share	Future Budget increase for division
2	New York	$ 2,000,000.00	$ 3,500,000.00	$ 4,000,000.00	15%	2000000
3	L.A.	$ 2,000,000.00	$ 4,500,000.00	$ 4,200,000.00	18%	2000000
4	Chicago	$ 2,000,000.00	$ 5,600,000.00	$ 5,578,000.00	25%	2400000
5	Boston	$ 2,000,000.00	$ 3,478,000.00	$ 3,200,000.00	14%	2000000
6	Miami	$ 1,500,000.00	$ 4,000,000.00	$ 1,960,000.00	11%	1800000

The spreadsheet in the image shown above is used to calculate the future budget increase for different divisions of a fictional company. If you look carefully, you will notice that the Miami division has had its budget increased from 1,500,000 to 1,800,000. Also, the Chicago division has had its budget increased from 2,000,000 to 2,400,000. The question that arises here is, "What reasoning was used to arrive at these budget increases for these particular cities?

For the answer to this question, look at the formula bar when a cell from the Future Budget Increase for Division column is made active.



When cell F2 is the active cell, you can see the formula from cell F2 in the formula bar.

=IF(OR(C2-D2>500000,AND(C2>D2,E2>20%)),B2*1.2,B2)

To understand this formula as a whole, you must first understand how each logical function is used in the formula. To start, remember that the IF function will return a certain value based on a logical test.

=IF(logical test ,value if true, value if false)

In this case, the logical test is:

OR(C2-D2>500000,AND(C2>D2,E2>20%))

The OR function will return true if one or more of its arguments are true. The first argument in the OR function is C2-D2>500000. This is a simple argument that will return true when the value of cell D2 subtracted from cell C2 is greater than 500,000.

The second argument AND(C2>D2,E2>20%) is a logical AND function. This function will return true only when both of its arguments are true. That is, the AND function

will return true only when the value in C2 is greater than the value in D2, and the value in E2 is grater than 20%.

If the first argument to the OR function is false, the AND function must return true for the OR function to be true. Alternatively, if the AND function returns false, the first argument to the OR function must return true for the OR function itself to return true.

If we use the column headings instead of specific cell references, the logic of this OR statement reads:

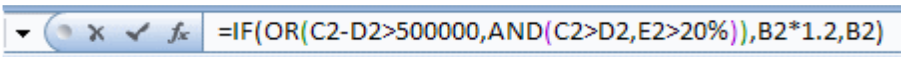
If (%Market Share is greater than 20% and Expected Revenues are greater than Revenues for Previous period) or (Expected Revenues subtract Revenues

Remember, the original format of the logical IF function is:

=IF(logical test ,value if true, value if false)

When the OR function (the logical test for the IF function) returns true, the IF function will give the value B2*1.2 (the original budget increased by 20 percent).

When the OR function returns false, the IF function will give the value B2 (the original budget value).



If you wanted to summarize the formula shown in the formula bar in plain English, you could say the following:

> If the expected revenues show more than a 500000 dollar increase over the previous revenues, or the expected revenues are greater than the previous revenues and the market share is greater than 20

Logical functions can be difficult to grasp at first, especially if they are nested into a larger formula. If you are confused by a formula involving one or more logical functions, take your time and carefully study the function arguments. Study one function at a time until you understand the logic, test conditions, and the value or values that the function will return. If you understand each individual function in a formula, pretty soon the entire formula will make sense to you.

Module Two: Working with Lookups

This module introduces the use of the VLOOKUP and HLOOKUP function.

Excel 2010 provides two lookup functions that you can use to quickly retrieve information from a table. The functions are called HLOOKUP (horizontal lookup) and VLOOKUP (vertical lookup).

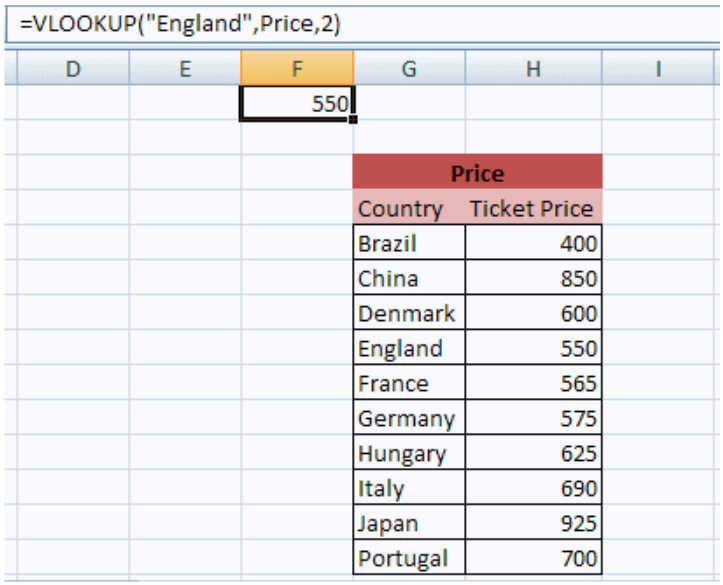
The VLOOKUP function will look in the leftmost column of a table for a value you specify. When it finds the value you specified, it will return a value that is located in the same row, a specified number of columns into the table. It is called VLOOKUP because it looks vertically down a column for a match, and then retrieves data from across the row.

HLOOKUP is similar, but it will look horizontally across the upper row of your table, and then retrieve data from down a column.

Using VLOOKUP to Find Data

The best way to learn how lookup functions work is to look at an example. Here we have a table of ticket prices for flights to different countries. To simplify matters, the data range for the table has been given a defined name (Price) that can be used in functions and formulas.

If we activate a cell F1 and enter =VLOOKUP("England",Price,2) into the formula bar, cell F1 will show the value 550.



Price	
Country	Ticket Price
Brazil	400
China	850
Denmark	600
England	550
France	565
Germany	575
Hungary	625
Italy	690
Japan	925
Portugal	700

The lookup function looked vertically down the leftmost column of the lookup table (Price) until it found a match for the text string "England." The function then returned the value that is in the second (2) column of the table, from the row where the match was found. You should notice that England, Price, and 2 are the exact arguments used in the function.

The arguments for the lookup function are: VLOOKUP(value to match, lookup table name or range, number of the column in the table containing the relevant data, true or false).

For the example shown above, the true or false argument was left out. The relevance of the true or false argument in the VLOOKUP function will be discussed shortly.

HLOOKUP is the same as VLOOKUP, except that it looks across rows for a match rather than down columns. To use HLOOKUP, the lookup table would be arranged in this way.

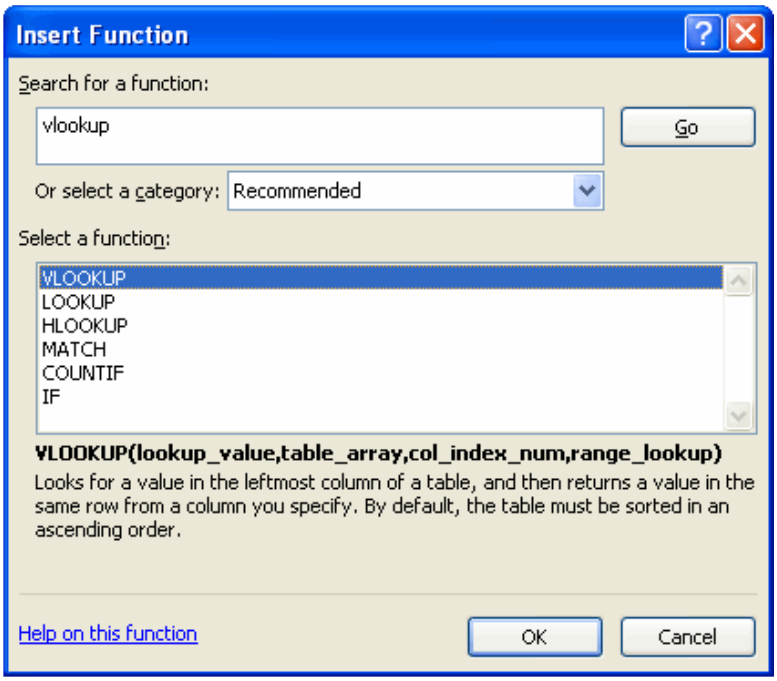
Brasil	China	Denmark	England	France	Germany	Hungary	Italy	Japan
400	850	600	550	565	575	625	690	925

Since Excel is designed with more cells in the vertical direction than in the horizontal direction, and because vertical table design is more intuitive for most people, VLOOKUP is generally used more often than HLOOKUP.

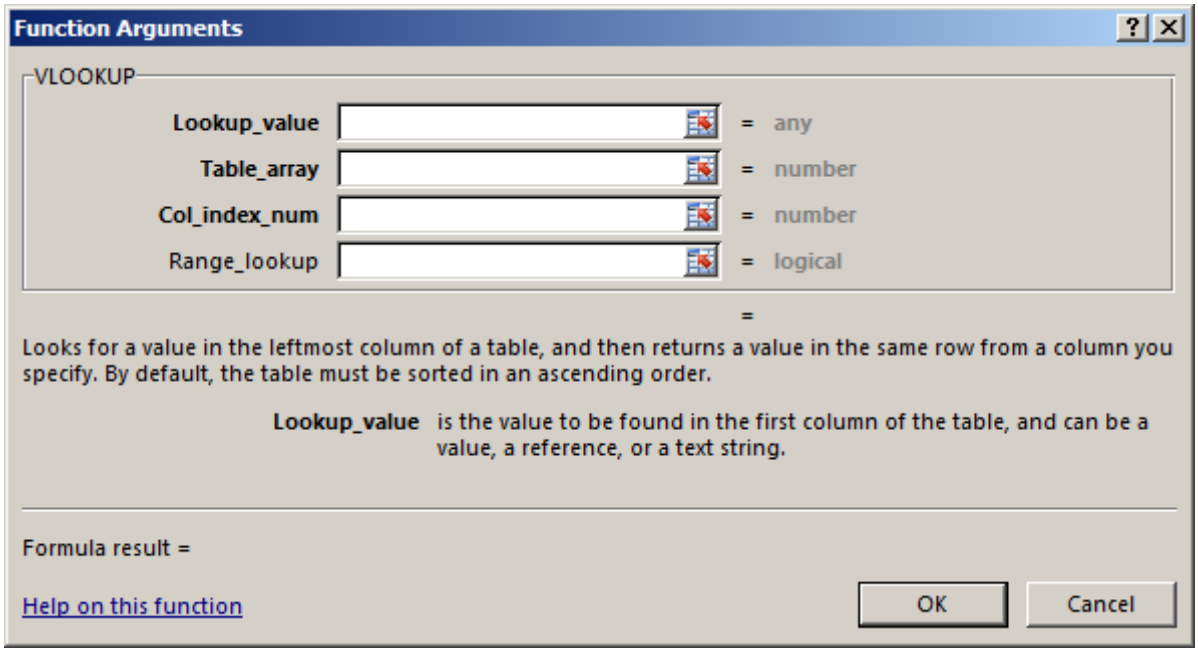
To use the VLOOKUP function correctly, you need to have your spreadsheet data laid out properly in a table with at least two columns. The first column in the table will contain the keys (identifiers that the VLOOKUP function will look through for a match). In the example just shown, the keys (or identifiers) are the names of the countries. This first column can be referred to as the look up column.

The other columns in your table will contain data that is associated with the column of identifiers. Your table can be several columns wide, and you can specify which column VLOOKUP will return data from by putting a number corresponding to the given column in the function. In the previous example, we wanted VLOOKUP to return the ticket price, so we used the number 2 (for the second column) as an argument in the function.
If your table has 10 columns and you want to return data from the ninth column, you would use 9 as an argument.

You do not have to use text values (like the country names used here) in your lookup column. If it is more appropriate, you can use numbers or dates. If you want some help when you are using VLOOKUP, use the Insert Function dialogue by clicking the fx button next to the formula bar.



You will find the VLOOKUP function in the Lookup & Reference category. If you click the OK button in the Insert Function dialogue, you will see the helpful Function Arguments box.

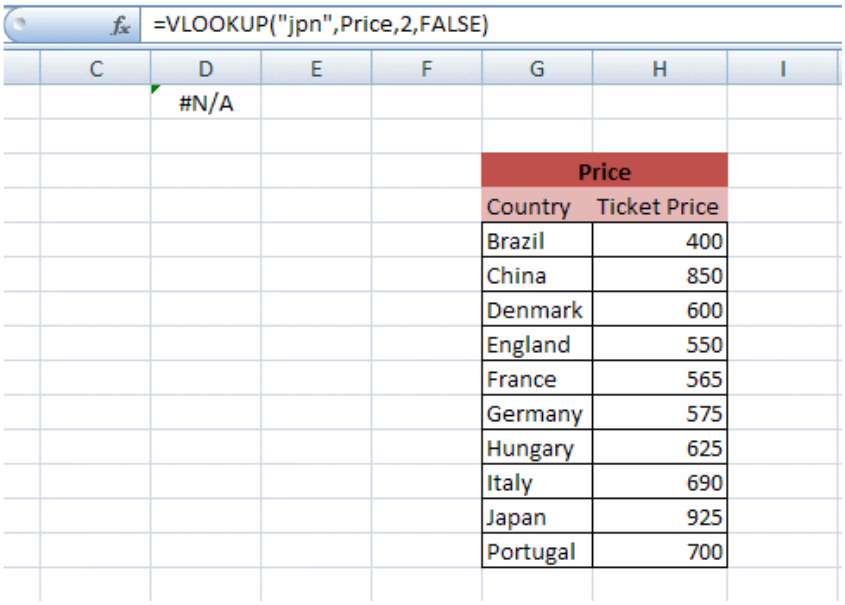


Here, you can enter the function arguments in the fields provided by following the instructions and descriptions that correspond to the given fields.

How to Find an Exact Match with VLOOKUP

As mentioned briefly before, the VLOOKUP function can take a final argument of either true or false. If you specify this final argument as false, VLOOKUP will search for an exact match in the lookup column (column of identifiers). If it cannot find an exact match, no data will be returned.

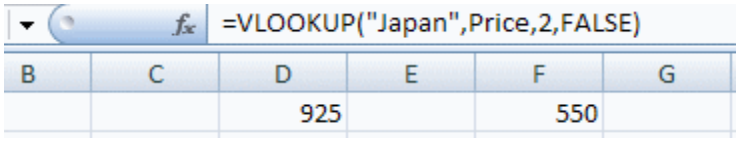
In the following example, the function =VLOOKUP("jpn",Price,2,FALSE) has been entered in cell D1. This means that VLOOKUP will search the first column of the table for an exact match of "jpn," and if it is found, it will return the corresponding value in column 2.



Price	
Country	Ticket Price
Brazil	400
China	850
Denmark	600
England	550
France	565
Germany	575
Hungary	625
Italy	690
Japan	925
Portugal	700

Because there is no country that exactly matches "jpn" no value will be returned to cell D1.

If you change the function so that the first argument is "Japan," an exact match will be found and the value 925 will be returned to cell D1. If there are two or more exact matches, the first matching value found in the column of identifiers will be returned.



The VLOOKUP function's exact match is not case sensitive, so jaPan is considered an exact match of Japan. You can also use wild card characters so that Ja?an and J* will match Japan as well.

Finding the Closest Match with VLOOKUP

If the final argument in your VLOOKUP function is true, VLOOKUP will search the column of identifiers for the closest match to the search value you enter as the first function argument.

For example, in you enter a search value, and you enter true as your final argument, a value associated with the first exact match of your search value will be returned (if one is found).

However, if no exact match is found, a value corresponding to what Excel deems is the closest match will be returned.

Following our example, if you enter "jpn" as the search value for the table of ticket prices, the value associated with the country Japan (925) will be returned, as Japan is the closest match to jpn that is found in the look up column.

If you omit the final argument in your VLOOKUP function, it will default to true, and search for a closest value if it cannot find an exact match.

If you are using the True argument, the values in the look up column should be sorted in ascending order. Otherwise, you may get unexpected results. If you are using text values in your lookup column, avoid using leading or trailing spaces, as this also may produce unexpected results. However, if you use 'False' (for an exact match) as the final argument in your function, the lookup column does not have to be sorted.

Module Three: Advanced List Management

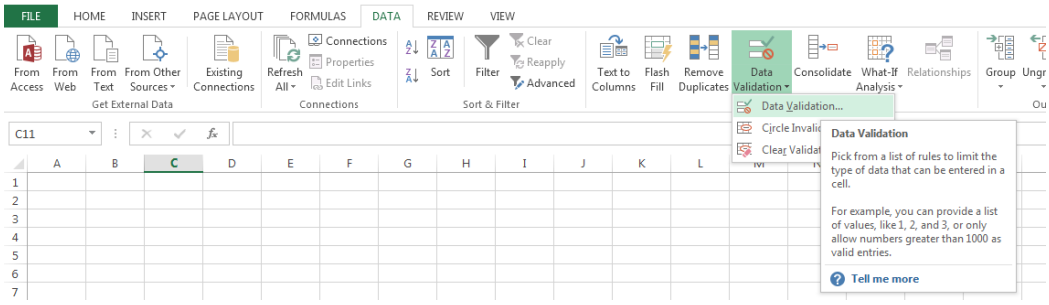
This module will explore applying data validation settings to your worksheet to control data entry. You will also identify and use database functions to perform calculations using multiple database fields.

Creating Data Validation Rules

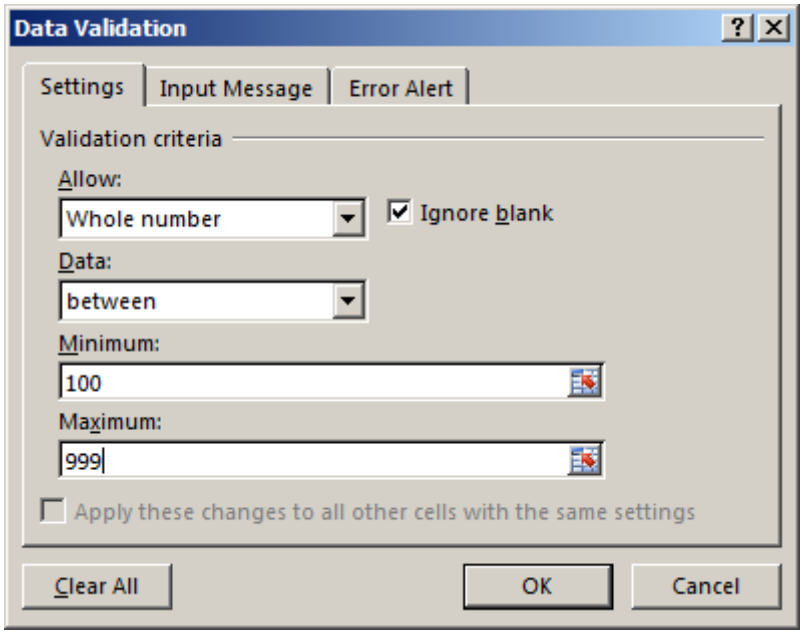
Data validation allows you to create restrictions on what can be entered into a cell. You can have Excel display a warning message or prevent invalid entry if a user of the worksheet does not enter the correct type of value.

Use the following procedure to create a data validation rule.

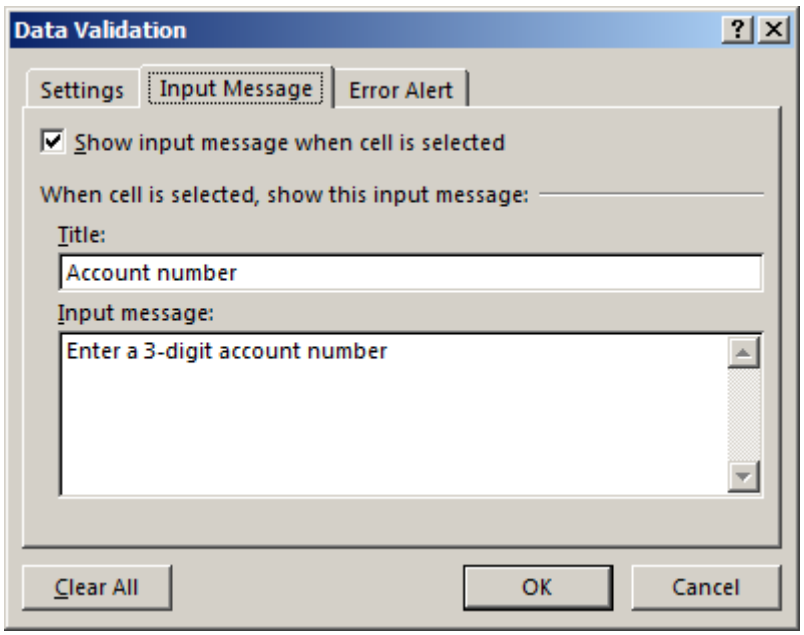
1. Select the cells where you want to apply the data validation rule.
2. Select the **Data** tab from the Ribbon.
3. Select **Data Validation**. Select **Data Validation**.



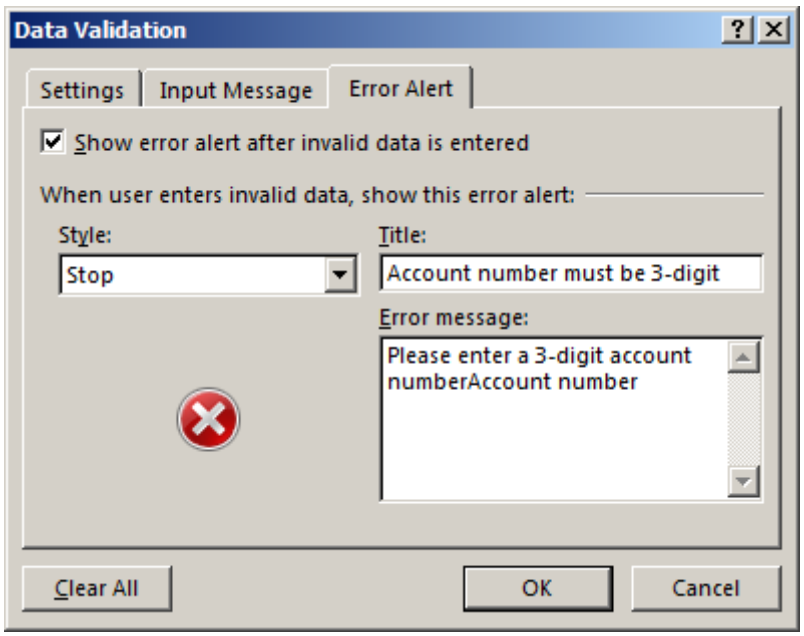
4. On the **Settings** tab of the *Data Validation* dialog box, set up the **Validation Criteria**. Use the drop down lists to help you build your criteria. In this example, we are requiring a three-digit number.



