Excel 2013 Advanced

Contents

Unit 1: Using Logical Functions

In this unit, you will learn how to:

- Define range names and use names in formulas
- Identify and apply Excel's logical functions, IF, AND, OR
- Create nested functions by combining logical functions in a formula

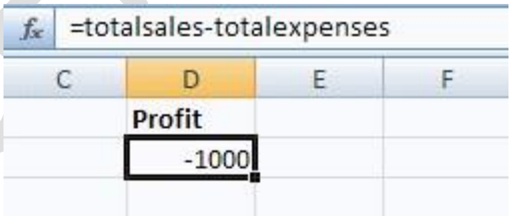
Working with Names and Ranges

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.



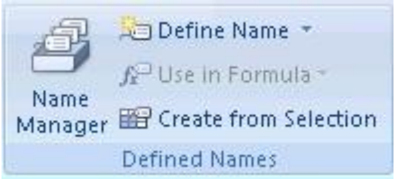
=totalsales-totalexpenses

C	D	E	F
	Profit		
	-1000		

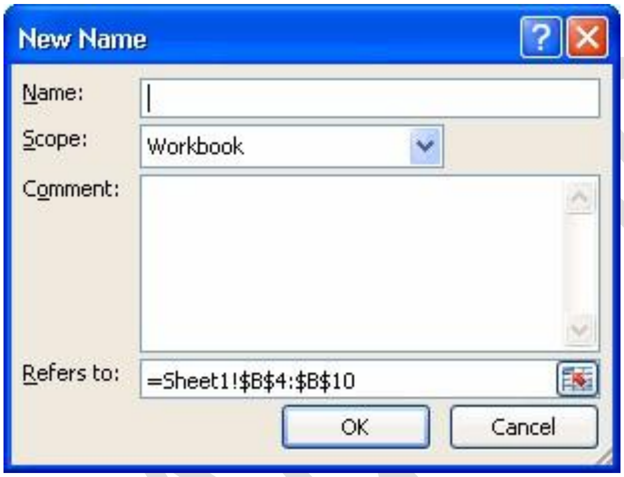
Range names will make formulas much more readable and they will make it easier to find and reference individual cells. When you are first designing your worksheet, you can create formulas using names instead of traditional cell references, and then define the names for the corresponding ranges of data 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.



Name:

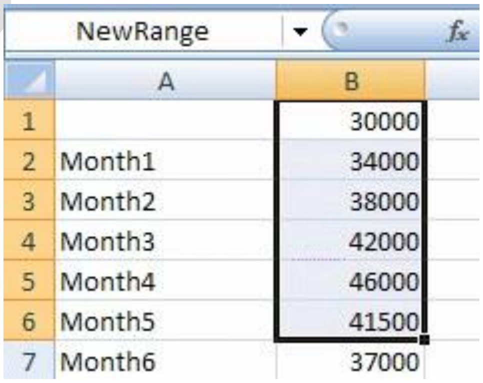
Scope: Workbook

Comment:

Refers to: =Sheet1!$B$4:$B$10

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.



NewRange

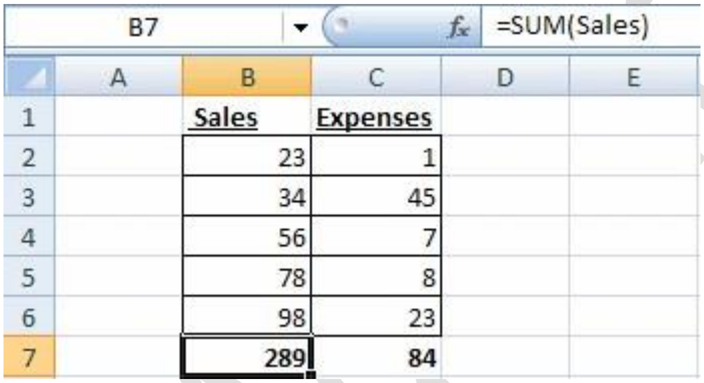
	A	B
1		30000
2	Month1	34000
3	Month2	38000
4	Month3	42000
5	Month4	46000
6	Month5	41500
7	Month6	37000

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, “new range” or “new Range” 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.



B7 =SUM(Sales)

	A	B	C	D	E
1		Sales	Expenses		
2		23	1		
3		34	45		
4		56	7		
5		78	8		
6		98	23		
7		289	84		

In the above image, notice the function =SUM (sales) in the formula bar. This will produce the same result as =SUM (B2:B7).