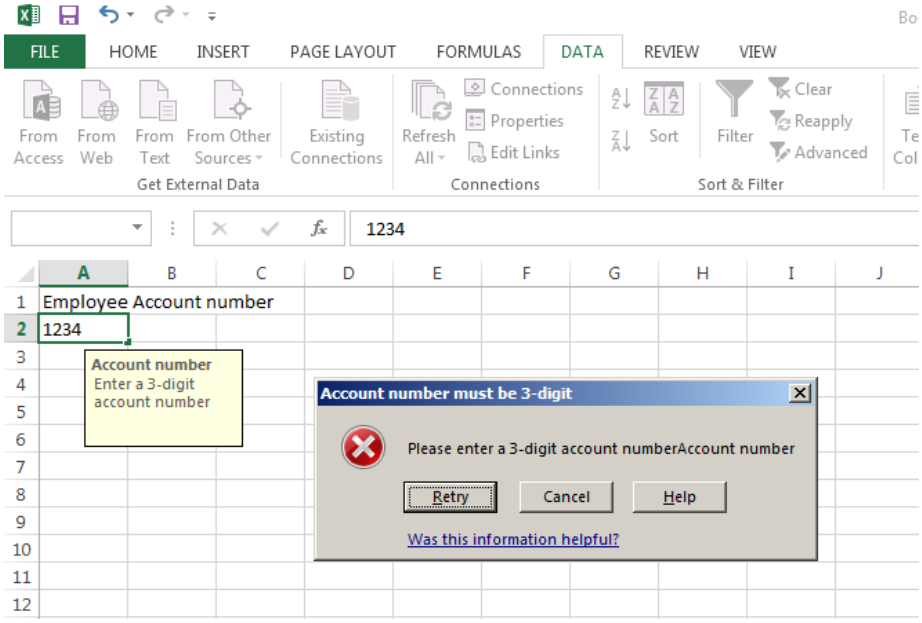
5. Select the **Input Message** tab.



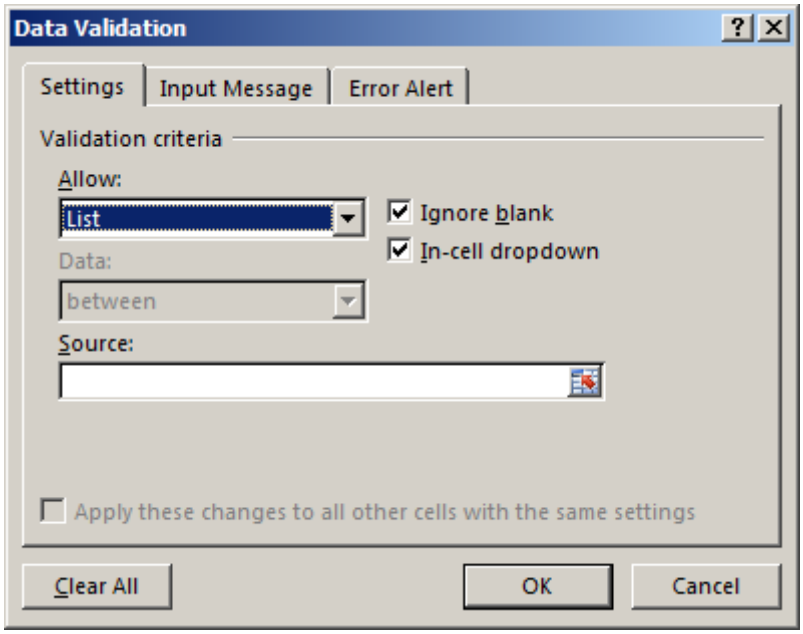
1. Enter a **Title** and **Message** that the user will see when he or she selects the cell.
2. Select the **Error Alert** tab.



3. Select the **Style** of error from the drop down list. Enter a **Title** and **Error** message to display if the user enters invalid data.
4. Select **OK**.

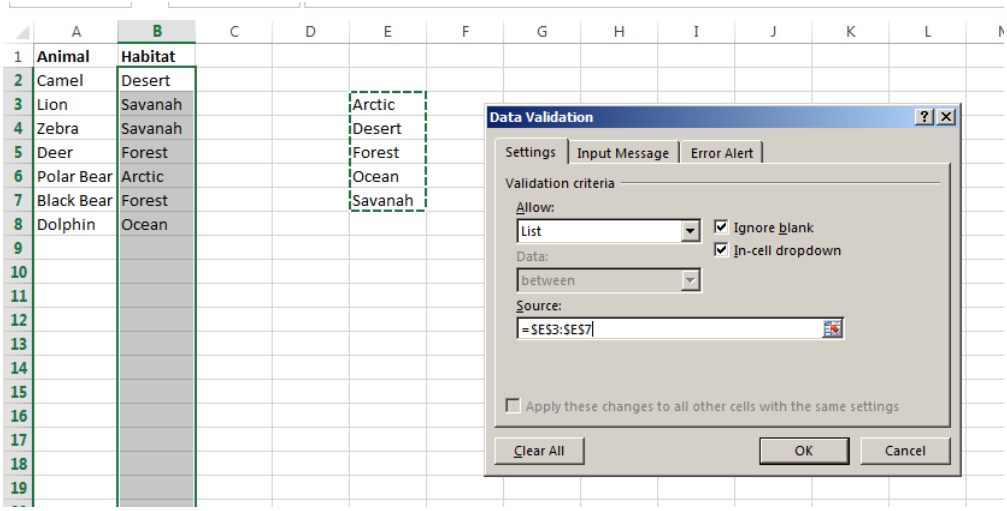


If you select the Settings tab and choose List from the Allow drop menu, you will see a data box (labeled Source) where you can enter permissible source data for the cell.

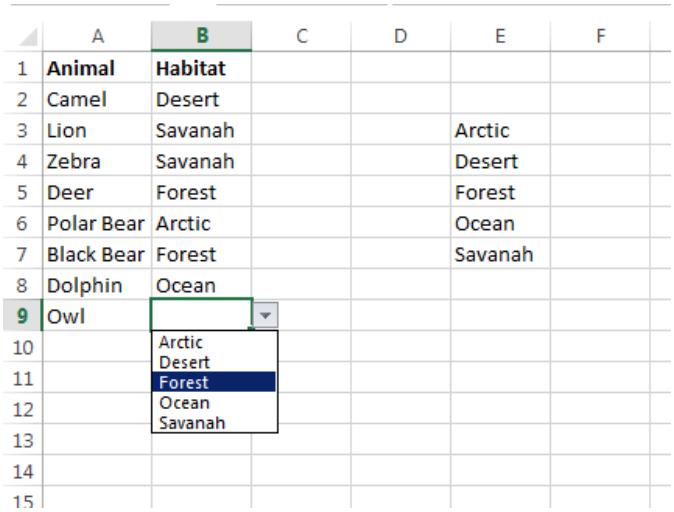


You can type the values that are permissible for the cell directly, or specify a range that contains the values.

If you specify a cell range, make sure there is an equals sign = in front of it.



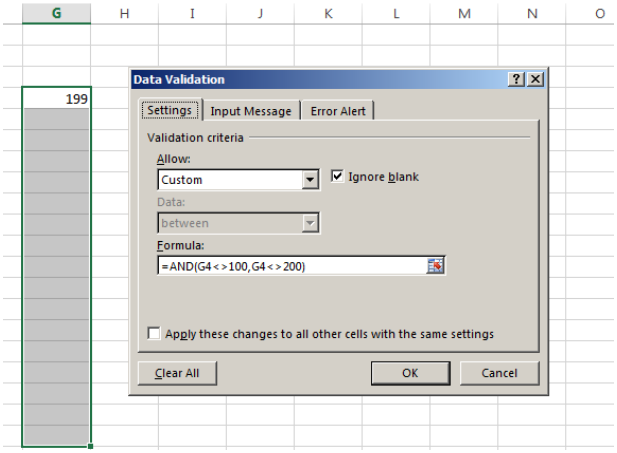
In this image, the values for the validation list is in the range E3:E7cell. But it might be preferable to place the list on a separate worksheet.



Now, a user chooses a type of habitat from the validation drop list.

Using a Formula to Validate Entries

If you select Custom from the Allow drop menu, you can enter a formula to validate your data. You can base the formula on data from another cell if you wish.



Here the custom type has been chosen from the Allow drop list and a formula containing an AND function has been entered. The formula is:

=AND(G4<>100,G4<>200)

This means that if the value entered into cell G4 (or any cell in the validated range G4:G20) the formula returns True If the value is not 100 or 200 and False if it is 100 or 200. Basically, this formula allows any number other than 100 or 200 to be entered.

There are many possibilities for setting data validation rules. For example, you can select a range of cells, click within the range, and click the Data Validation button on

the Data Ribbon to set up validation rules for the entire range. You can also AutoFill a cell with validation rules into other cells in a row or column. These cells will all have the same validation properties.

Using Database Functions

Database functions allow you to perform operations on an Excel database that involve multiple fields. In a sense, they offer some of the power of array formulas, but generally make worksheets faster because they do not require the same amount of recalculation.

With database functions you can get counts, averages, or sums of your database on selection criteria involving multiple fields. Implementing database functions is a little like implementing advanced filters: you have to establish a criteria range first before you use the function itself.

Some useful database functions are:

- **DSUM** Used to create total values in a field based on specified criteria
- **DAVERAGE** Used to average values in a field based on specified criteria
- **DCOUNT** Counts the cells that contain numbers and meet the specified criteria
- **DMAX** and **DMIN** Return the largest and smallest values respectively from records that meet the specified conditions.
- **DPRODUCT** Multiplies values in a field according to specific conditions
- **DGET** Returns a single record value from a record that meets the specified conditions.

All of the database functions use the same argument format:

Function Name (Database range, Field to be returned or calculated, Criteria range).

Take the following Excel table as an example.

	A	B	C	D
1	Day	Sales	Expenses	Profit
2	Mon	2200	400	1800
3	Tue	2200	400	1800
4	Wed	2200	400	1800
5	Thu	2200	400	1800
6	Fri	2200	400	1800
7	Week1	11000	2000	9000
8	Mon	2100	450	1650
9	Tue	2133	450	1683
10	Wed	2166	450	1716
11	Thu	2199	450	1749
12	Fri	2232	450	1782
13	Week2	10830	2250	8580
14	Mon	2345	500	1845
15	Tue	2344	500	1844
16	Wed	2343	500	1843
17	Thu	2342	500	1842
18	Fri	2343	500	1843
19	Week3	11717	2500	9217
20	Mon	1200	500	700
21	Tue	2200	500	1700
22	Wed	3200	500	2700
23	Thu	4200	500	3700
24	Fri	5200	500	4700
25	Week4	16000	2500	13500

We will use the DSUM database function to calculate the total profit for all Tuesdays and Thursdays with Sales greater than 2200 and Profit less than 1900.

Creating a Criteria Range

The first step in using a database function is to set up a criteria range. We do this by entering the field headings that we want to use exactly as they are in the original data list. For this example, we will need the fields Day, Sales, and Profit for our criteria range. If we enter these field headings in cells F1, G1, and H21 respectively, we can then move on to setting up our criteria.

This is the criteria range for the DSUM database function.

E	F	G	H	I
	Day	Sales	Profit	
	T*	>2200	<1900	

In the row under the field headings, we have T*, >2200, <1900. This means that records with a Day starting with T, Sales greater than 2200, and Profit less than 1900 will be selected for summation.

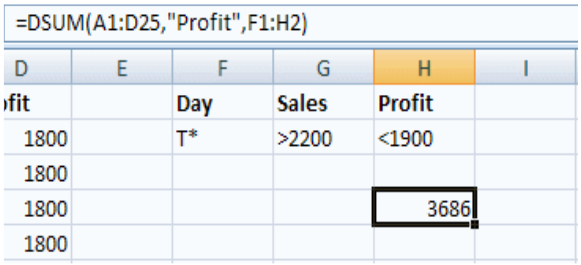
If you wanted to join the criteria with a logical Or operator, you would offset the criteria on separate rows (like an advanced filter).

Entering the Database Function

The next step is to choose a cell (H4) to place the function in. To enter the function, activate the cell and click the function (fx) button on the formula bar. In the Insert Function dialogue box, choose DSUM from the database category to reveal the Function Arguments box.

First, select your original list with your mouse, or enter the range directly into the Database data area. Press Tab or use your mouse to move to the Field data area and type Profit (this is the field that will be summed).

Then, put your cursor in the Criteria data area, and select the entire criteria range (F1:H2) with your mouse. To finish, click the OK button.



You can see the sum of the Profit fields for records that meet the criteria in cell H4.

You could have also activated cell H4, and entered the function directly into the formula bar:

=DSUM(A1:D25,"Profit",F1:H2)

The **database range** is A1:D25

The **field** that is being summed is Profit

The **criteria range** is F1:H2.

If you enter a database function directly into the formula bar, remember to keep your cell ranges accurate, and put the field that will be summed, averaged, or otherwise operated on, in quotation marks.

Module Four: Creating Pivot Tables

PivotTables allow you to analyse numeric data in depth. You can use this tool to answer unanticipated questions about data. PivotTables are interactive, cross-tabulated Excel reports that summarize and analyze data. In this module, you'll learn how to insert a chart. You'll gain an understanding of the PivotTable Tools tab. You'll also learn how to choose fields for your table and group data.

What is a PivotTable

A PivotTable is a powerful tool for exploring and analysing information. A PivotTable helps you organise and manipulate the raw data in your spreadsheet, giving you insight into patterns or relationships that might not be obvious at first glance. PivotTables also give you the power to view your data in a different context without changing the original content or structure.

You can base a PivotTable on data in your current workbook or even external data from another source if you wish. With a PivotTable, you can conveniently drag and drop columns of your data to different areas of the table to examine relationships or trends that may not be obvious in a traditional Excel table or database.

You could build several separate tables to explore how columns from an Excel worksheet relate to each other, or you can use one PivotTable to do the same thing. With a PivotTable, you can alter the table design without cutting, copying, pasting, or adjusting formulas and cell references. (These tasks can be frustrating when dealing with a large volume of data.)

In short, PivotTables enable you to organise your data in meaningful ways without doing a lot of tedious work. You could say that a PivotTable is like several data tables rolled into one.

Preparing Data to Create a PivotTable

Ideally, source data for a PivotTable should be structured like a traditional Excel table or database. The source data should have a row of unique column headings distinguishing the data and there should be no empty columns interspersed within the data. Also, blank rows in a source list or database can limit the usefulness of your PivotTable.

The following image shows a block of contiguous data that is well suited for a PivotTable.

	A	B	C	D	E	F
1						
2				Profit Table		
3	Month	Salesman	Region	Product	Sales	Profit
4	Month 1	A.Smith	Northeast	Type 1	100	$ 200.00
5	Month 1	J.Adams	Southwest	Type 2	250	$ 500.00
6	Month 1	B.Doe	North	Type 3	300	$ 600.00
7	Month 1	M.Parker	Midwest	Type 4	400	$ 800.00
8	Month 1	A.Smith	East	Type 5	300	$ 600.00
9	Month 1	J.Adams	West	Type 6	525	$ 1,050.00
10	Month 2	A.Smith	Northeast	Type 1	200	$ 400.00
11	Month 2	J.Adams	Southwest	Type 2	250	$ 500.00
12	Month 2	B.Doe	North	Type 3	300	$ 600.00
13	Month 2	M.Parker	Midwest	Type 4	400	$ 800.00
14	Month 2	M.Parker	East	Type 5	450	$ 900.00
15	Month 2	B.Doe	West	Type 6	500	$ 1,000.00
16	Month 3	B.Doe	Northeast	Type 1	100	$ 200.00
17	Month 3	J.Adams	Southwest	Type 2	700	$ 1,400.00

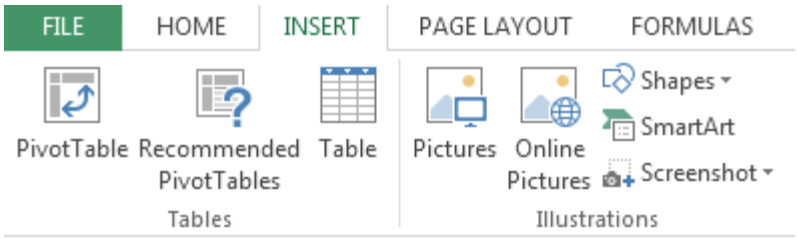
Notice that there are no empty rows or columns and that every column of data has a unique label.

Creating a PivotTable

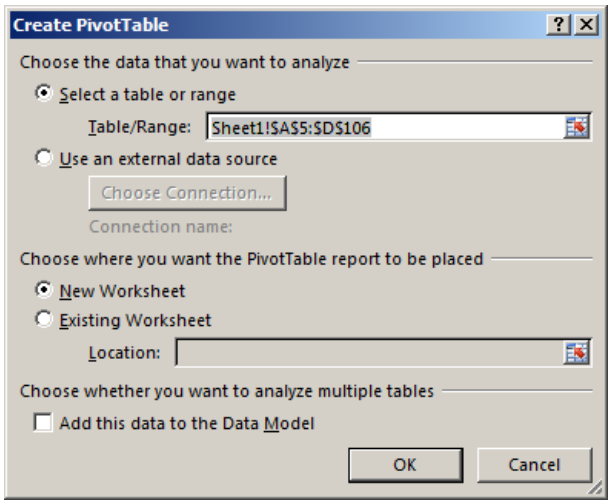
To create an Excel 2010 PivotTable, just select the range of data that you want to base the table on and then click the PivotTable button on the Insert Ribbon to display the PivotTable dialogue.

Use the following procedure to insert a PivotTable.

1. Place your cursor somewhere in the data you want to analyze.
2. Select the **Insert** tab from the Ribbon.
3. Select **PivotTable**.



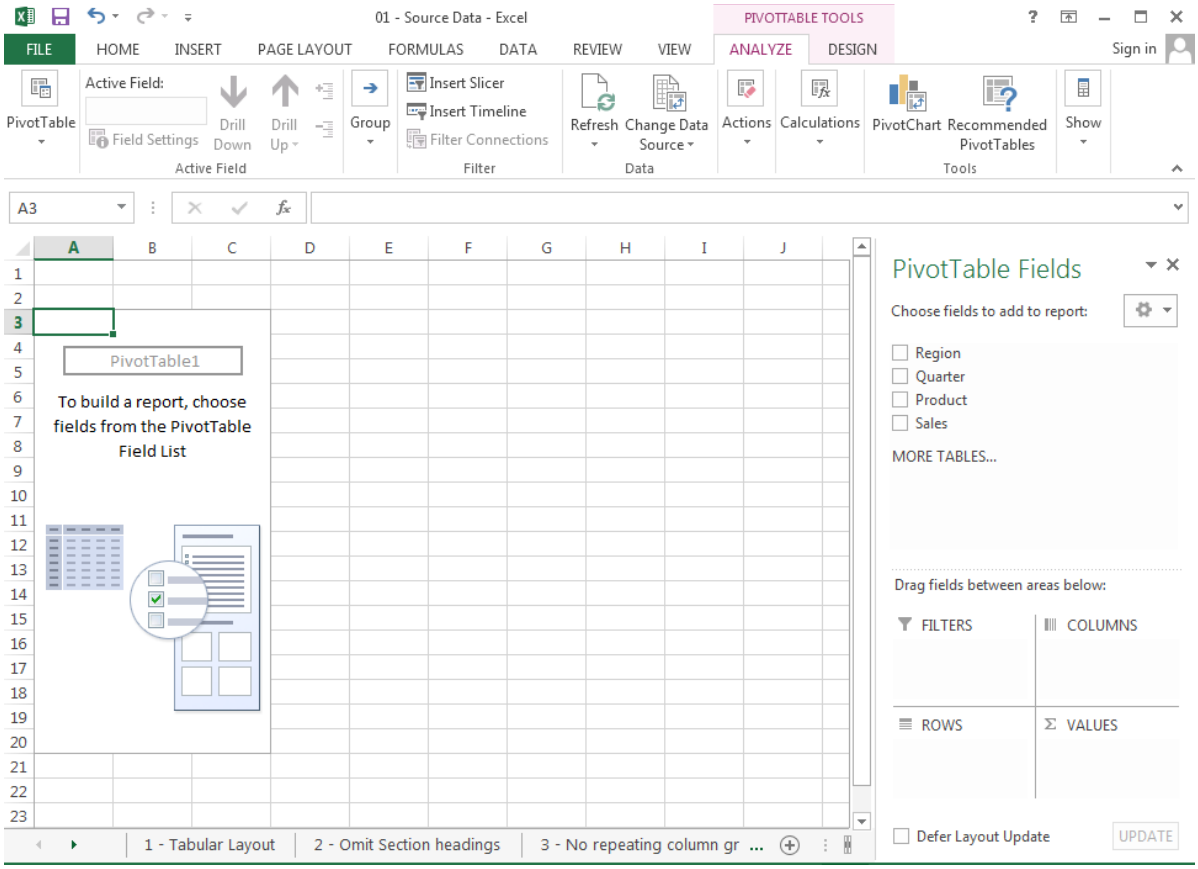
Excel displays the Create PivotTable dialog box.



4. Excel automatically provides a range of cells based on your selection. You can change the table or range if desired.
5. Select a location for the PivotTable. You can have Excel create a new worksheet or select one of the existing sheets.
6. Select **OK**.

If you choose the External Data Source option, you can base your PivotTable on data outside your current workbook (such as another workbook or perhaps an external database). If you select the Use an External Data Source radio button, you will be able to display a drop list of existing connections in the Connection Name field. A typical existing connection could be a Microsoft Query or a connection you previously made to an Access database for some other purpose.

Excel displays the PivotTable and the Field List for you to begin choosing your fields and grouping data (discussed in the next topic).

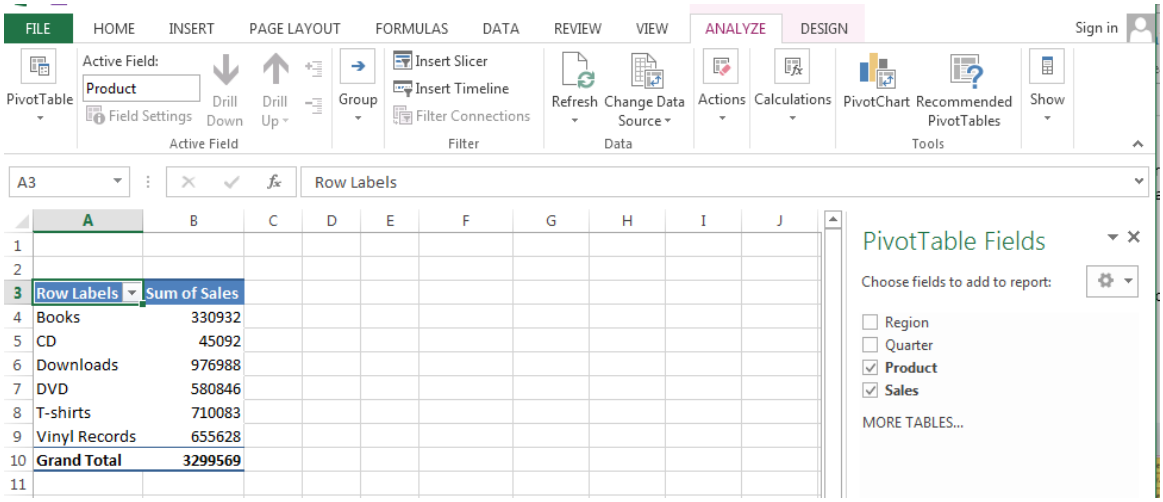


Choosing Fields and Grouping Data

When you add a PivotTable or PivotChart to your worksheet, the table or chart is blank at first. You must choose the fields you want to display on the chart. The PivotTable Field List pane makes it easy to select the fields you want on the chart.

Use the following procedure to add fields to the PivotTable report.

1. Check the box next to a field listed in the PivotTable Field list to include it in the report. The default location where fields are added are as follows:
   - Nonnumeric fields are added to the Row Labels.
   - Numerical fields are added to the Values area.
   - Date and time values are added to the Column Labels.

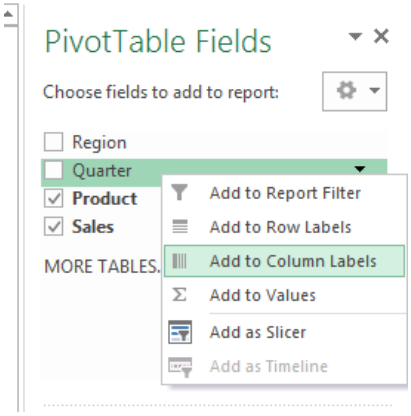


The bottom of the PivotTable Field List pane includes four areas:

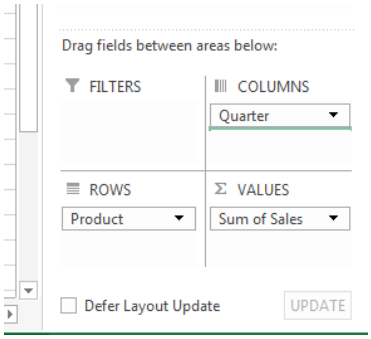
- Report Filter
- Axis Fields (categories or row labels)
- Legend Fields (column labels)
- Values

Use the following procedure to group the data.

1. Right click on a field label in the PivotTable Field List and select one of the options from the context menu.



2. You can also simply drag the fields from one area to another. You can even drag a field from the top portion of the pane to one of the bottom areas.

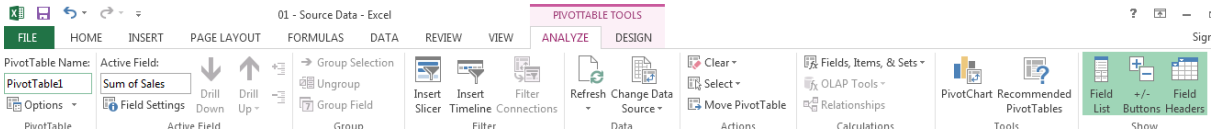


Overview of the Pivot Table Tools Tabs

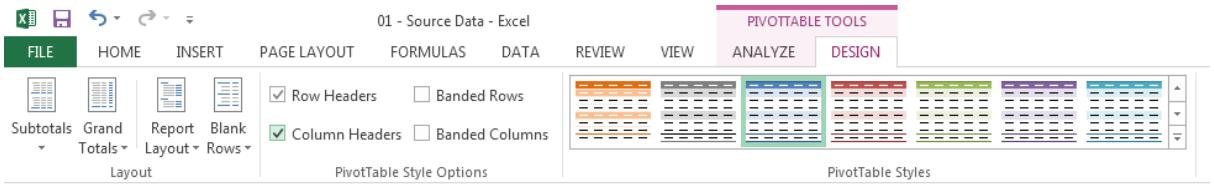
You may have noticed the PivotTable Tools tabs that appear when you inserted your chart. These contextual tabs are used throughout Office 2010. The appropriate tab appears, depending on which type of object you are using.

Tools tabs for working with PivotTables

ANALYSE



DESIGN

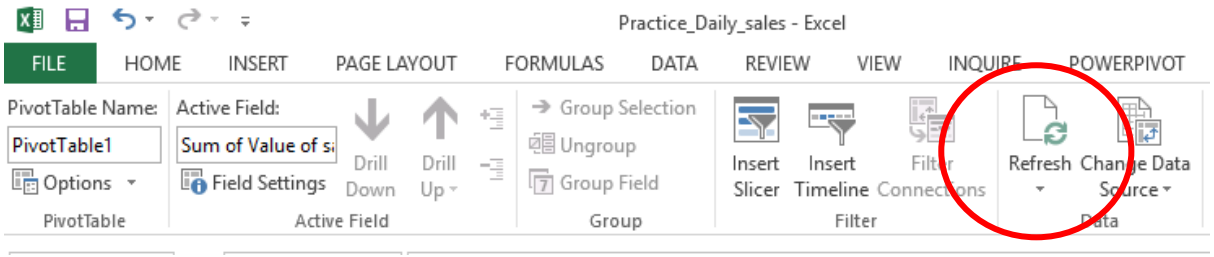


Changing Data Displayed and Refreshing the PivotTable

PivotTables are meant to be interactive, so Excel makes it easy to change the data. You can select the Refresh tool at any time to update the table.

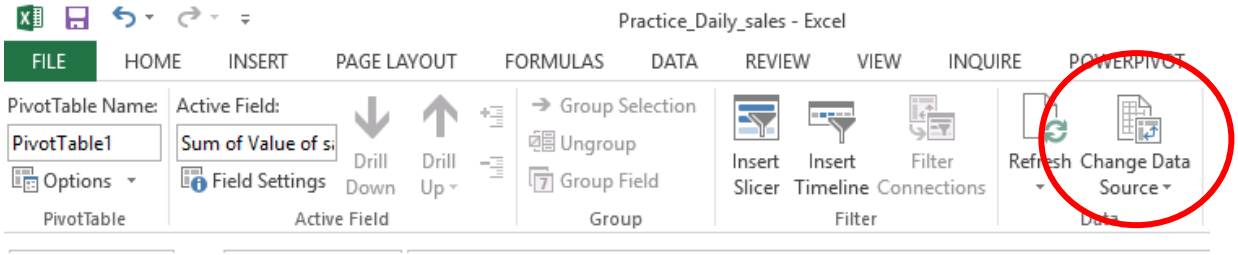
Use the following procedure to refresh the PivotTable after making a change to the data.

1. Return to the worksheet containing the PivotTable.
2. Click somewhere on the PivotTable.
3. Select the **Analyze** tab from the Ribbon.
4. Select **Refresh**.

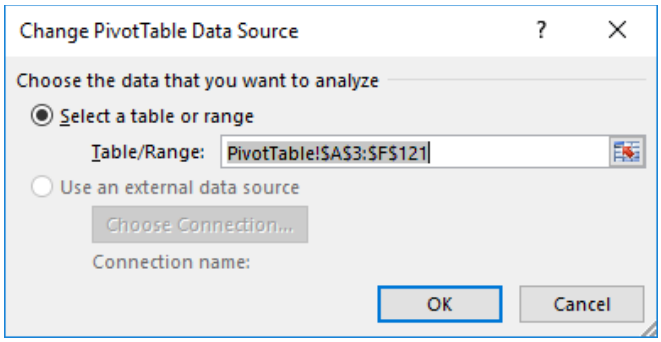


Use the following procedure to change the data source.

1. Select the **Analyze** tab from the Ribbon.
2. Select **Change Data Source**.



Excel returns to the worksheet of the source data and highlights the current data source. It also displays the Change PivotTable Data Source dialog box.



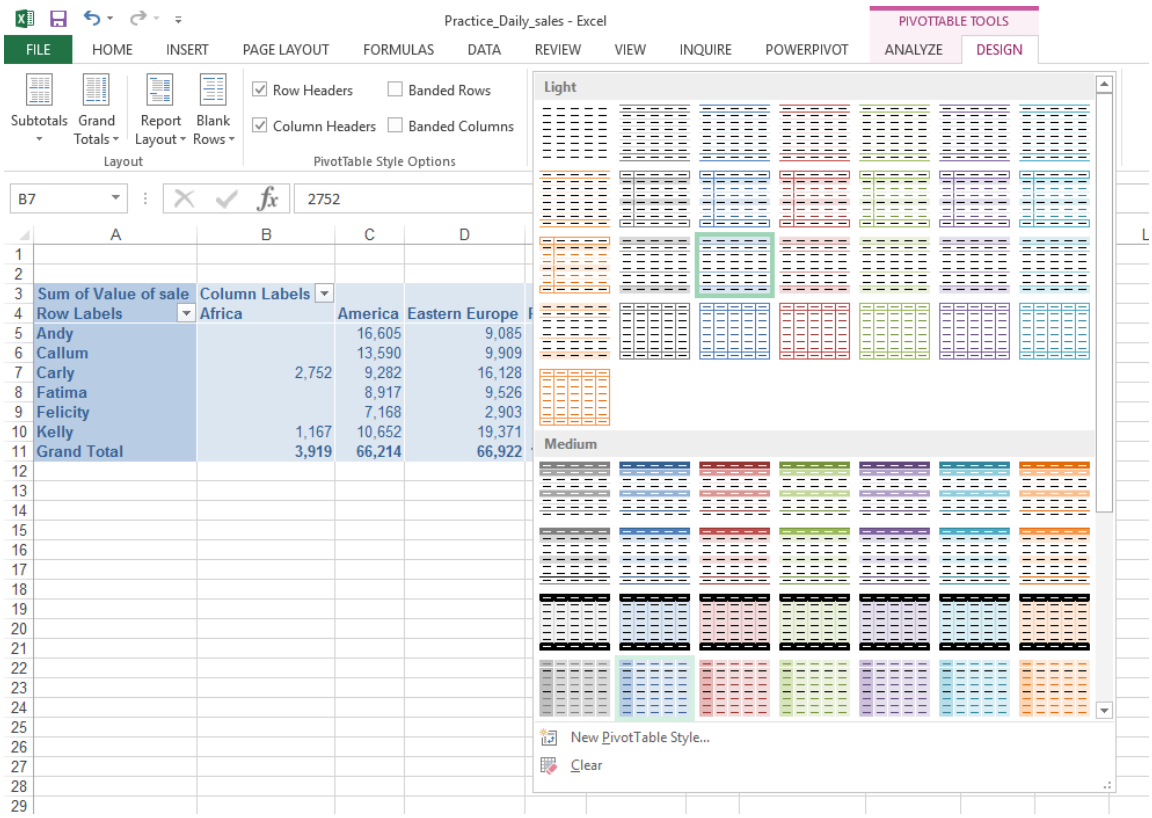
3. Highlight the new data area on the worksheet.

Applying a Style to Your Pivot Table

Excel includes many different predefined styles to update the look of your PivotTable.

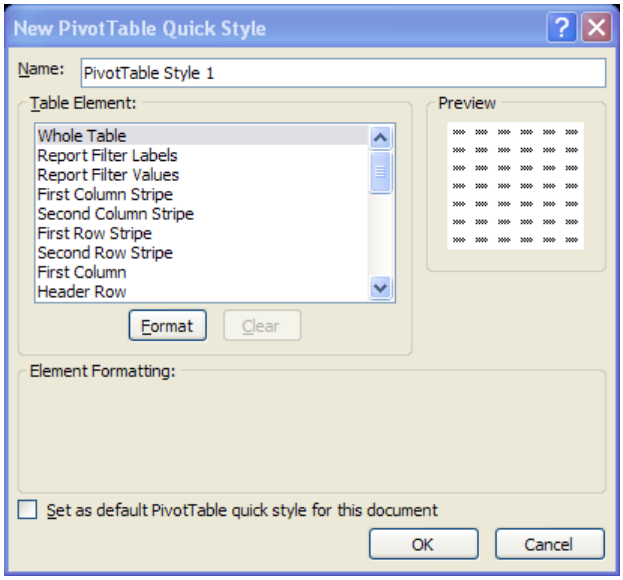
Use the following procedure to apply a style to the PivotTable.

1. Select the **Design** tab of the Ribbon.
2. Select the small arrow in the PivotTable Styles area to see the PivotTable Style gallery.

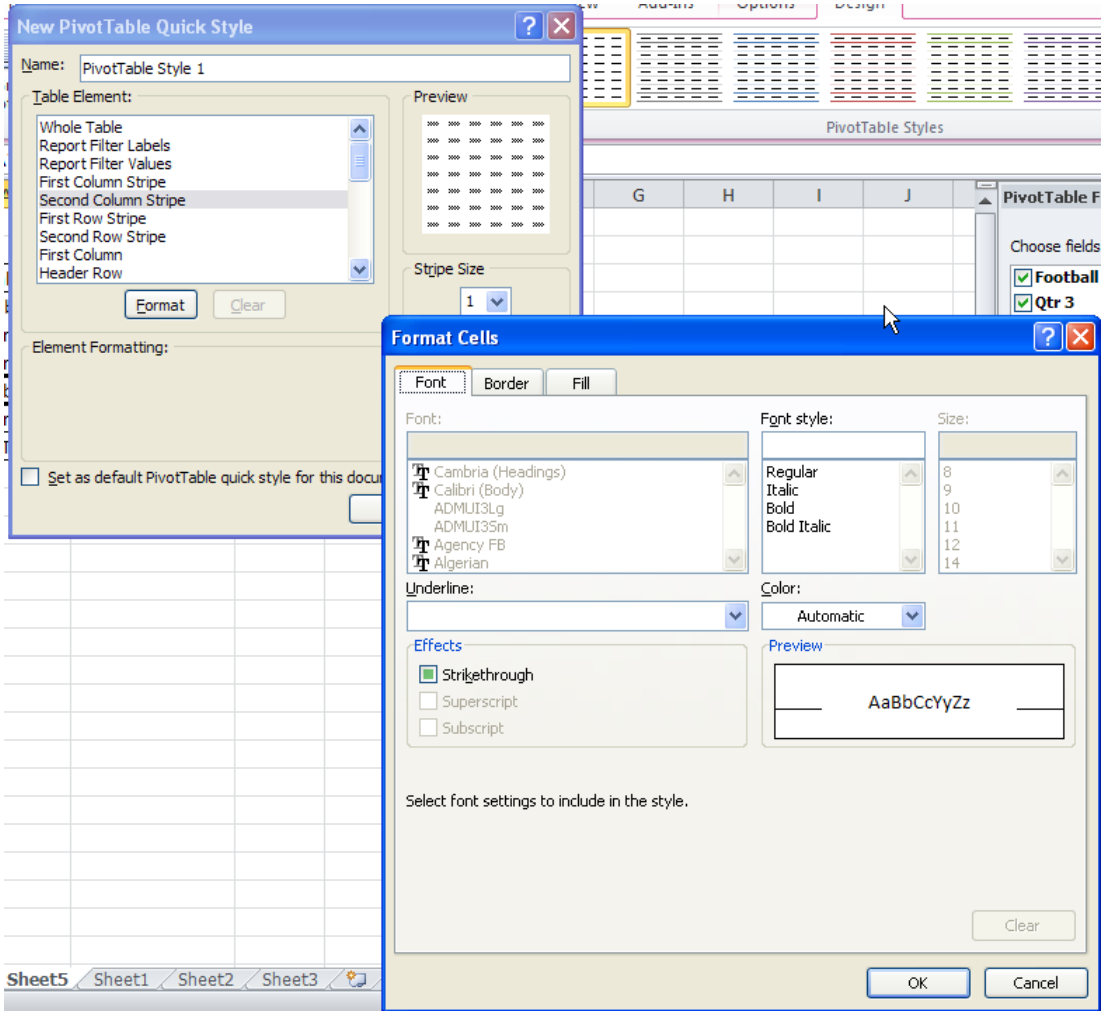


3. Select an option to apply the style.

Shown here is the New PivotTable Style dialog box.



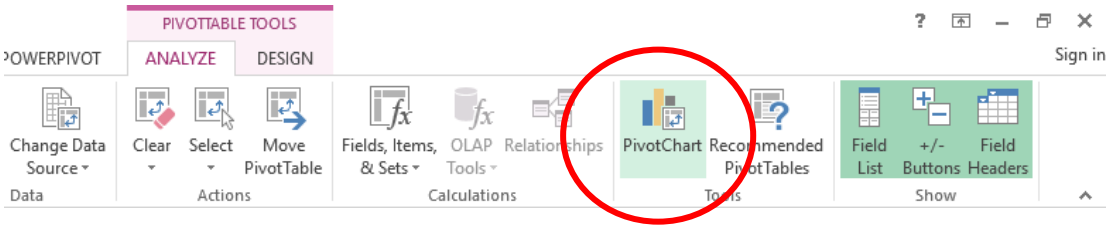
Shown here are the formatting options for one of the table elements.



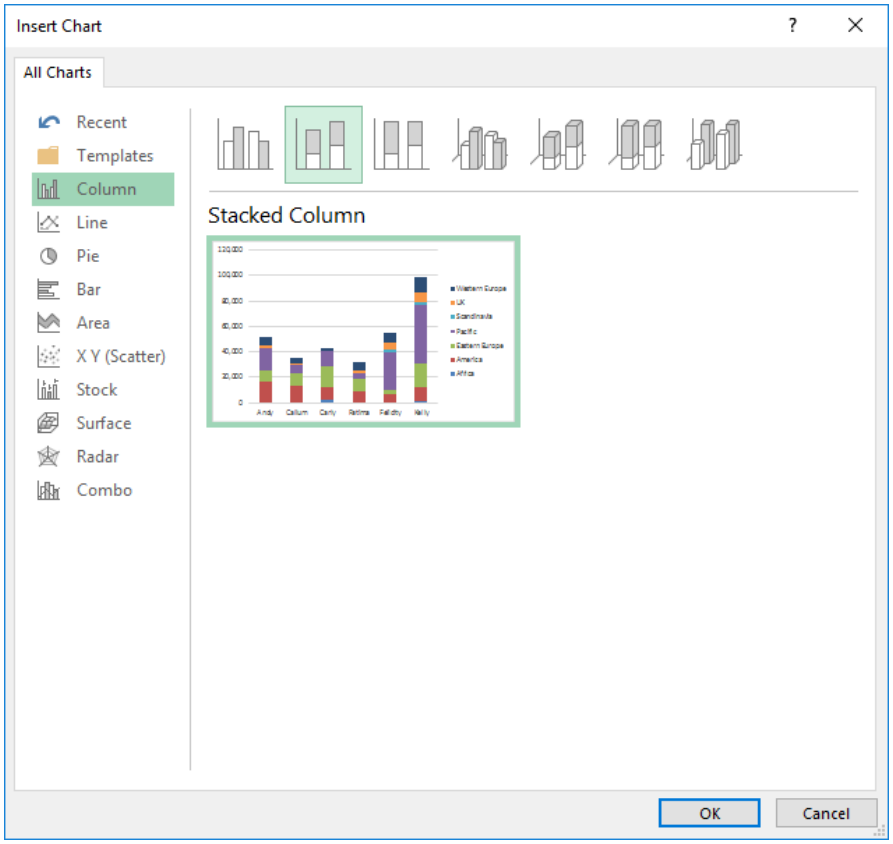
Creating a Pivot Chart from a Pivot Table

Click anywhere in the PivotTable for which you want to add a chart.

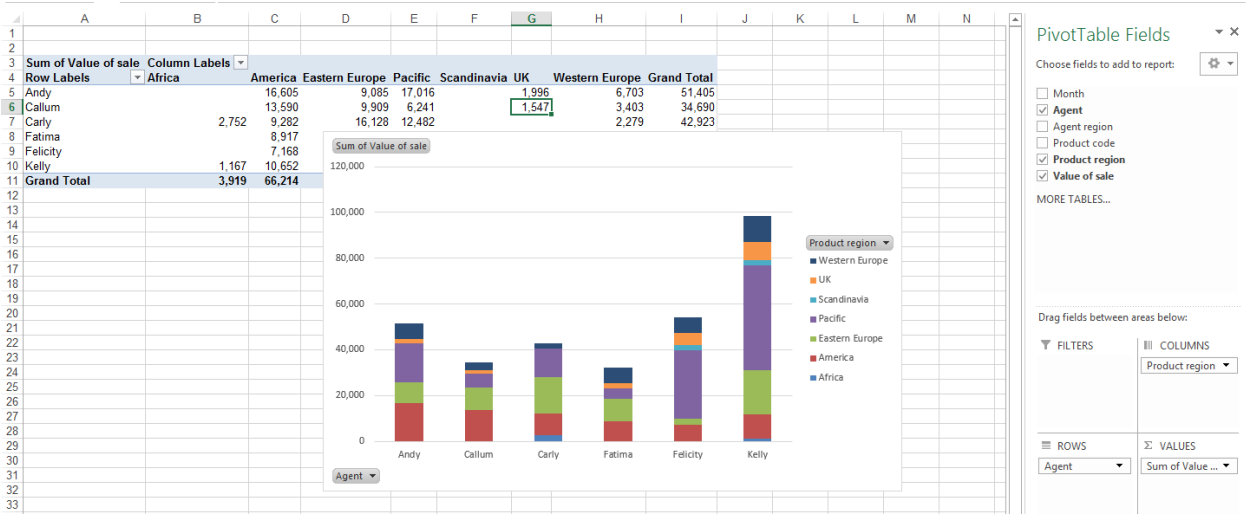
1. Select the **Analyze** tab from the Ribbon.
2. Select **PivotChart**.



Excel displays the *Insert Chart* dialog box.



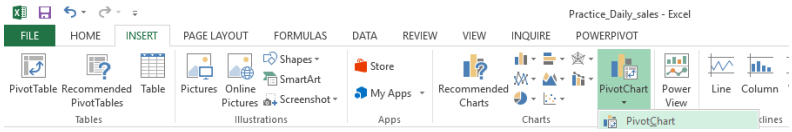
3. Select the desired type of chart and select **OK**.



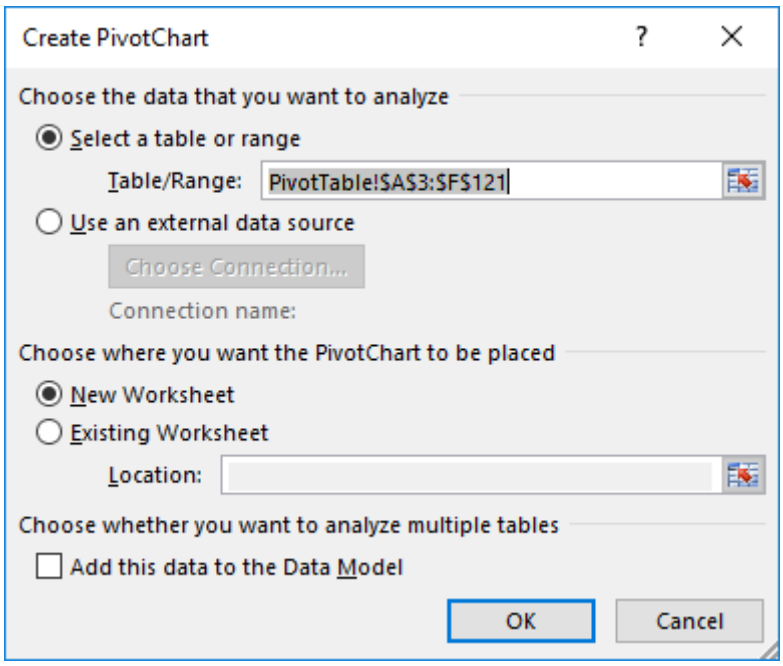
Creating a Pivot Chart from Data

Use the following procedure to insert a PivotChart.