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 2013’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 unit 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)

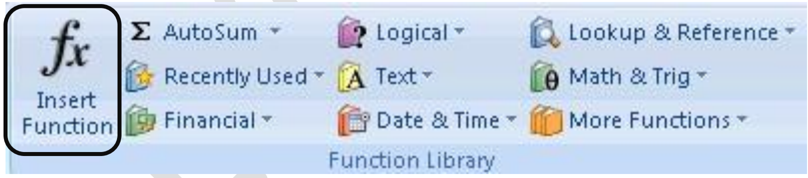
This function states that if the contents of cell A1 is less than or equal to 100, the value in cell B2 will be the value in A1 multiplied by .5; else, the value in B2 will be the value of cell C3 multiplied by 2.

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.

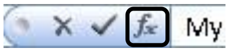
Using the Function Library

Excel 2013 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.

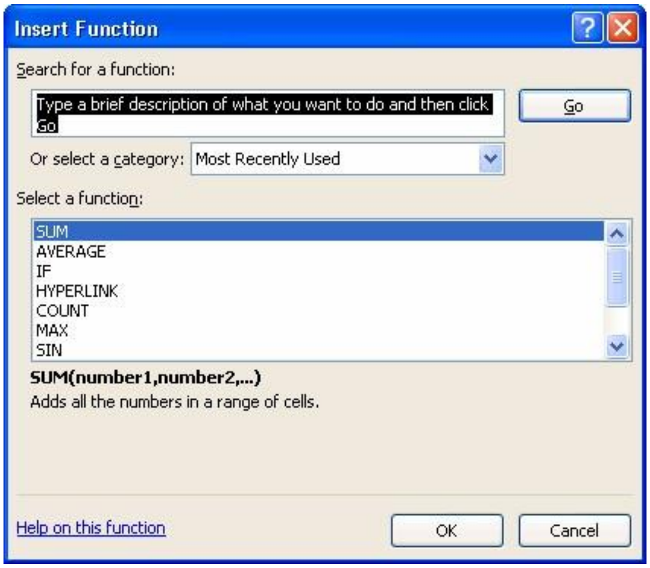


The first and largest button is Insert Function. This button will open a dialog allowing you to search for and insert hundreds of functions.

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.



Insert Function

Search for a function:

Type a brief description of what you want to do and then click Go

Or select a category: Most Recently Used

Select a function:

SUM
AVERAGE
IF
HYPERLINK
COUNT
MAX
SIN

SUM(number1,number2,...)

Adds all the numbers in a range of cells.

Once the Insert Function dialogue box is open:

Select the function you wish to use from the available list and click OK or

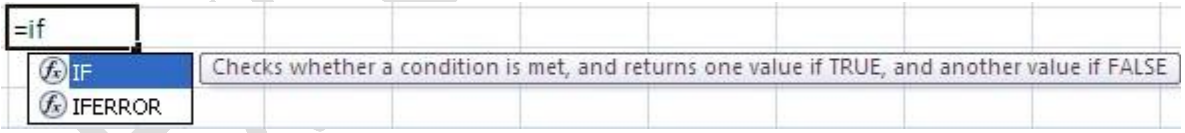
Type the name of the function you wish to use in the Search for a function area, press Enter, select the function when it appears in the list and click OK.

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 2013, 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

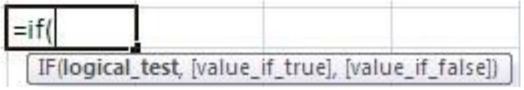
IF

IFERROR

Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE

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

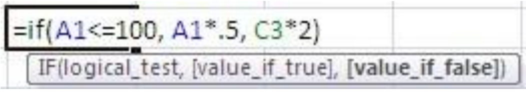
Once you have selected the function you wish to use, continue by typing in a left bracket, which will cause the function syntax to display in another screen tip.



=if(

IF(**logical_test**, [value_if_true], [value_if_false])

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

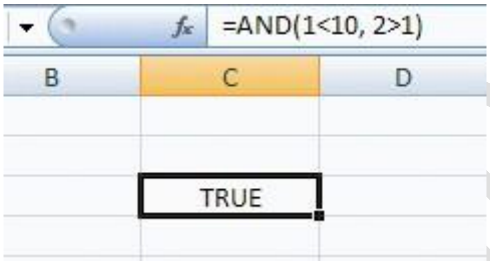


=if(A1<=100, A1*.5, C3*2)

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.