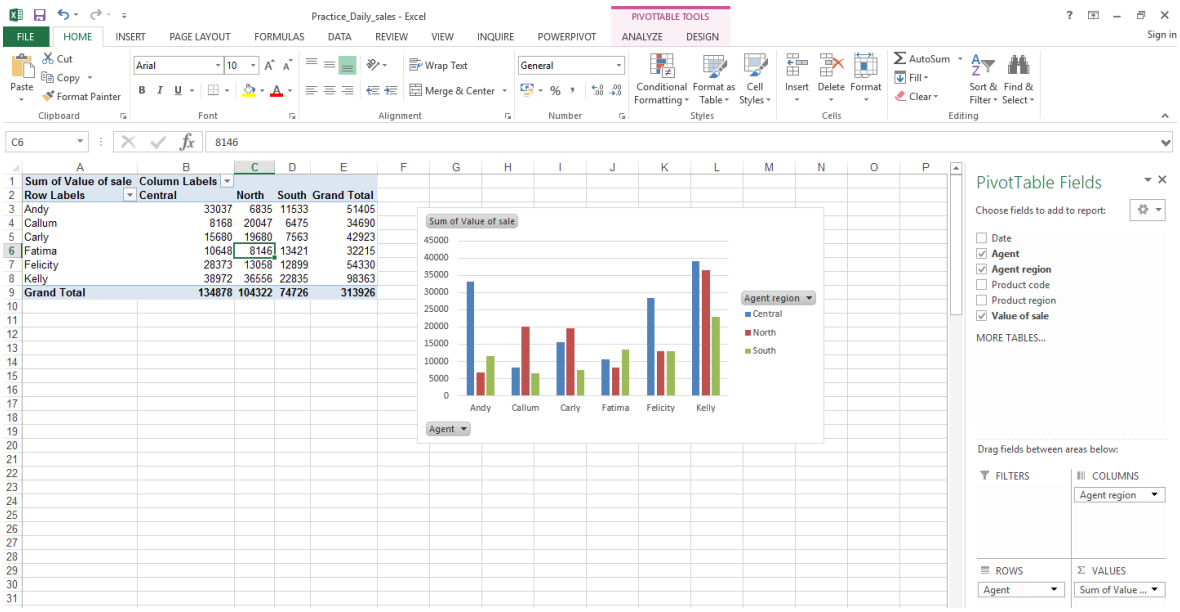
1. Place your cursor somewhere in the data you want to analyze.
2. Select the **Insert** tab from the Ribbon.
3. Select **PivotChart**.



Excel displays the Create PivotTable with PivotChart dialog box.



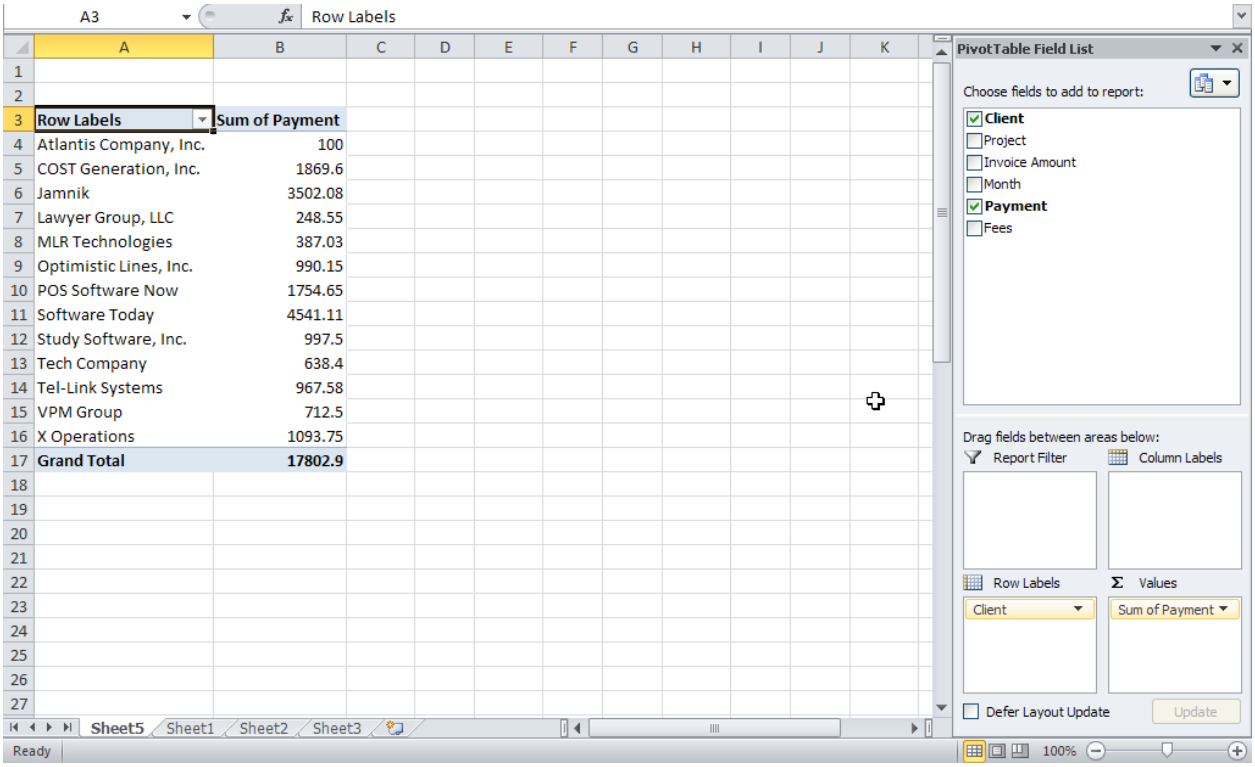
4. Excel automatically provides a range of cells based on your selection. You can change the table or range if desired.
5. Select a location for the PivotChart. You can have Excel create a new worksheet or select one of the existing sheets.
6. Select **OK**.
   Excel displays the PivotChart and the Field List for you to begin choosing your fields and grouping data.
7. Add fields to view the chart.


Practice_Daily_sales - Excel
PIVOTTABLE TOOLS
FILE
HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
INQUIRE
POWERPIVOT
ANALYZE
DESIGN
Sign in
Paste
Cut
Copy
Format Painter
Arial
10
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Clipboard
Font
Alignment
Number
Styles
Cells
Editing
C6
8146
Sum of Value of sale
Column Labels
Row Labels
Central
North
South
Grand Total
Andy 33037 6835 11533 51405
Callum 8168 20047 6475 34690
Carly 15680 19680 7563 42923
Fatima 10648 8146 13421 32215
Felicity 28373 13058 12899 54330
Kelly 38972 36556 22835 98363
Grand Total 134878 104322 74726 313926
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
Andy
Callum
Carly
Fatima
Felicity
Kelly
Agent region
Central
North
South
Agent
PivotTable Fields
Choose fields to add to report:
Date
Agent
Agent region
Product code
Product region
Value of sale
MORE TABLES...
Drag fields between areas below:
FILTERS
COLUMNS
Agent region
ROWS
VALUES
Agent
Sum of Value ...

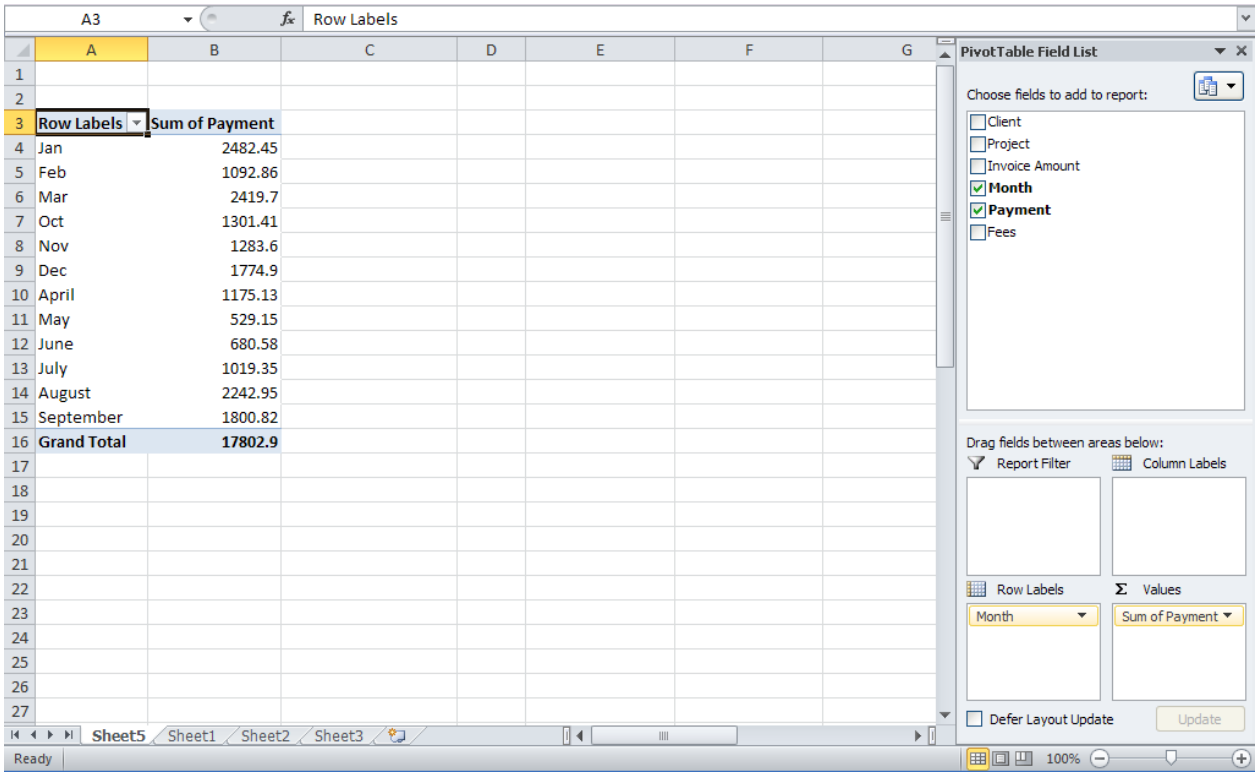
Some Real-life Examples

Many people shy away from PivotTables because they seem complex. However, PivotTables are a really powerful tool to aggregate your data or view it in different ways. We'll take a look at a few examples to get you started on your PivotTable journey.

For the reallifeexample1.xlsx, find out the sum of payments from each client.

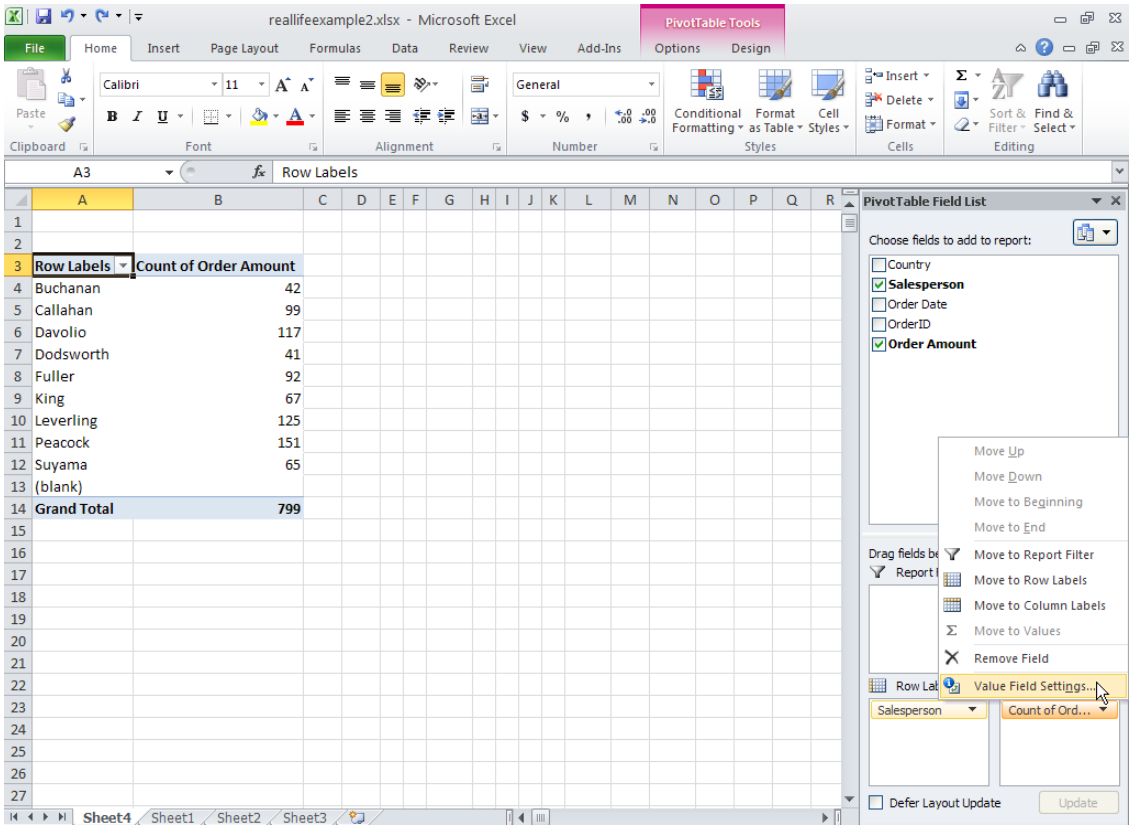


Now find out the sum of payments by month.

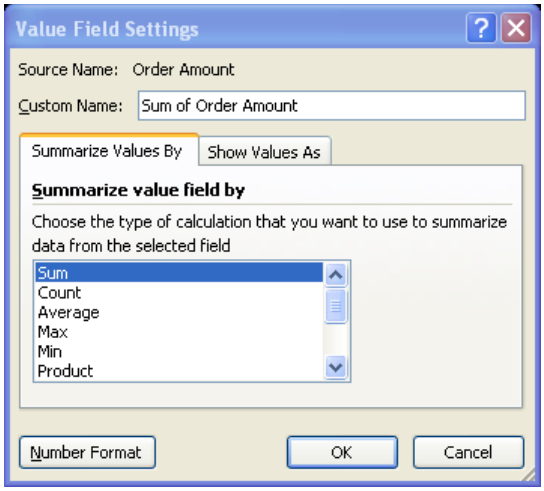


In *reallifeexample2.xlsx*, what are the order amounts for each salesperson?

If you create a PivotTable that includes the Salesperson and the Order Amount, you get this.



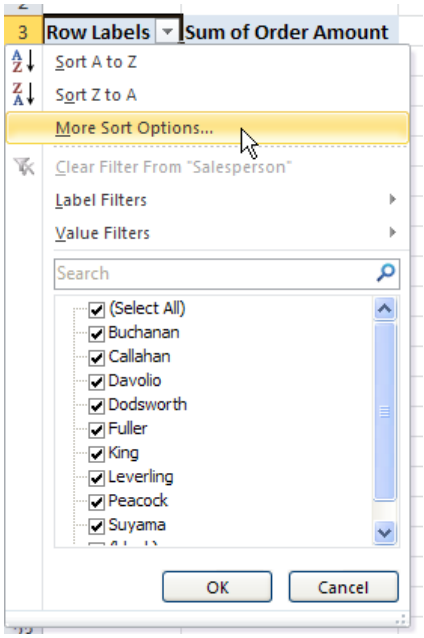
Select the arrow next to the Count of Order field in the Value area and select **Value Field Settings** from the menu.



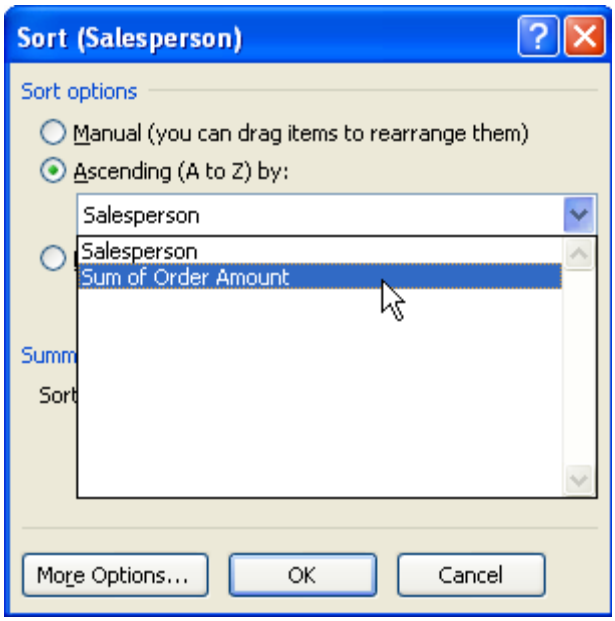
Select **Sum** and click **OK**.

Who are the top ten salesmen?

Click the arrow next to **Row labels** and select **More Sort Options**.



In the *Sort* dialog box, select **Ascending** (**A to Z**) by **Sum of Order Amount** and select OK**.**



Module Five: Using What If Analysis

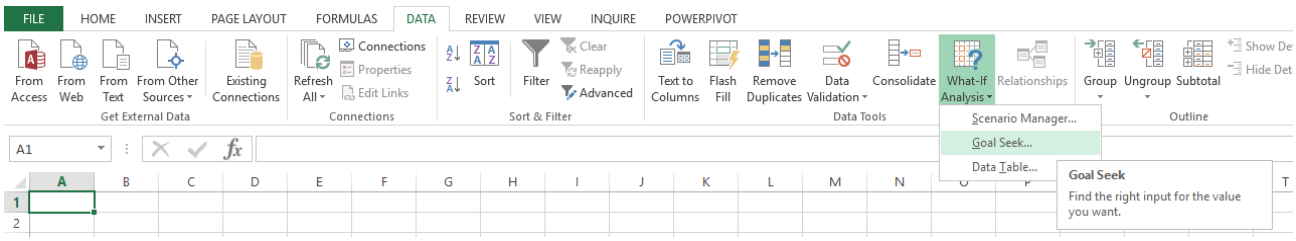
"What-if" analysis allows you to have Excel change the values in cells so that you can see how those changes affect the formulas outcomes. There are three kinds of what if analysis: goal seek, scenarios, and data tables. Goal seek allows you to find the necessary value for an unknown in a formula to obtain desired results. Scenarios allow you to view multiple different possible results for up to 32 variables. Data tables allow you to quickly calculate multiple results for one or two variables in one operation. You can view and compare the results of all the different variations together on your worksheet. This module introduces these tools.

Using Goal Seek

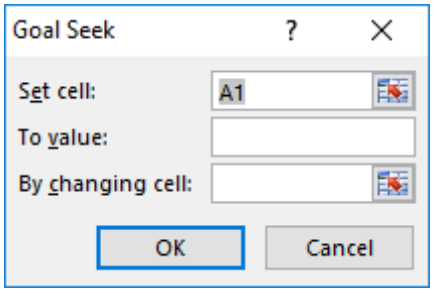
The goal seek feature allows you to plug in values in the formula you know and find out what the missing number is to get a value you want. For example, if you want to borrow money, you may want to find out what interest rate you need to get to get the payment you want. Or you may know what payment you can afford and the interest rate, but not the total amount you can afford.

Use the following procedure to use goal seek.

1. When using goal seek, one value from a formula should be left blank.
2. Select the **Data** tab from the Ribbon.
3. Select **What If Analysis**. Select **Goal Seek**.

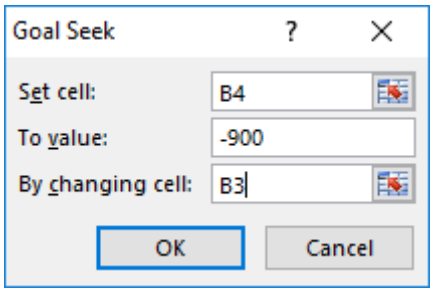


Excel displays the Goal Seek dialog box.



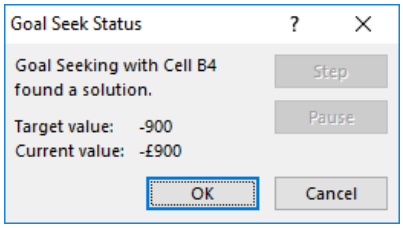
In the **Set Cell** field, enter or select from the worksheet the cell that contains the formula. In the sample file shown below, select B4. In this example the formula is =PMT(B3/12,B2,B1).

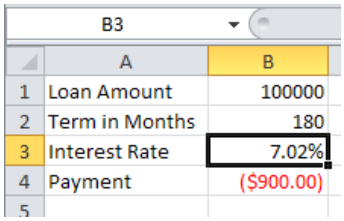
4. In the **To Value** field, enter the formula result you want. For example, in the sample file, you may want the resulting payment of 900. You would enter **-900** because it is a payment.
5. In the By Changing Cell field, enter or select the reference for the cell that contains the value you don't know. In the sample file, this is B3, the interest rate.



6. Select **OK**.

Excel displays the Goal Seek Status dialog box. Select **OK** to close it.





You may need to reformat the cell with the new answer to view the answer in the preferred format.

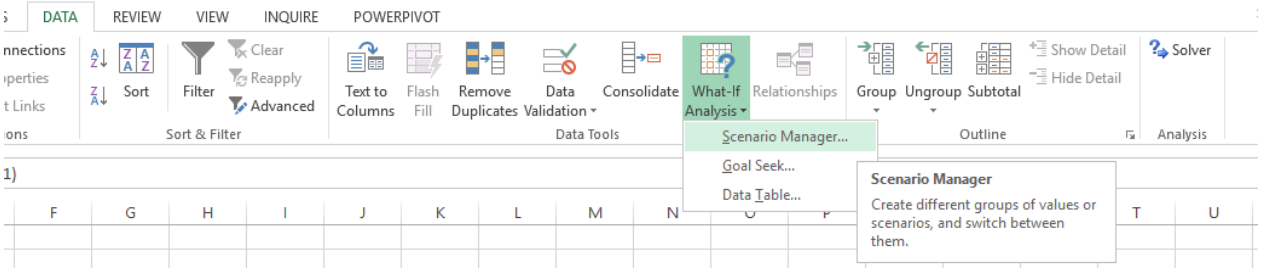
Using the Scenario Manager

A scenario is a set of values that Excel can substitute in cells on a worksheet to allow you to see how those different values influence the results.

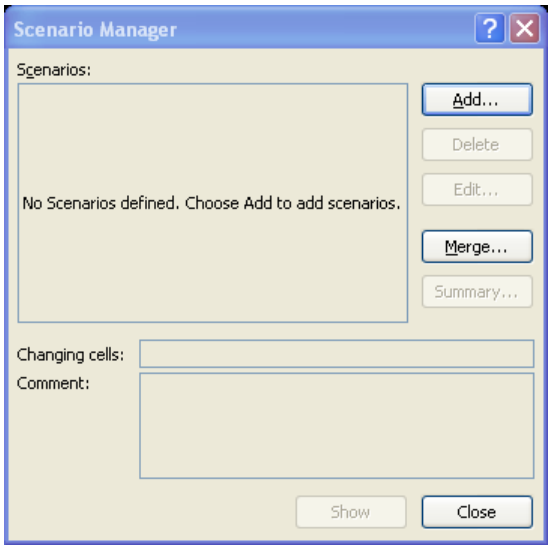
First you create a scenario for each different value. You can double-click the scenarios in the Scenario Manager to see the different results of each scenario. Then you create a summary to view a report containing the results.

Use the following procedure to add a scenario.

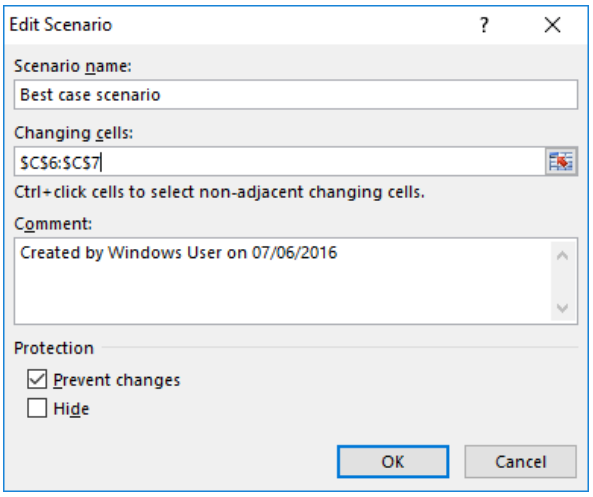
1. Select the Data tab from the Ribbon.
2. Select **What If Analysis**. Select **Scenario Manager**.



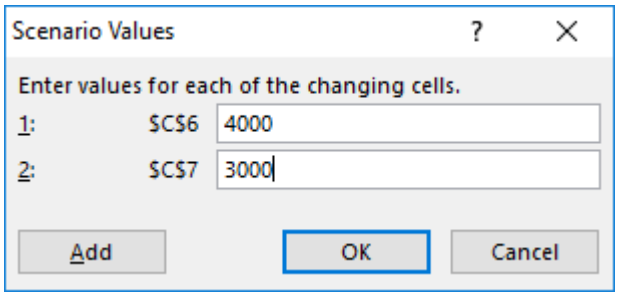
3. In the *Scenario Manager* dialog box, select **Add** to create a new scenario.



4. In the Edit Scenario dialog box, enter a **Scenario Name**.
5. In the **Changing Cells** field, enter (or select from the worksheet) the multiple cells of changing values in the first scenario. Press the CTRL key while selecting each value.
6. Enter a **Comment**, if desired.
7. Protect the scenario by checking the **Prevent changes** and/or the **Hide** boxes.
8. Select **OK**.



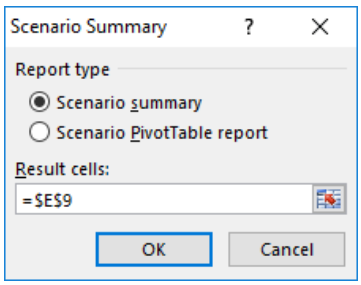
9. The *Scenario values* dialog box shows the values you selected.
   - For the original scenario, keep the values Excel displays.
   - For each subsequent scenario, enter the new values.
10. Select **Add** to create another set of values. If you have finished adding all the possibilities, select **OK** to return to the Scenario Manager.
11. Repeat steps 4 through 10 to create another scenario.



12. On the Scenario Manager dialog box, click **Show** to see the results. To view a report, select **Summary.**

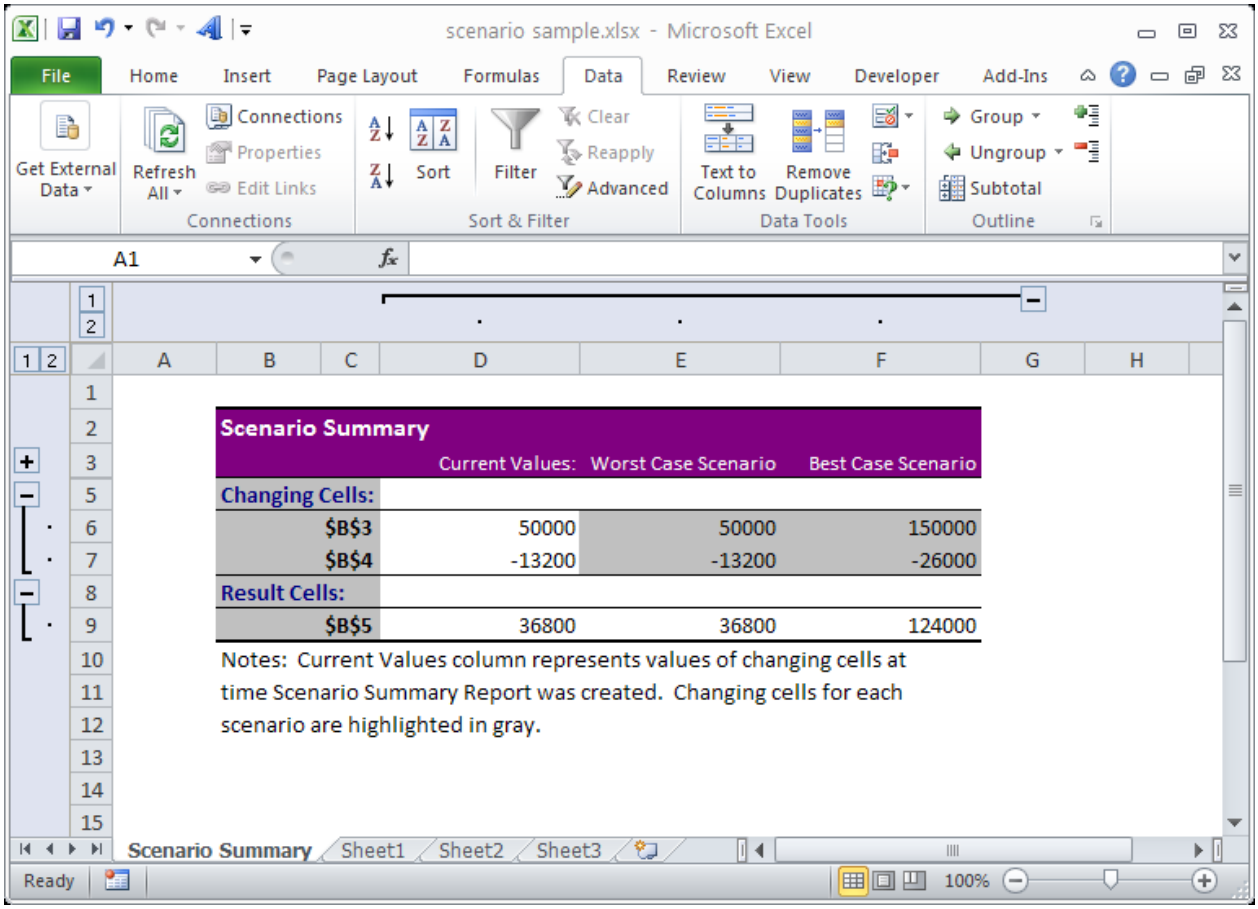
Excel displays the *Scenario Summary* dialog box.

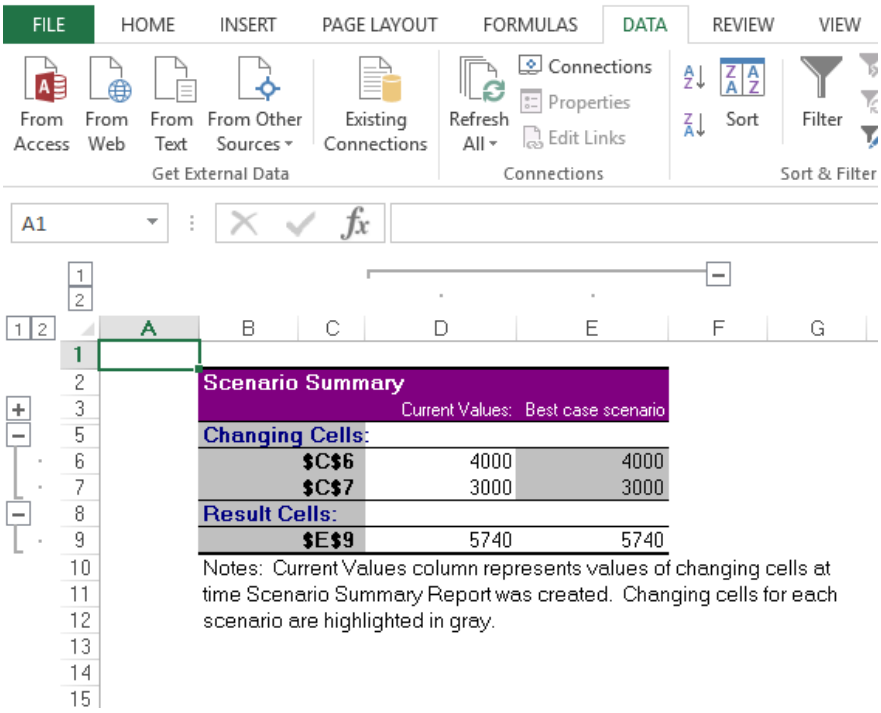
13. Indicate whether Excel should display the **Scenario Summary** or a **Scenario PivotTable Report**.
14. Select the cell that contains the results you want to compare (or the formula cell).



15. Select **OK**.

Excel displays your results in the selected format.




FILE
HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
From Access
From Web
From Text
From Other Sources
Existing Connections
Get External Data
Refresh All
Connections
Properties
Edit Links
Connections
Sort
Filter
Sort & Filter
A1
Scenario Summary
Current Values: Best case scenario
Changing Cells:
C6 4000 4000
C7 3000 3000
Result Cells:
E9 5740 5740
Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.

Using a One and Two Input Data Table

You can use Excel data tables to see how your formula results change when the data that the formula is based on changes. You do this by specifying a series of hypothetical values for Excel to evaluate the formulas with and then view the results of the evaluations. For example, you could examine how changes in the number of clients for a business will affect the income or profit.

What is a Data Table?

Data tables save you the trouble of entering several values into the worksheet and recording each recalculation of the worksheet results for later comparison.

When you use a data table, Excel will substitute a range of values into the worksheet formulas for you and tabulate the results so they can be viewed easily.

In Excel, you can create a single input data table or a two input data table. A single input table will substitute a range of values as a single variable in as many formulas as you like. With a two input data table, you can specify ranges for two input variables, but these input variables can only be applied to one formula.

An example could be, Currently, the firm has 10 clients. The values for wages, total client costs, total expenses, and profit are all calculated by formulas dependent on the number of clients the firm can retain.

To see what results the profit formula and other formulas would produce for a range of hypothetical client numbers, we can use a single input data table.

Preparing to Create a Data Table

There are some rules you should follow when building your data table to help ensure that it works correctly.

First, list the values that you want to input into the formulas in a row or column of adjacent cells. For this example, a column of input values is used. In the row just above your input column, enter cell references to the formulas that you want to evaluate.

Make sure you enter the references starting one cell to the right of the column of input values.

In the example spreadsheet that follows, the input variables are in the cell range E5:E30. Cell F4 contains the reference =C5, cell G4 contains the reference =C13, and cell H4 contains the reference =C15, for the total income, total expenses, and total profit formulas respectively.

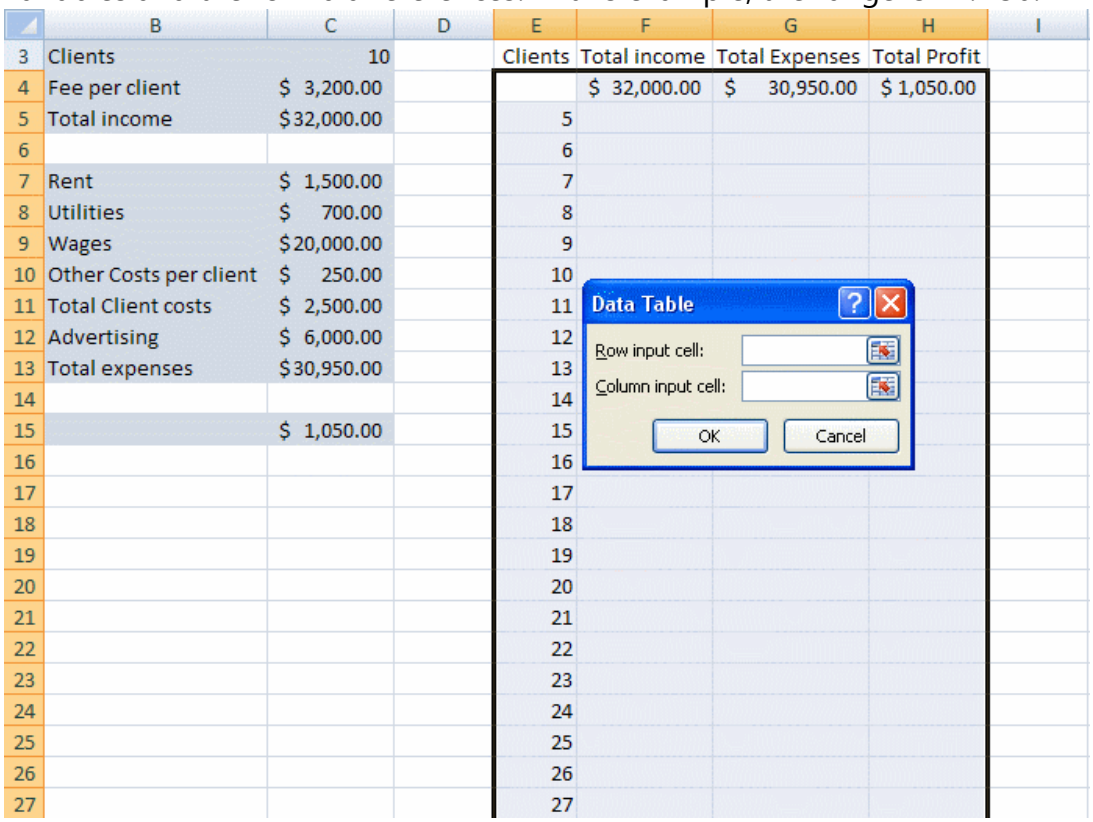
It is a good idea to label your columns appropriately, so you can clearly understand the data table results. In this example, the same labels that appear in the source data are used for the single input data table.

	B	C	D	E	F	G	H	I
1	**Expert Consultants**							
2								
3	Clients	10		Clients	Total income	Total Expenses	Total Profit	
4	Fee per client	$ 3,200.00			$ 32,000.00	$ 30,950.00	$ 1,050.00	
5	Total income	$32,000.00		5				
6				6				
7	Rent	$ 1,500.00		7				
8	Utilities	$ 700.00		8				
9	Wages	$20,000.00		9				
10	Other Costs per client	$ 250.00		10				
11	Total Client costs	$ 2,500.00		11				
12	Advertising	$ 6,000.00		12				
13	Total expenses	$30,950.00		13				
14				14				
15		$ 1,050.00		15				
16				16				
17				17				
18				18				
19				19				
20				20				
21				21				
22				22				
23				23				
24				24				
25				25				

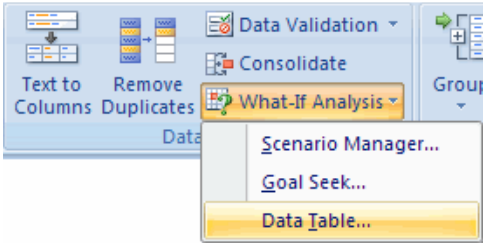
Notice that the formula references (F4:H4) are in a row just above and one cell to the right of the first input variable (E5). The data table is now ready. The input variables are listed in the Clients column and the formula references are one row above and one cell to the right. All of the elements in the data table are clearly identified.

Creating a One Input Data Table

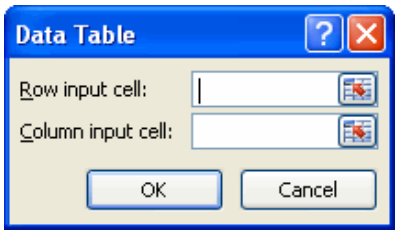
The next step is to select the range of cells from the data table containing the input variables and the formula references. In this example, the range is E4:H30.



Next, choose the Data Table option from the What-If Analysis button to display the Data Table dialogue box.



The range of input variables and formula references has been selected, and the Data Table dialogue box is ready for input.



Because the input variables are arranged in a column, we will use the "Column input cell" text field in the table dialogue box rather than the Row input cell field. In the

"Column input cell" text field, enter C3, which is the cell from original data area that contains the number of clients.

(Remember that the number of clients is also our chosen input variable.)

Clicking the OK button in the Table dialogue box will complete the data table.

	B	C	D	E	F	G	H	I
1	**Expert Consultants**							
2								
3	Clients	10		Clients	Total income	Total Expenses	Total Profit	
4	Fee per client	$ 3,200.00			$ 32,000.00	$ 30,950.00	$ 1,050.00	
5	Total income	$32,000.00		5	16000	19700	-3700	
6				6	19200	21950	-2750	
7	Rent	$ 1,500.00		7	22400	24200	-1800	
8	Utilities	$ 700.00		8	25600	26450	-850	
9	Wages	$20,000.00		9	28800	28700	100	
10	Other Costs per client	$ 250.00		10	32000	30950	1050	
11	Total Client costs	$ 2,500.00		11	35200	33200	2000	
12	Advertising	$ 6,000.00		12	38400	35450	2950	
13	Total expenses	$30,950.00		13	41600	37700	3900	
14				14	44800	39950	4850	
15		$ 1,050.00		15	48000	42200	5800	
16				16	51200	44450	6750	
17				17	54400	46700	7700	
18				18	57600	48950	8650	
19				19	60800	51200	9600	
20				20	64000	53450	10550	
21				21	67200	55700	11500	
22				22	70400	57950	12450	
23				23	73600	60200	13400	
24				24	76800	62450	14350	
25				25	80000	64700	15300	
26				26	83200	66950	16250	
27				27	86400	69200	17200	
28				28	89600	71450	18150	
29				29	92800	73700	19100	
30				30	96000	75950	20050	
31								

You can now see at a glance how the 26 different values in the Clients column influence the income, total expenses, and total profit results.

If you change the values in the Client column, the data table will recalculate in accordance with the new values automatically. You can apply formatting styles, borders, shading, and other enhancements to the data table in the same way as you would any other area of your worksheet.

	B	C	D	E	F	G	H	I
1	Expert Consultants							
2								
3	Clients	10		Clients	Total income	Total Expenses	Total Profit	
4	Fee per client	$ 3,200.00			$ 32,000.00	$ 30,950.00	$ 1,050.00	
5	Total income	$32,000.00		5	$ 16,000.00	$ 19,700.00	$ (3,700.00)	
6				6	$ 19,200.00	$ 21,950.00	$ (2,750.00)	
7	Rent	$ 1,500.00		7	$ 22,400.00	$ 24,200.00	$ (1,800.00)	
8	Utilities	$ 700.00		8	$ 25,600.00	$ 26,450.00	$ (850.00)	
9	Wages	$20,000.00		9	$ 28,800.00	$ 28,700.00	$ 100.00	
10	Other Costs per client	$ 250.00		10	$ 32,000.00	$ 30,950.00	$ 1,050.00	
11	Total Client costs	$ 2,500.00		11	$ 35,200.00	$ 33,200.00	$ 2,000.00	
12	Advertising	$ 6,000.00		12	$ 38,400.00	$ 35,450.00	$ 2,950.00	
13	Total expenses	$30,950.00		13	$ 41,600.00	$ 37,700.00	$ 3,900.00	
14				14	$ 44,800.00	$ 39,950.00	$ 4,850.00	
15		$ 1,050.00		15	$ 48,000.00	$ 42,200.00	$ 5,800.00	
16				16	$ 51,200.00	$ 44,450.00	$ 6,750.00	
17				17	$ 54,400.00	$ 46,700.00	$ 7,700.00	
18				18	$ 57,600.00	$ 48,950.00	$ 8,650.00	
19				19	$ 60,800.00	$ 51,200.00	$ 9,600.00	
20				20	$ 64,000.00	$ 53,450.00	$10,550.00	
21				21	$ 67,200.00	$ 55,700.00	$11,500.00	
22				22	$ 70,400.00	$ 57,950.00	$12,450.00	
23				23	$ 73,600.00	$ 60,200.00	$13,400.00	
24				24	$ 76,800.00	$ 62,450.00	$14,350.00	
25				25	$ 80,000.00	$ 64,700.00	$15,300.00	
26				26	$ 83,200.00	$ 66,950.00	$16,250.00	
27				27	$ 86,400.00	$ 69,200.00	$17,200.00	
28				28	$ 89,600.00	$ 71,450.00	$18,150.00	
29				29	$ 92,800.00	$ 73,700.00	$19,100.00	
30				30	$ 96,000.00	$ 75,950.00	$20,050.00	
31								

Creating a Two Input Data Table

If you want to examine the effects of two input variables on a single formula, you can use a two input data table.

To build a two input data table, first choose an empty cell and enter a reference to the formula you want to examine. Using the same consulting firm example as before, we will choose cell E2 and enter the reference =C15 for the Profit formula.

The next step is to create two variable ranges. One range will be a column starting in the cell immediately beneath the formula cell and the other range will be a row starting on the immediate right of the formula cell. This means that the formula cell (E2) will be at the upper left corner of the two input data table.

For the column of input variables, we will again use the number of clients. For the row of input variables, we will use the Fee Per Client. (The formula for the Profit values is indirectly dependant on both of these variables.)

After entering some hypothetical values for the number of clients in cells E3:E15 and some different client fee values in cells F2:J2, we will end up with a worksheet something like this:

	A	B	C	D	E	F	G	H	I	J
1		Expert Consultants				Client fees				
2					$1,050.00	$ 3,000.00	$ 3,200.00	$ 3,500.00	$4,000.00	$4,200.00
3	Income	Clients	10		7					
4		Fee per client	$ 3,200.00		8					
5		Total income	$32,000.00		9					
6					10					
7	Expenses	Rent	$ 1,500.00		11					
8		Utilities	$ 700.00		12					
9		Wages	$20,000.00		13					
10		Other Costs per client	$ 250.00		14					
11		Total Client costs	$ 2,500.00		15					
12		Advertising	$ 6,000.00		20					
13		Total expenses	$30,950.00		25					
14					30					
15	Profit		$ 1,050.00		Clients					
16										

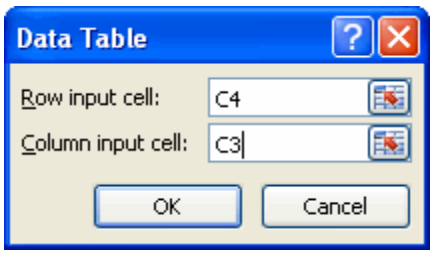
Shading, labels, and currency formatting have been added to the data table for clarity.