=AND(1<10, 2>1)

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

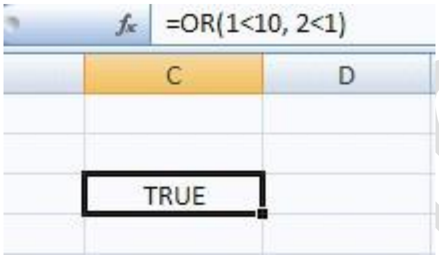
The OR Function

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.



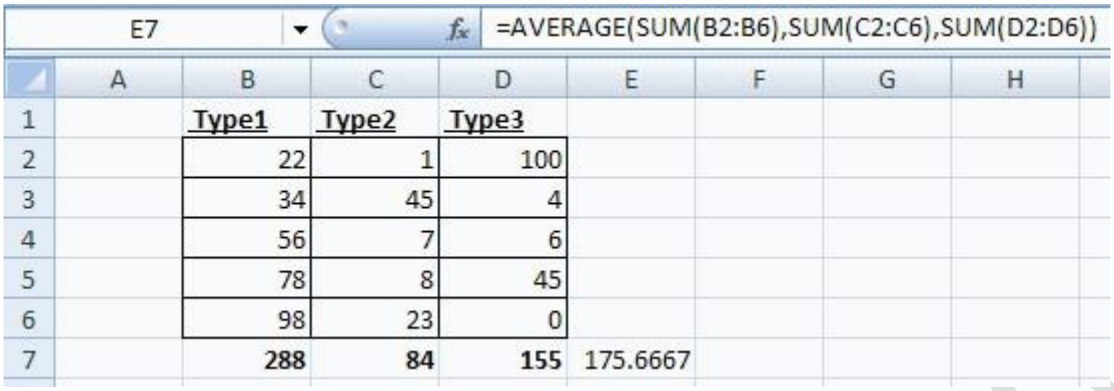
=OR(1<10, 2<1)

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. Look at the following worksheet as an example.



E7 =AVERAGE(SUM(B2:B6),SUM(C2:C6),SUM(D2:D6))

	A	B	C	D	E	F	G	H
1		Type1	Type2	Type3				
2		22	1	100				
3		34	45	4				
4		56	7	6				
5		78	8	45				
6		98	23	0				
7		**288**	**84**	**155**	175.6667			

By looking in the formula bar, you can see that the average function for cell E7 contains three sum functions (the sums of the three type columns). This means that the value in E7 is the average of the sums of columns B, C, and D.

The following function is also possible:

=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.

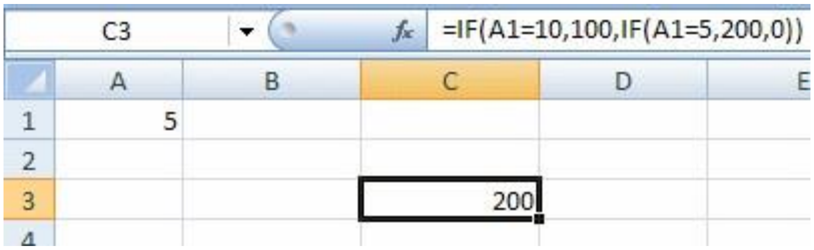
In terms of nested functions, nesting averages within sums and sums within averages is probably not that practical; however, nested IF functions can be extremely useful for a wide variety of situations.

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=5,200,0))

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 return 0.

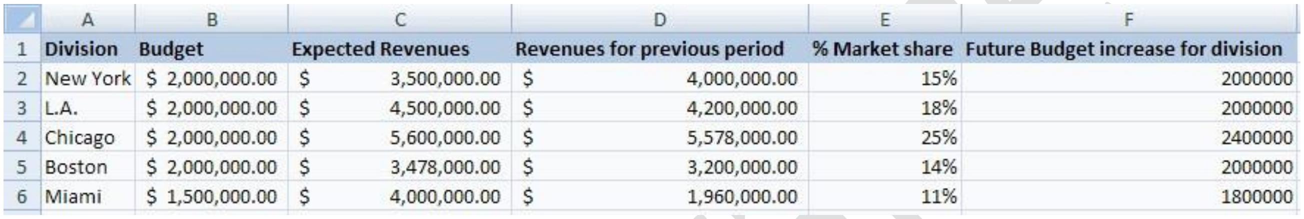


C3 =IF(A1=10,100,IF(A1=5,200,0))

	A	B	C	D	E
1	5				
2					
3			200		
4					

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.