The next step is to invoke the Data Table dialogue – select Data ribbon, Data Tools group, click the What-If Analysis button and select Data Table. . This time, the formula referenced in E2 will be evaluated once for every combination of Clients and Client fees, that exists in the table.

First we will select the range of data cells E2:J14 and then we will choose the Data Table option from the What If Analysis button.

This time, we have two input variables: one corresponding to the row of Client fees and one corresponding to the column containing the Clients data.

In the Row Input Cell text area type C4, because the row of client fees corresponds to cell C4 in the original data. Similarly, in the Column Input Cell, type C3. The Data table dialogue box should look like this.



It is now just a matter of clicking the OK button to complete the table.

	A	B	C	D	E	F	G	H	I	J
1		Expert Consultants				Client fees				
2					$1,050.00	$ 3,000.00	$ 3,200.00	$ 3,500.00	$4,000.00	$4,200.00
3	Income	Clients	10		7	-3200	-1800	300	3800	5200
4		Fee per client	$ 3,200.00		8	-2450	-850	1550	5550	7150
5		Total income	$32,000.00		9	-1700	100	2800	7300	9100
6					10	-950	1050	4050	9050	11050
7	Expenses	Rent	$ 1,500.00		11	-200	2000	5300	10800	13000
8		Utilities	$ 700.00		12	550	2950	6550	12550	14950
9		Wages	$20,000.00		13	1300	3900	7800	14300	16900
10		Other Costs per client	$ 250.00		14	2050	4850	9050	16050	18850
11		Total Client costs	$ 2,500.00		15	2800	5800	10300	17800	20800
12		Advertising	$ 6,000.00		20	6550	10550	16550	26550	30550
13		Total expenses	$30,950.00		25	10300	15300	22800	35300	40300
14					30	14050	20050	29050	44050	50050
15	Profit		$ 1,050.00		Clients					

Now the data table contains speculative profit values based on the number of clients and the fee per client.

Using Solver

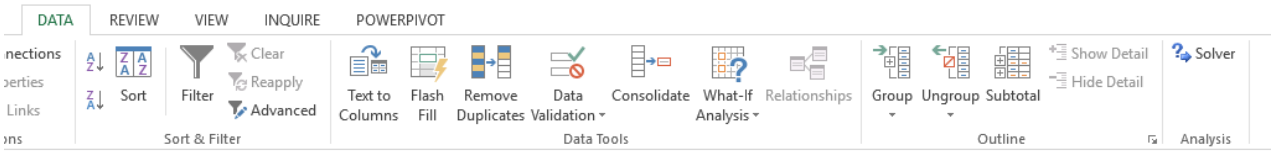
Sometimes, when dealing with more complex problems, Excel data tables or the Goal Seek feature cannot provide the kind of forecast or analysis you are looking for. In this type of situation, Excel 2010's Solver feature might be able to help.

The Solver is an Excel feature that is designed for optimising systems of equations subject to specific constraints. The Solver can be used to find optimal solutions for linear programming problems involving multiple equations and multiple unknowns. An optimal solution might be one that maximizes profit, or it could be one that minimizes costs. Basically, the optimal solution will depend on the context of the situation and what you are looking for.

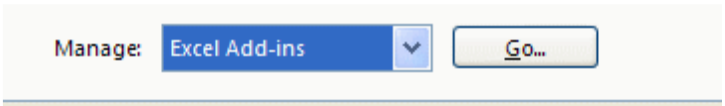
If you are trying to solve a complex problem, the Solver will require certain information for it to work correctly. You will have to designate a formula that references the unknowns you want to solve for, and you will have to define constraints that model the given situation. The best way to get an idea of how Solver works is to see it used in an example.

Checking Solver Installation

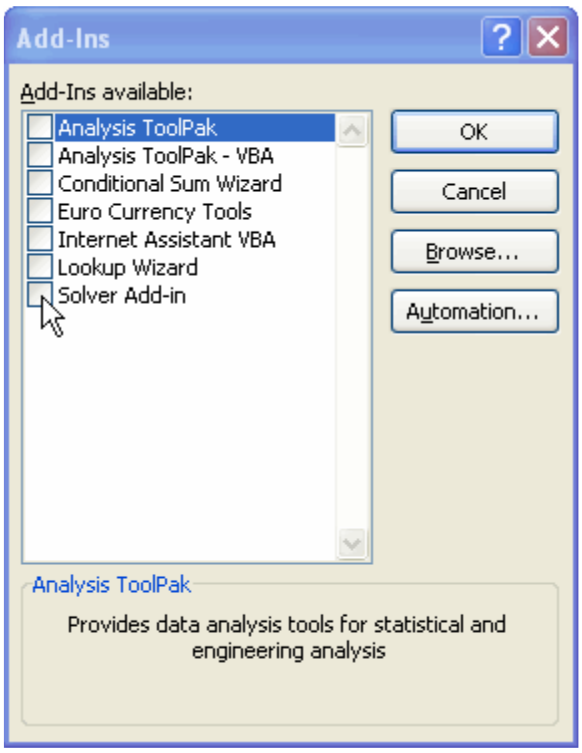
First, since Solver is an Excel add-in, it may not yet be installed. You can check this by clicking the Data tab to display the Data Ribbon. You may see the Solver button to the far right of the ribbon.



If there is no Solver option, the Solver has not yet been installed. To install the Solver, open the File tab and click the Options button at the bottom of the left pane. When you see the Excel Options window, choose Add-Ins from the panel on the left, and then use the drop list at the bottom to specify Excel Add-ins.



When you are ready, click Go to display the Excel Add-ins.



When you see the Excel Add-Ins box, put a check next to the Solver Add-in option in the available add-ins list and click OK. When Excel finishes installing the Solver, you should be able to access the Solver option on the Data Ribbon.

When to use Solver

In order to use the Excel Solver, you must set up the worksheet correctly. This requires a solid understanding of the problem you are trying to solve.

The following example involves a business that assembles and sells computers. The business sells two desktop models: the Budget PC and the Power PC. The Budget model is less powerful than the other model, but the price is very reasonable. The Power PC has more computing power and storage than the budget model, but it is also more expensive.

	A	B	C	D	E	F	G	H
1		Computer Assembly						
2								
3		Hardware labor	Software labor	# Assembled	Price	Revenue	Total SW hours	Total HW hours
4	Budget PC	3	1		550	0	0	0
5	Power PC	4	2		1200	0		
6	Total			0		0		
7								
8								
9							Constraints	
10							Hardware Hrs <=	80
11							Software Hrs <=	30
12							# Power pc <=	10
13							# assembled >=	0
14								

From the worksheet, you can see that it takes 3 hours of hardware labour to assemble the Budget PC and 4 hours of hardware labour to assemble the Power PC.

Once the hardware is assembled, it takes 1 hour of software labour to install and configure the operating system and other programs on the Budget PC, and 2 hours of software labour to install all the necessary programs on the Power PC.

Our goal is to find out how many of each type of computer we should make to maximize our weekly total revenue, and what the maximum total revenue would be.

To complicate matters, the business employs one part time software installer available for 30 hours of software labour per week, and two full time hardware technicians that provide 80 hours of hardware labour a week. Also, the company that supplies the business with processors can supply only 10 Power PC processors a week.

This means that the total software labour used must be less than or equal to 30 hours for the week and the total hardware labour must be less than or equal to 80 hours. Furthermore, the number of Power PC models we can make in a week must be less than or equal to 10.

	A	B	C	D	E	F	G	H
1		Computer Assembly						
2								
3		Hardware labor	Software labor	# Assembled	Price	Revenue	Total SW hours	Total HW hours
4	Budget PC	3	1		550	=D4*E4	=C4*D4+C5*D5	=B4*D4+B5*D5
5	Power PC	4	2		1200	=D5*E5		
6	Total			=D4+D5		=F4+F5		
7								
8								
9							Constraints	
10							Hardware Hrs <=	80
11							Software Hrs <=	30
12							# Power pc <=	10
13							# assembled >=	0

Cells B4 and C4 hold the number of hardware and software labour hours required to build a Budget PC. Similarly, cells B5 and C5 contain the number of hardware and software hours required to assemble a Power PC.

The revenues from the Budget computers can be calculated by multiplying the number assembled by the sale price. (This is what the formula in F4 does.)

The revenues from the power computers are calculated by a similar formula in cell F5. Notice that in both cases, the revenue formulas depend on the number of computers assembled (cells D4 and D5). Because of this, the total revenue formula (F6) is indirectly dependent on D4 and D5.

The formula we want to optimise (also known as the objective formula) represents the total revenue (F6). The cells we will change to maximize the total revenue formula represent the quantities of each type of computer assembled (D4 for Budget PC, and D5 for Power PC).

The constraints for this problem are shown in the green highlighted area. It is not absolutely necessary to label the constraints as they are shown here, but clearly identifying them on the worksheet helps when entering the constraints in the solver. Basically the constraints specify that the hardware hours are to be less than or equal to 80, the software hours are to be less than or equal to 30, the number of Power PC's that can be assembled is less than or equal to 10, and the number of each type of computer assembled has to be greater than or equal to 0.

This last constraint may seem obvious and silly, but it is important to include it so the solver knows that using negative values in the changing cells is not an option when optimising the objective formula.

The formulas that calculate the total software hours and hardware hours used are in cells G4 and H4 respectively. You should notice that these formulas are also dependent on the number of each computer type assembled.

To summarize, in order to use Excel's Solver you must have a formula to optimise (called the objective formula) and you must have cells that can be changed to optimise the objective. The cells to be changed should be precedents to the objective formula; that is, the calculation of the objective formula should depend on results in the precedent cells. If constraints are involved, the formulas to be subjected to the constraints should also be dependent on the changing cells.

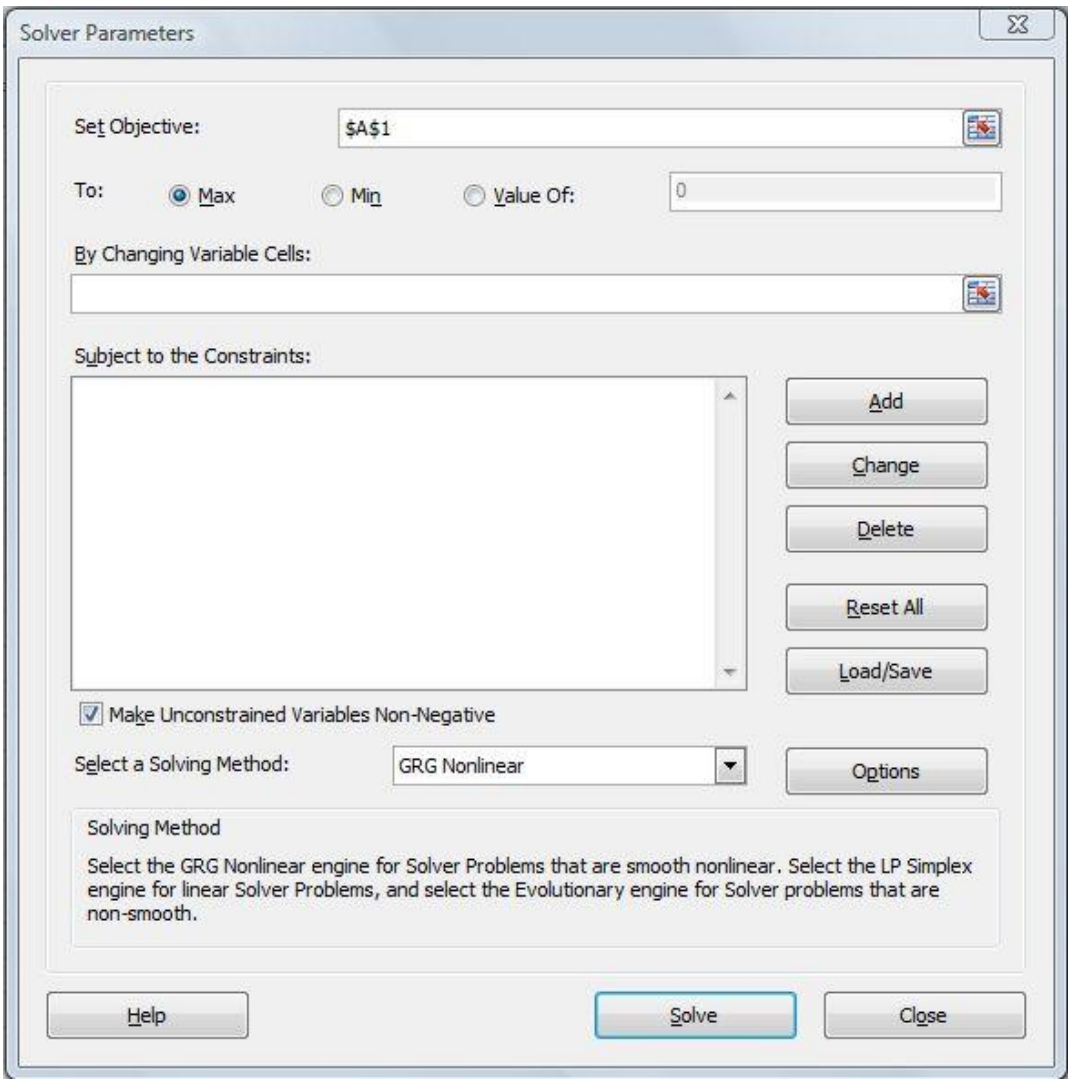
In the preceding worksheet, Cells G4 and H4 contain formulas that are subject to the constraints. Cell F6 contains the objective formula, and cells D4 and D5 are the changing cells.

You should notice that the formulas in cells G4, H4, and F6, are all dependent on the changing cells (either directly or indirectly).

Note: It is assumed throughout that there is enough demand to ensure that every computer made will be sold.

Setting Solver Parameters

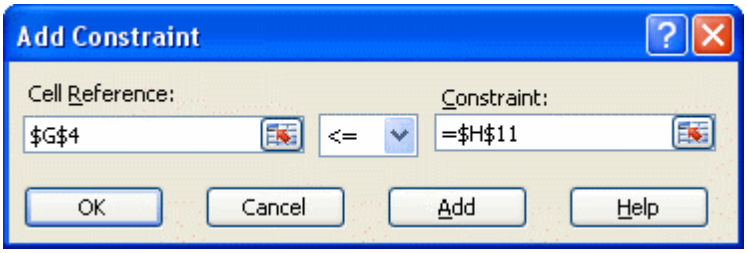
To use the Solver, click the Solver button on the Data Ribbon to display the Solver Parameters dialogue box.



Place your cursor in the Set Target Cell text box and click on the worksheet cell that contains your objective formula (cell F6 from the preceding worksheet).

Since we want to find the maximum total revenue, select the Max radio button next to the Equal to label. To enter the changing cells, place your cursor in the text area under the By Changing cells heading, and select the appropriate cells from the worksheet with your mouse. If they are non adjacent cells, press the Ctrl button when selecting them. For the problem shown here, the changing cells are D4 and D5.

The next step is to add the constraints by clicking the Add button to the right of the large white constraints area. This will display the Add Constraint box.



Place the cursor in the Cell Reference text field, and then select a cell with a formula you want to constrain. In this particular example, cell G4 is selected, which contains the formula for calculating the total software hours used. Follow the same process for the Constraint text field.

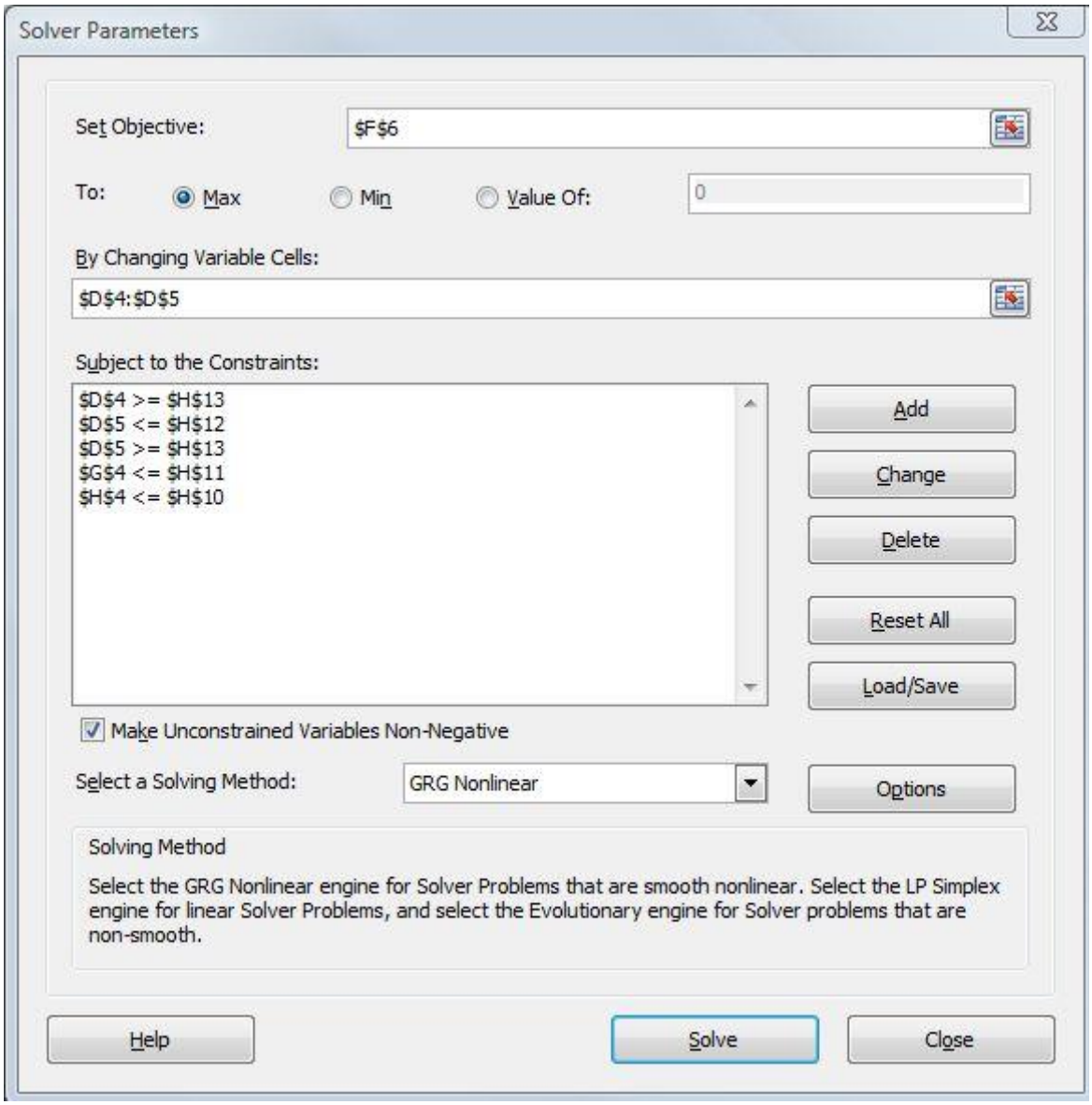
In this example, cell H11; containing the value 80 is entered. Next, use the drop down list in the center to specify the type of relationship required between the two cells. In this case, the constraint reads G4 <= H11 (that is, total software hours <= 80).

Click OK to enter the constraint into the Solver Parameters dialogue box.

Click the Add button again and follow the same process to enter the cell references for the rest of the constraints:

- Total hardware hours <= 30 (H4 <= H10)
- Number of Power PC's <= 10 (D5 <= H12)
- Number of Power PC's >= 0 (D5 >= H13)
- Number of Budget PC's >= 0 (D4 >= H13)

Here is the resulting Solver Parameters dialogue.

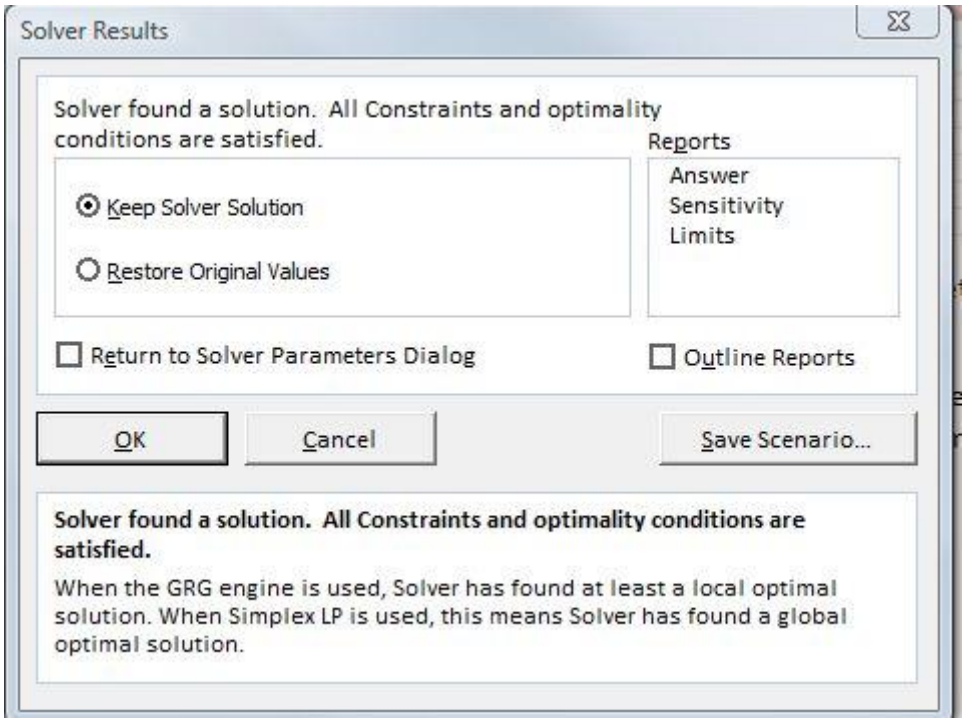


Here is the corresponding worksheet with formulas shown

	A	B	C	D	E	F	G	H
1		Computer Assembly						
2								
3		Hardware labor	Software labor	# Assembled	Price	Revenue	Total SW hours	Total HW hours
4	Budget PC	3	1		550	=D4*E4	=C4*D4+C5*D5	=B4*D4+B5*D5
5	Power PC	4	2		1200	=D5*E5		
6	Total			=D4+D5		=F4+F5		
7								
8								
9							Constraints	
10							Hardware Hrs <=	80
11							Software Hrs <=	30
12							# Power pc <=	10
13							# assembled >=	0

Examine the worksheet so that you understand the relationships between the target cell, the changing cells, and the constraints specified in the Solver Parameters box.

To implement the solver, click the Solve button in the Solver Parameters dialog. If you designed the worksheet correctly, and entered the correct cell references and constraints, you should see the following Solver Results box.



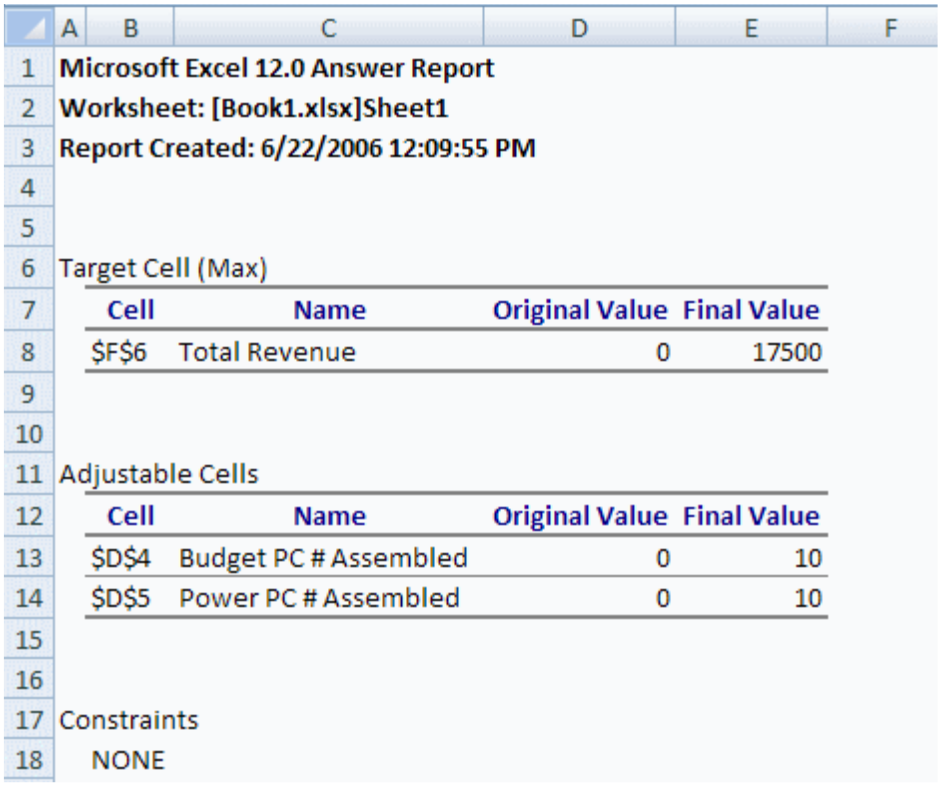
The values that maximize the total profit will now be visible in the changing cells (D4 and D5). The value of the maximum profit will be visible in the target cell (F6).

The Keep Solver Solution radio button will be selected by default. If you click OK the new values will remain in the worksheet. If you select Restore original values, the solutions that the solver found will not be entered into the worksheet and the original values will be retained.

You have the option to save the solver results as a scenario that you can name and reload into the worksheet at a later date. You can also select one or more report

types from the list at the right of the Solver Results box. These formatted reports will be generated on separate worksheets.

This following image shows an answer report based on the Solver solution. It is generated on a separate worksheet if you select Answer from the report list in the solver results box.



Microsoft Excel 12.0 Answer Report
Worksheet: [Book1.xlsx]Sheet1
Report Created: 6/22/2006 12:09:55 PM

Target Cell (Max)

Cell	Name	Original Value	Final Value
F6	Total Revenue	0	17500

Adjustable Cells

Cell	Name	Original Value	Final Value
D4	Budget PC # Assembled	0	10
D5	Power PC # Assembled	0	10

Constraints

NONE

This is the same worksheet after the price of the Budget PC has been changed to 750; and the Solver has been applied with the same target, changing cells, and constraints as before. Notice that because the price has been changed, the solution is now 20 Budget PC's and 5 Power PC's for a total revenue of 21000.

	A	B	C	D	E	F	G	H
1		Computer Assembly						
2								
3		Hardwar	Software labor	# Assembled	Price	Revenue	Total SW hours	Total HW hours
4	Budget PC	3	1	20	750	15000	30	80
5	Power PC	4	2	5	1200	6000		
6	Total			25		21000		
7								

Module Six: Macros

A macro is a set of recorded computer instructions. These instructions are associated with a shortcut key or macro name that makes it easy to tell your computer to run that set of instructions. This module will explain how to save time with macros. You'll learn how to display the Developer tab, which contains the tools you'll need to record macros. You'll learn how to record and run macros. This module also explains macro security levels to avoid allowing malicious content to damage your computer with macros. Finally, you'll learn how to customize and change the Quick Access Toolbar so that you have instant access to your favorite macros.

What is a macro?

A macro can be described as a tiny program that uses Visual Basic code to automate a sequence of actions or instructions.

A macro can be simple and consist of only a few tasks or commands, or be quite complex, involving lots of data manipulation and calculations. To create complex macros, it is worthwhile to invest time in learning Visual Basic programming. But for simple macros, you don't have to do any programming at all.

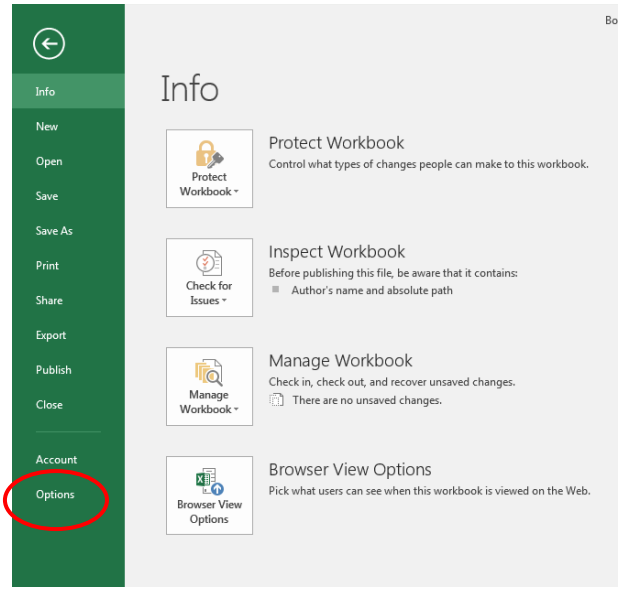
Simple macros are great for any long sequence of keystrokes that you find yourself repeating often, or for combinations of basic Excel actions that would be convenient to automate for a particular workbook. Macros can be saved with the workbook in which they were created, or they can be saved in a separate personal macro workbook where they are more accessible.

Displaying the Developer Tab

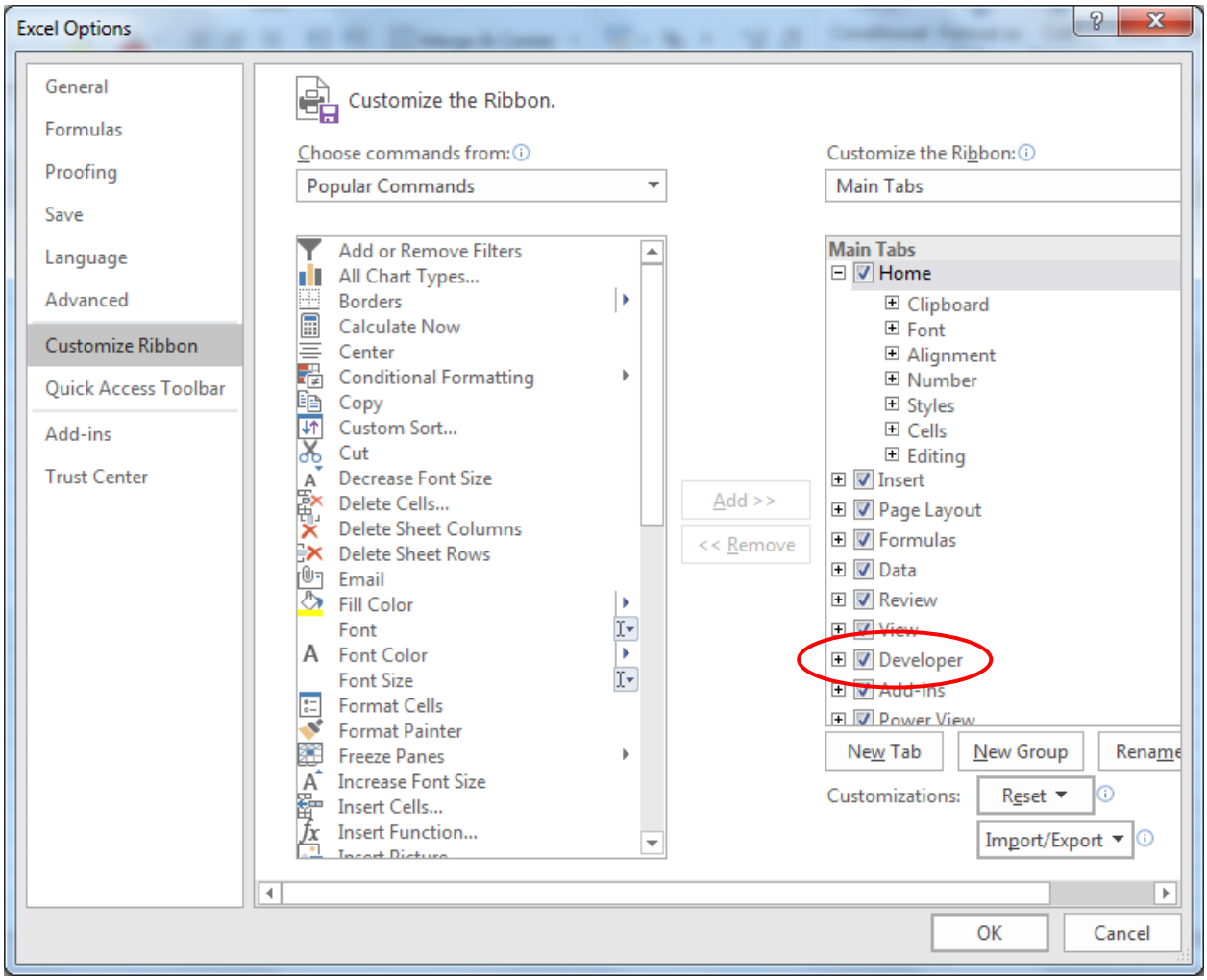
The Developer tab is where the tools involving macros are located. The Developer tab is not displayed by default, so you'll need to let Excel know that you want to work with it. You use the Options dialog box to display the Developer tab.

Use the following procedure to display the Develop tab.

1. Select the **File** tab from the Ribbon.
2. Select **Options**.

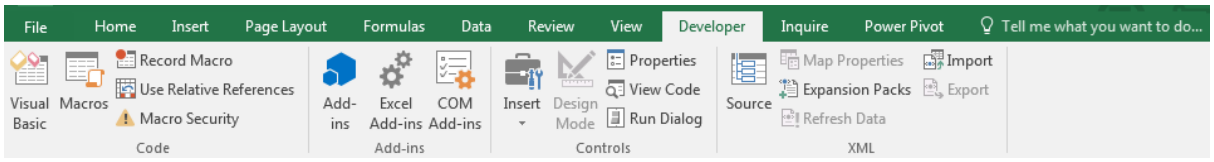


3. Select **Customize the Ribbon**.
4. In the **Customize the Ribbon** area on the right, check the **Developer** box.



5. Select **OK**.

Developer tab



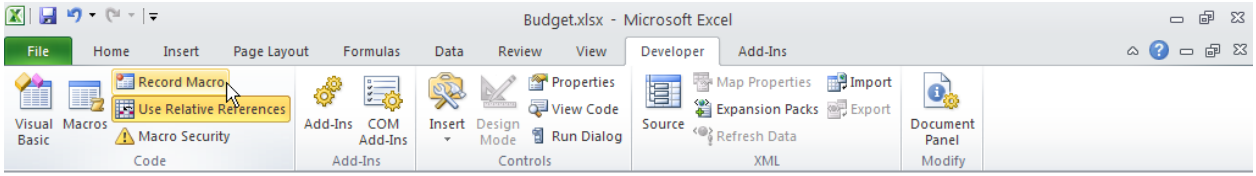
Recording and Running Macros

With Excel macros can be created by turning on the macro recorder, then performing a series of actions to be recorded, and then turn off the recorder. Excel stores the recording and makes it accessible to any Excel worksheet. You can record any menu action or command to make it part of the macro.

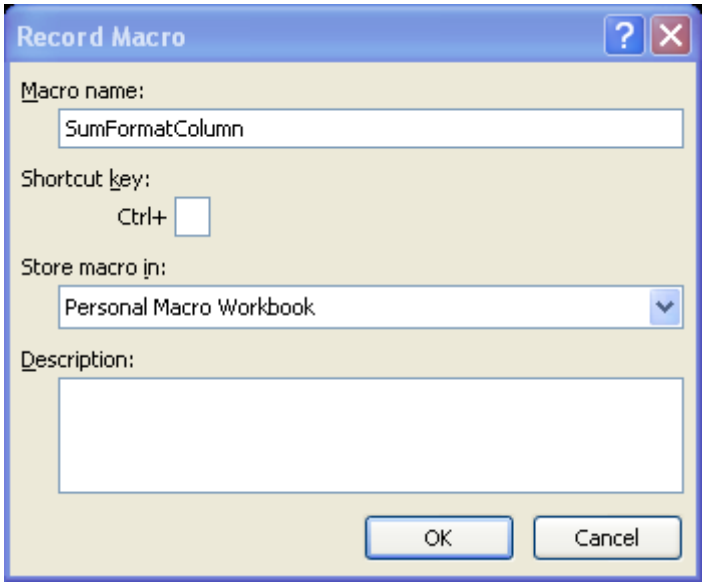
To record a macro, you turn on the recorder, perform the actions you want to record, then you turn off the recorder.

Use the following procedure to record a macro. In this example, we will sum the column and add formatting to the numbers.

1. Select the **Developer** tab from the Ribbon.
2. Select **Use Relative References**.
3. Select **Record Macro**.



4. In the Record macro dialog box, give your macro a name.

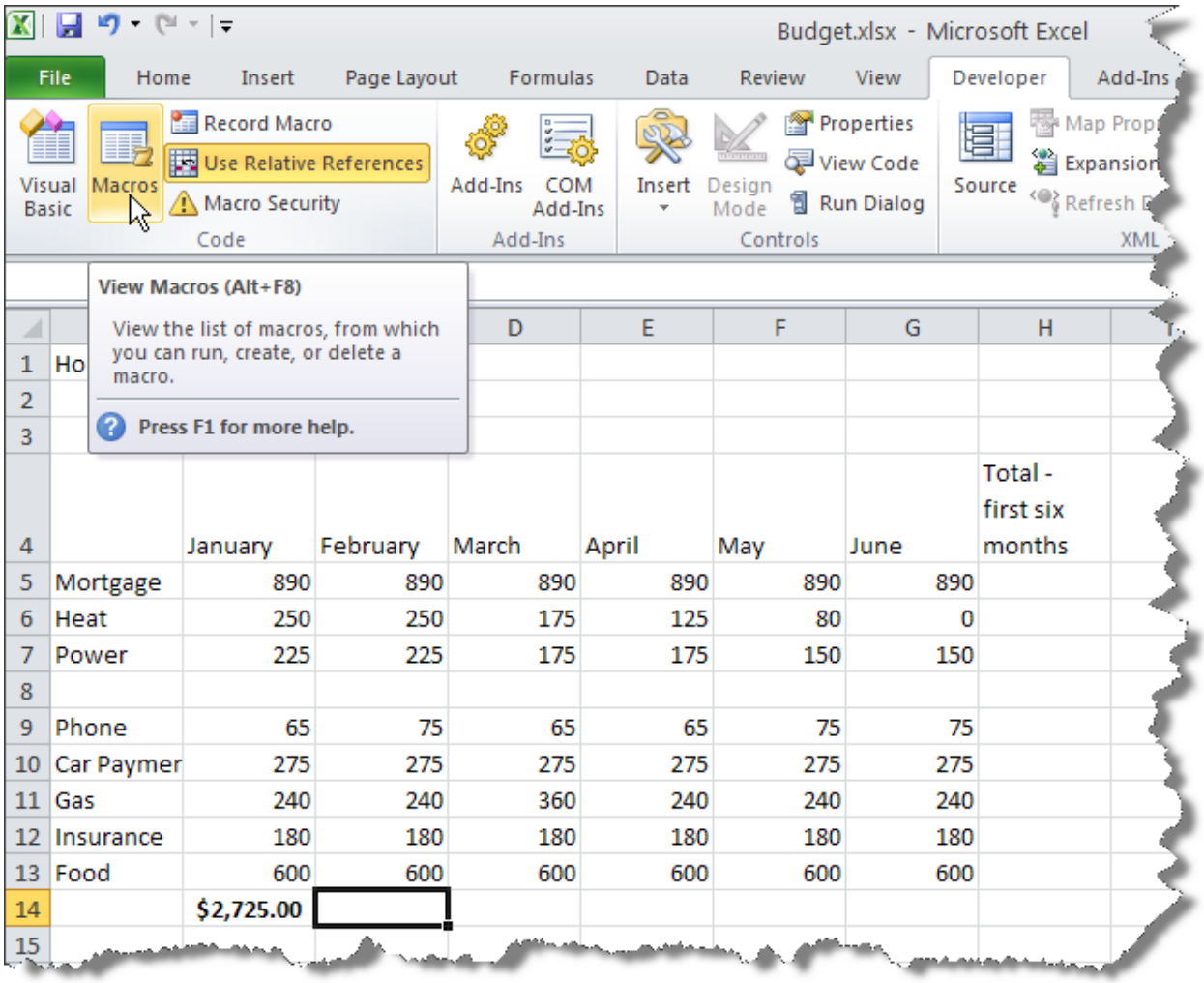


5. To make the macro available to other worksheets, select **Personal Macro Workbook** from the **Store Macro In** drop down list.
6. Select **OK** to begin recording.
7. Perform the actions you want to record. In this example, we inserted a Sum and then formatted the total with a currency formatting and added bold face formatting.
8. Select the **Developer** tab.

9. Select **Stop Recording**.

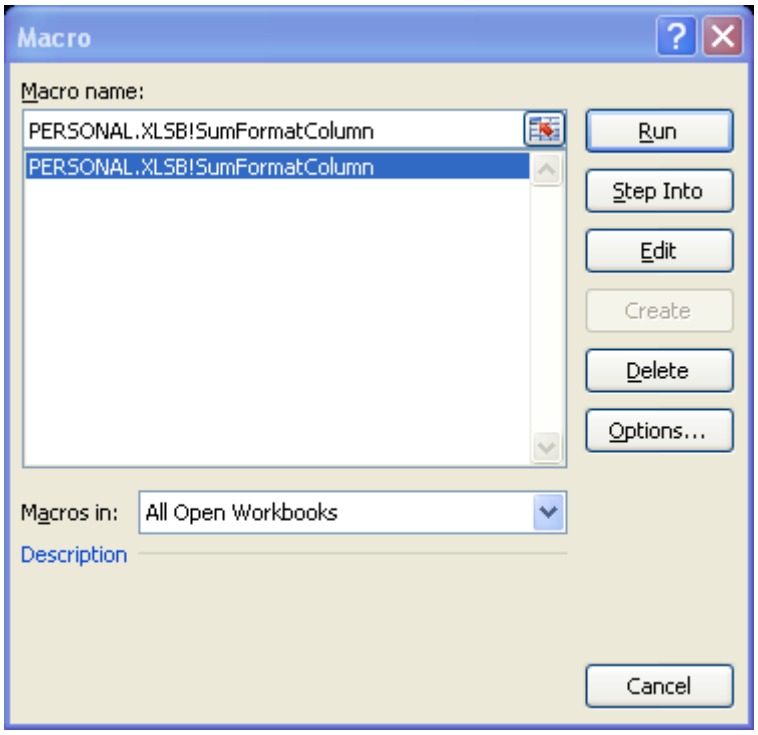
Use the following example to run a macro

1. Place your cursor in the cell where you want to perform the macro.
2. Select the **Developer** tab.
3. Select **Macros**.



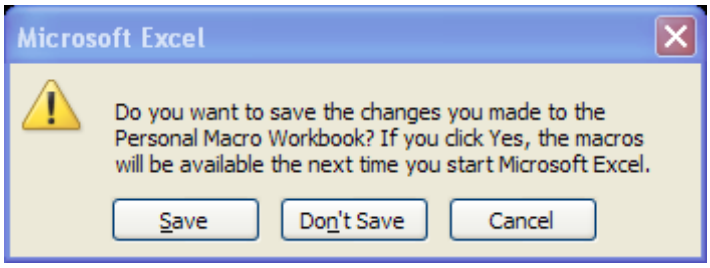
		January	February	March	April	May	June	Total - first six months
5	Mortgage	890	890	890	890	890	890	
6	Heat	250	250	175	125	80	0	
7	Power	225	225	175	175	150	150	
8								
9	Phone	65	75	65	65	75	75	
10	Car Paymer	275	275	275	275	275	275	
11	Gas	240	240	360	240	240	240	
12	Insurance	180	180	180	180	180	180	
13	Food	600	600	600	600	600	600	
14		$2,725.00						

4. In the Macro dialog box, select your macro name from the list.



5. Select **Run**.

Note that when you close Excel, you will get the following warning message.