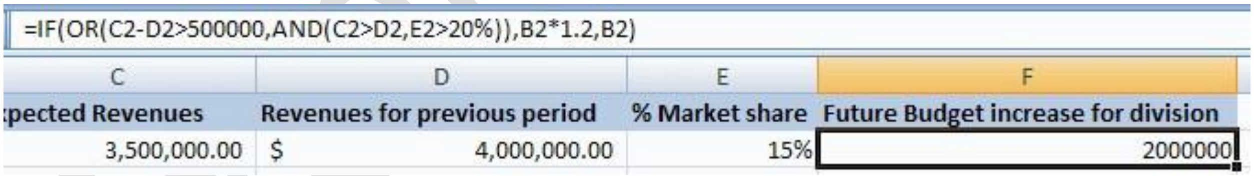
Applying Logical Functions



	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.



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

C	D	E	F
:pected Revenues	Revenues for previous period	% Market share	Future Budget increase for division
3,500,000.00	$ 4,000,000.00	15%	2000000

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:

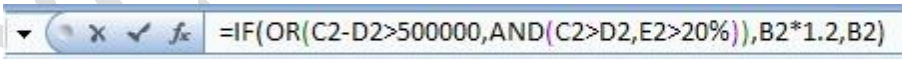
If (%Market Share is greater than 20% and Expected Revenues are greater than Revenues for Previous period) or (Expected Revenues subtract Revenues for Previous period is greater than 500000) return true.

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

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 percent, the budget will be increased by 20 percent. Otherwise, the budget will remain the same.

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.

Unit 2: Working with Lookup Functions

In this unit, you will learn how to:

Identify and use the VLOOKUP function
Identify and use the HLOOKUP function

Excel 2003 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).

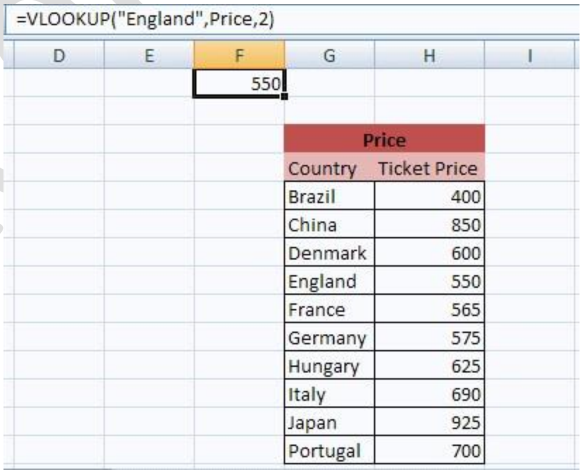
The VLOOKUP function will look in the leftmost column of a table for a value you specify. When it finds the value you specified, it will return a value that is located in the same row, a specified number of columns into the table. It is called VLOOKUP because it looks vertically down a column for a match,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



=VLOOKUP("England",Price,2)

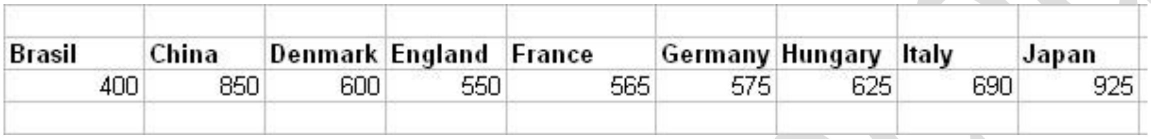
D	E	F	G	H	I
		550			
			Price		
			Country	Ticket Price	
			Brazil	400	
			China	850	
			Denmark	600	
			England	550	
			France	565	
			Germany	575	
			Hungary	625	
			Italy	690	
			Japan	925	
			Portugal	700	

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.