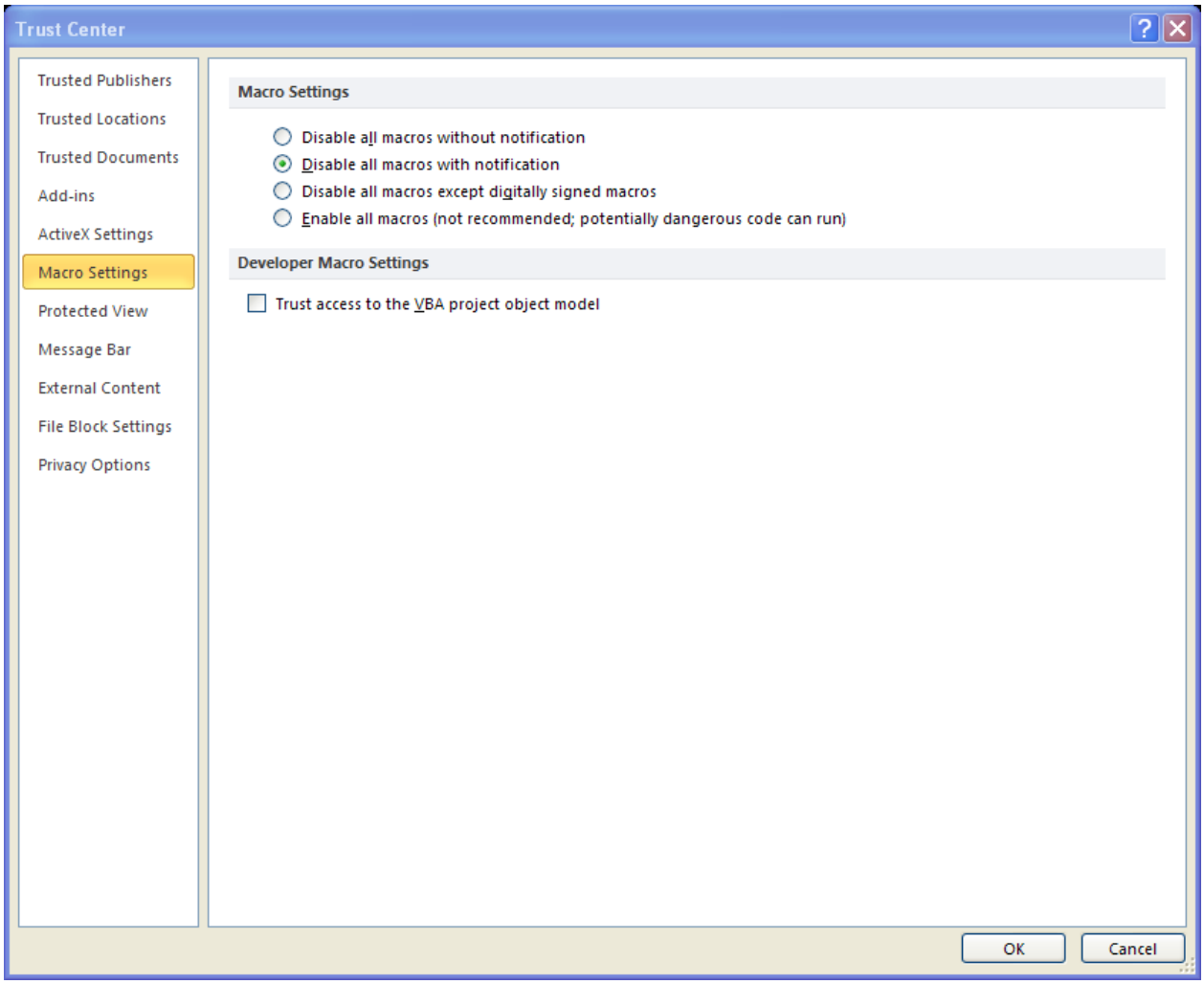
Select **Save** to keep the macro and make it available to other workbooks.

Changing the Security Level

Excel security may prevent some of your macros from working, depending on where the workbooks containing the macros are located. You can change the security settings to avoid warning messages each time your run certain macros.

Use the following procedure to change the macro security.

1. Select the **Developer** tab.
2. Select **Macro Security**.
3. Select one of the following options:
    - Disable all macros without notification – this option only runs macros in documents in trusted locations.
    - Disable all macros with notification – this option disables macros that are not in trusted locations, but it provides notification, so that you can choose to enable those macros on a case by case basis.
    - Disable all macros except digitally signed macros – this option allows not only macros in trusted locations, but also macros that are digitally signed by a trusted publisher. Other macros are disabled with notification to allow you to choose to enable those macros on a case by case basis.
    - Enable all macros – this option allows all macros to run, which is potentially dangerous since virus authors often use macros to distribute malicious code on computers. Microsoft does not advise using this setting.



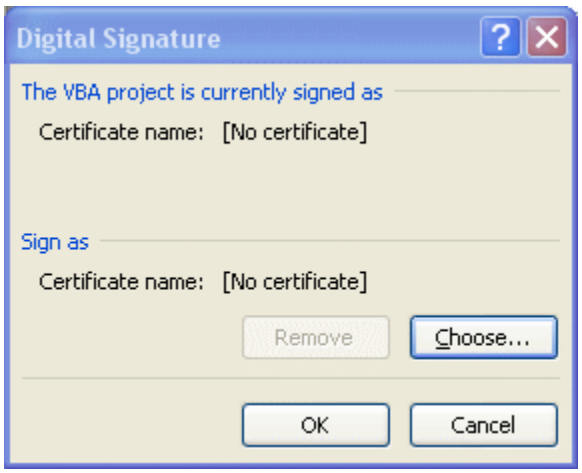
4. Check the Trust Access to the VBA Project Object Model box only if you are a developer. This security option makes it more difficult for unauthorized programs to build code that self-replicates.
5. Select **OK**.

To digitally sign a macro, click the Developer tab to display the Developer Ribbon and then click the Visual Basic button to launch the Visual Basic Editor.

In the tool bar at the top of the Visual Basic Editor's window, click the Project Explorer button.

In the navigation panel on the left of the editor window, highlight the VBA Project/Worksheet containing the macro you want to sign, and then click the Digital Signature option from the Visual Basic Editor's Tools menu.

This will display a Digital Signature dialogue box.



Click the Choose button to display a list of available certificates. When you are finished, click
the OK button.

Recording a Relative Reference Macro

By default, macros employ absolute cell referencing. This means that a macro's actions are performed on the same cells every time the macro is run.

If you specify relative cell referencing for a macro, the actions performed by the macro will be relative to the active cell when you start the macro. This means that the macro can perform actions on different cells each time it is run, depending on what cell is used as the starting point.

To record a relative reference macro, choose your active cell and click the Use Relative References button before you begin recording.

Once you click this button, simply click the Record Macro button as before to create the actual macro. When you stop the macro recording process, all of the cells that were involved in the macro will be treated as having relative references.

Let's say, for example, that the active cell is B1 and you create a relative reference macro that shades the cell C1 in blue. If you make cell K10 the active cell and run the macro, cell L10 will be shaded in blue.

If this were an absolute reference macro, cell C1 will become shaded in blue no matter what the active cell is when you run the macro.

Running a Relative Reference Macro

To run a relative reference macro, first make sure that the active cell is chosen such that the cells that you want the macro to run on are in the correct relative position.

As an example, suppose that you created your relative reference macro with A1 as the active cell. The macro that you created changes the column of six cells immediately below A1 (A2:A7) to accounting format.

	A	B
1	Amounts	
2	100	
3	200	
4	300	
5	400	
6	500	
7	600	

Here is the data after recording the macro.

	A	B
1	Amounts	
2	$ 100.00	
3	$ 200.00	
4	$ 300.00	
5	$ 400.00	
6	$ 500.00	
7	$ 600.00	

Now, if you wanted to convert cells H3:H8 to accounting format, first pick cell H2, which is in
the same relative position to H3:H8 as cell A1 was to A2:A7.

G	H
	1000
	2000
	3000
	4000
	5000
	6000

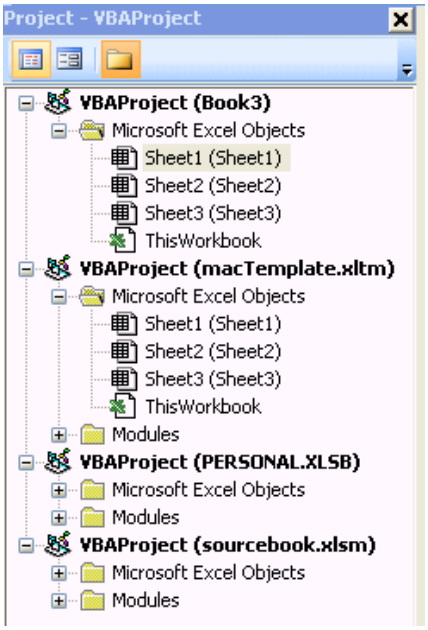
Now, click the Macro button and choose the correct macro from the Macro dialogue box.

Copying a Macro from a Workbook or Template

If you have an existing macro in a macro enabled workbook or a macro enabled template and you would like to use that macro in another workbook, you can copy the macro from the source workbook or template into the workbook of your choice.

First, you must use Excel to open the template or workbook containing the macro, as well as the workbook that you are copying the macro into. When both workbooks are open, select the Developer tab on the destination workbook and click the Visual Basic button to display the Visual Basic Editor.

When the Visual Basic Editor opens, you will see all of the currently open workbooks in the
Project Explorer panel on the right.

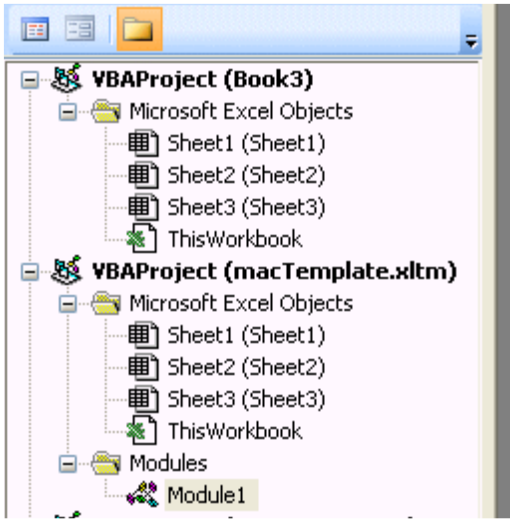


(If you cannot see the Project Explorer panel, click the Project Explorer button in the Visual Basic Editor Toolbar.)

When you record a macro for a workbook, it will be contained in a code module shown in the Visual Basic Editor. In the image shown above you can see that Book3 (the destination workbook) has no modules, but the other open workbooks and templates (macTemplate.xltm and sourcebook.xlsm) have module folders with them.

To copy a macro from a template or a workbook, open the module folder for the workbook or template that you want to copy from by clicking on the plus (+) sign corresponding to the folder.

In the following image, the modules folder for macTemplate.xltm has been opened.



To copy the macro from macTemplate.xltm to Book3, just select and drag Module1 from the open Modules folder up to the Book3 area (drag until you see your mouse pointer turn to a plus (+).

Now, both Book3 and macTemplate.xltm have code modules associated with them.

If you double click on the Module1 icon under Book3 and double click on the Module1 icon under macTemplate.xltm, you will see code windows for both modules (and the macros contained in the modules) open in the editor.

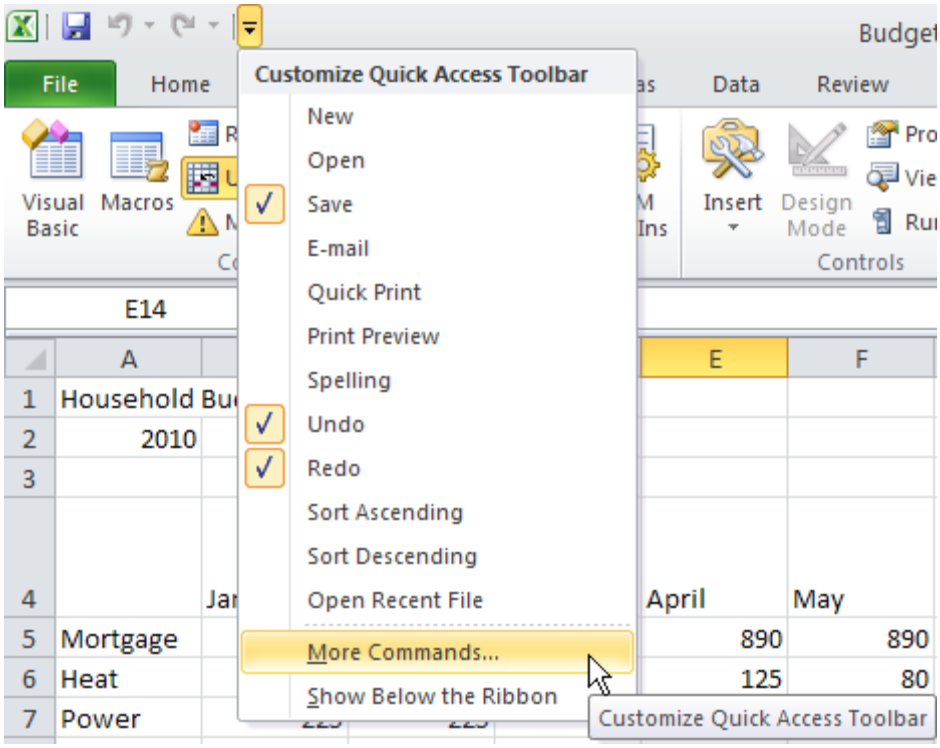
If you examine the VBA code for both modules (macros) in the editor window, you will see that they are identical.

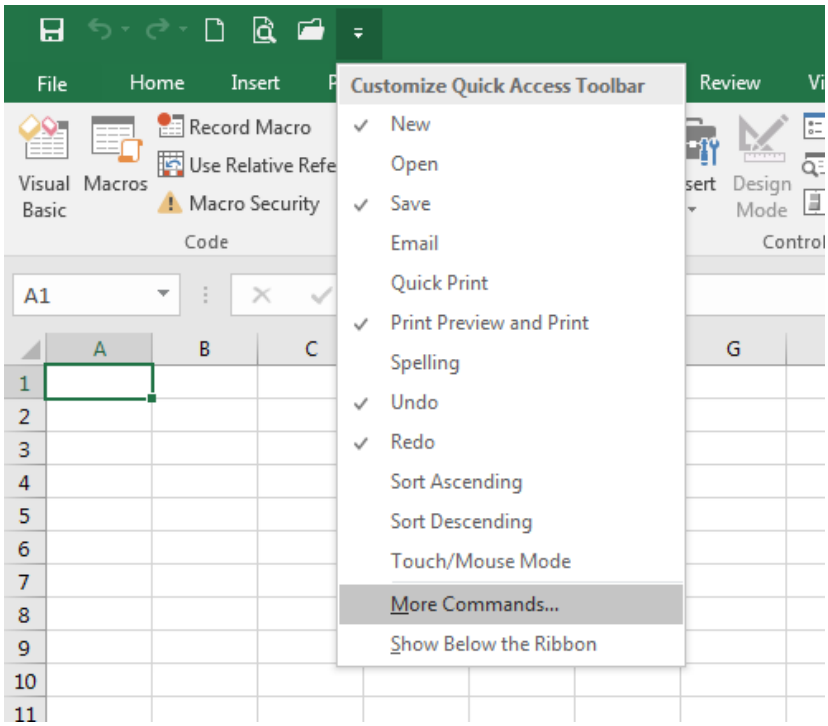
If you save Book3 as a macro enabled workbook, the process of copying the macro will be complete!

Customizing and Changing the Quick Access Toolbar

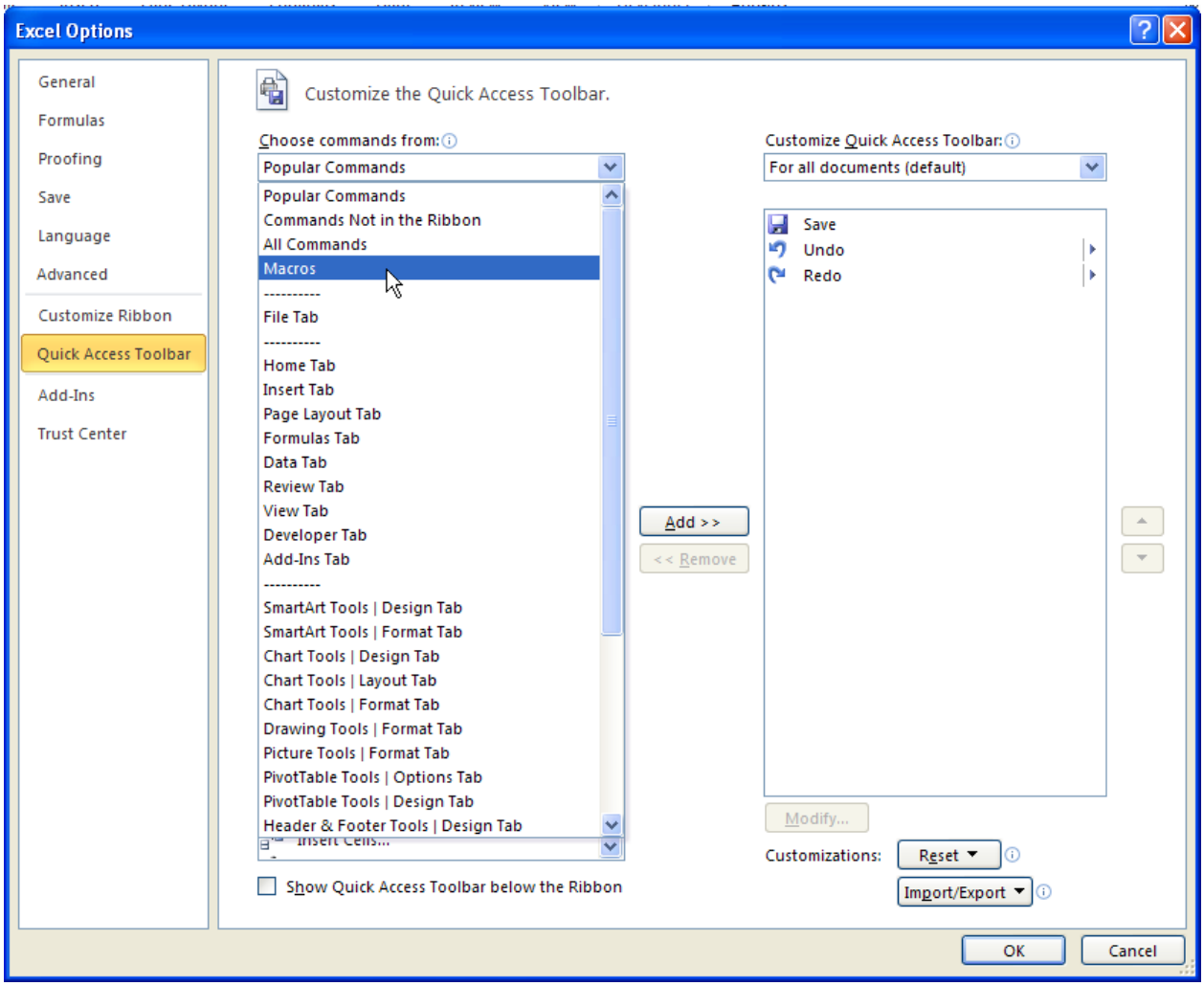
Use the following procedure to add a macro to the Quick Access Toolbar.

1. Select the arrow to the right of the Quick Access Toolbar.
2. Select **More Commands**.

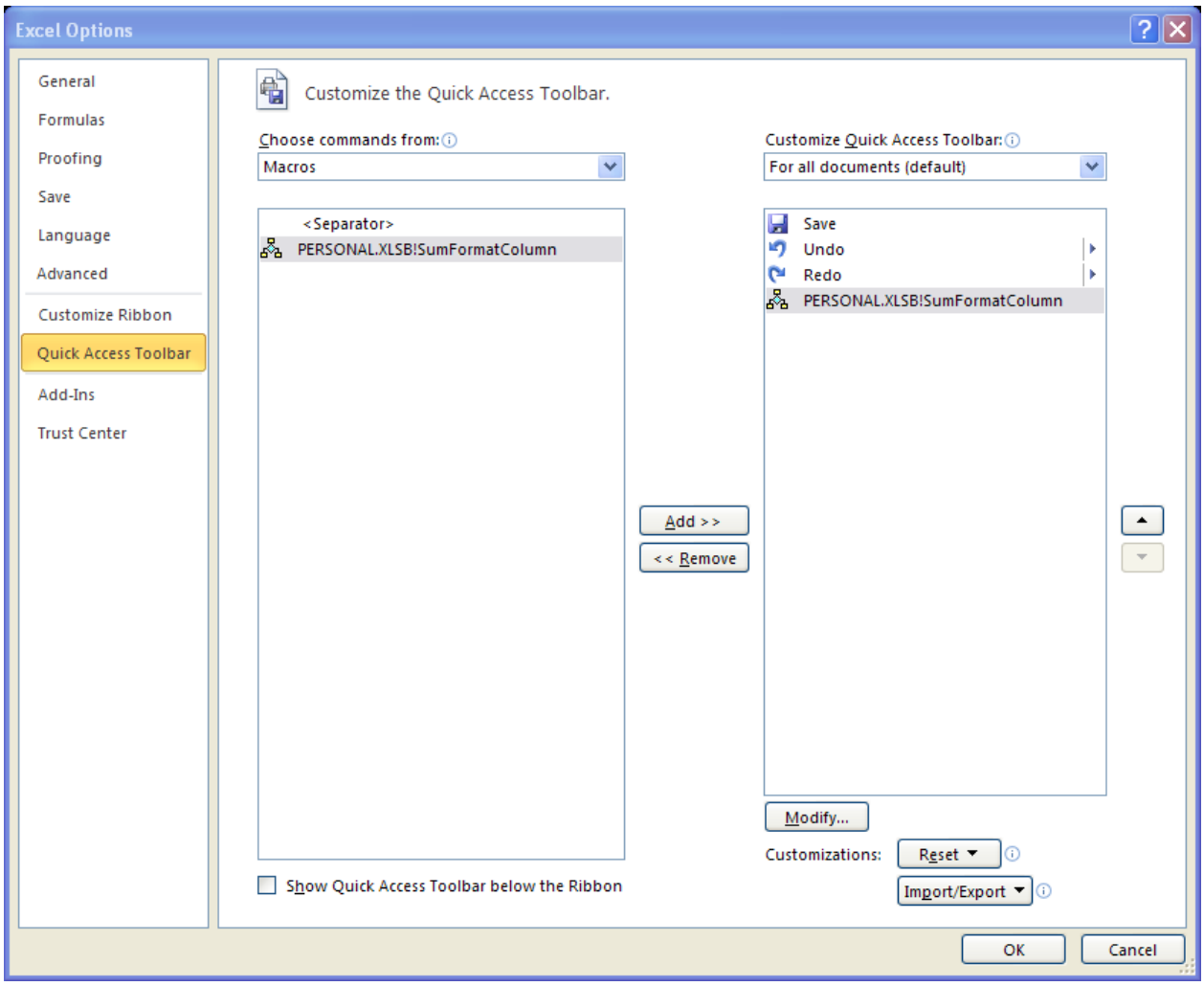




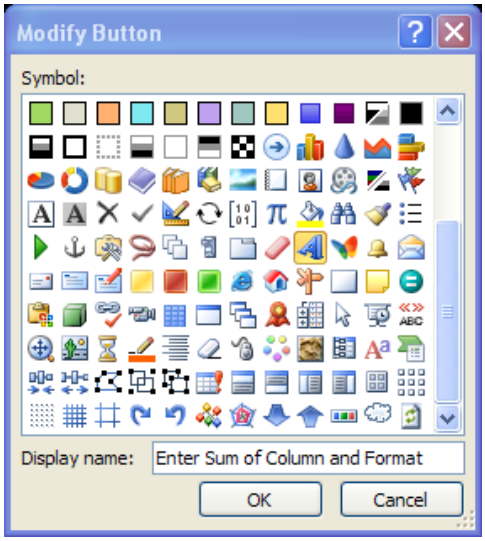
3. In the **Choose Command From** drop down list, select **Macros**.



4. The macro you recorded should be listed. Select it and select Add.



5. If you would like to modify the name of the macro, select **Modify**.
6. In the *Modify Button* dialog box, you can choose an icon to show in the Quick Access Toolbar. You can also modify the name.



7. Select **OK**.
8. Select OK in the *Excel Options* window.

E&OE

STL Training reserves the right to revise this publication and make changes from time to time in its content without notice.