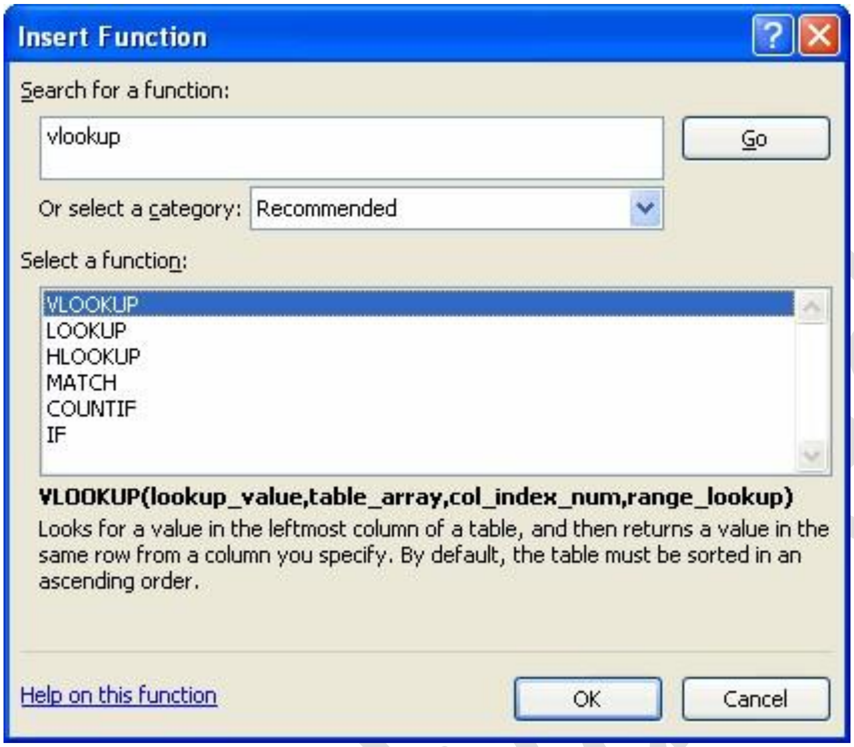
Brasil	China	Denmark	England	France	Germany	Hungary	Italy	Japan
400	850	600	550	565	575	625	690	925

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.



Search for a function: vlookup

Or select a category: Recommended

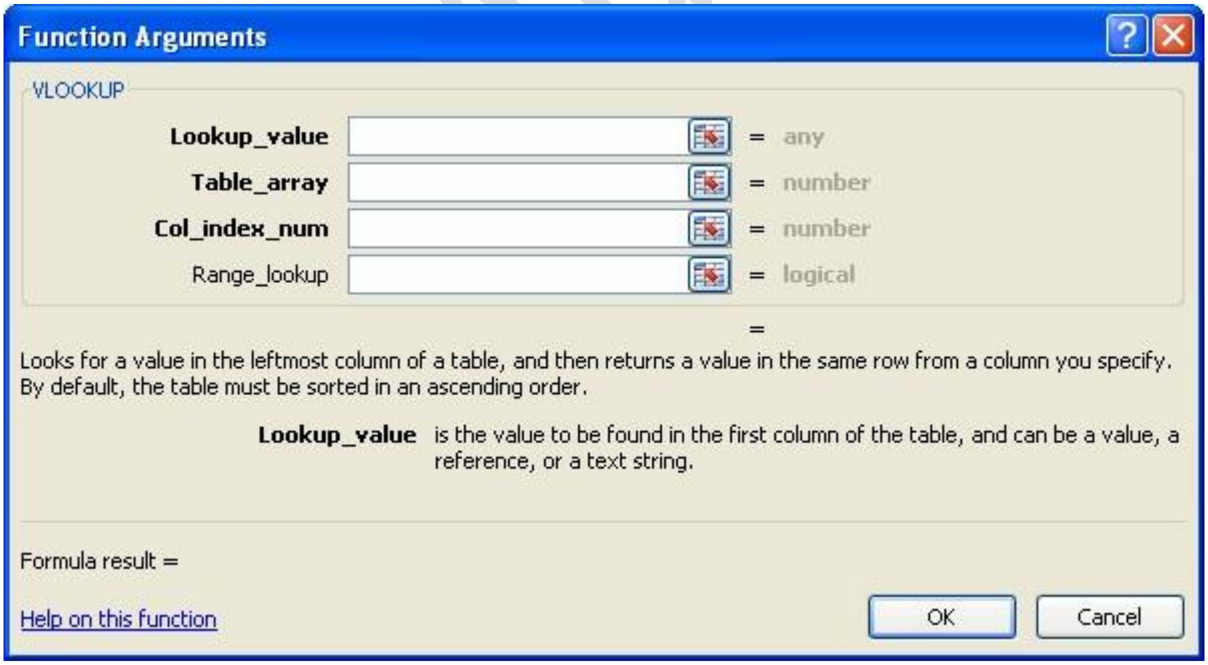
Select a function:

- VLOOKUP
- LOOKUP
- HLOOKUP
- MATCH
- COUNTIF
- IF

VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)

Looks for a value in the leftmost column of a table, and then returns a value in the same row from a column you specify. By default, the table must be sorted in an ascending order.

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.



Lookup_value	= any
Table_array	= number
Col_index_num	= number
Range_lookup	= logical

=

Looks for a value in the leftmost column of a table, and then returns a value in the same row from a column you specify. By default, the table must be sorted in an ascending order.

Lookup_value is the value to be found in the first column of the table, and can be a value, a reference, or a text string.

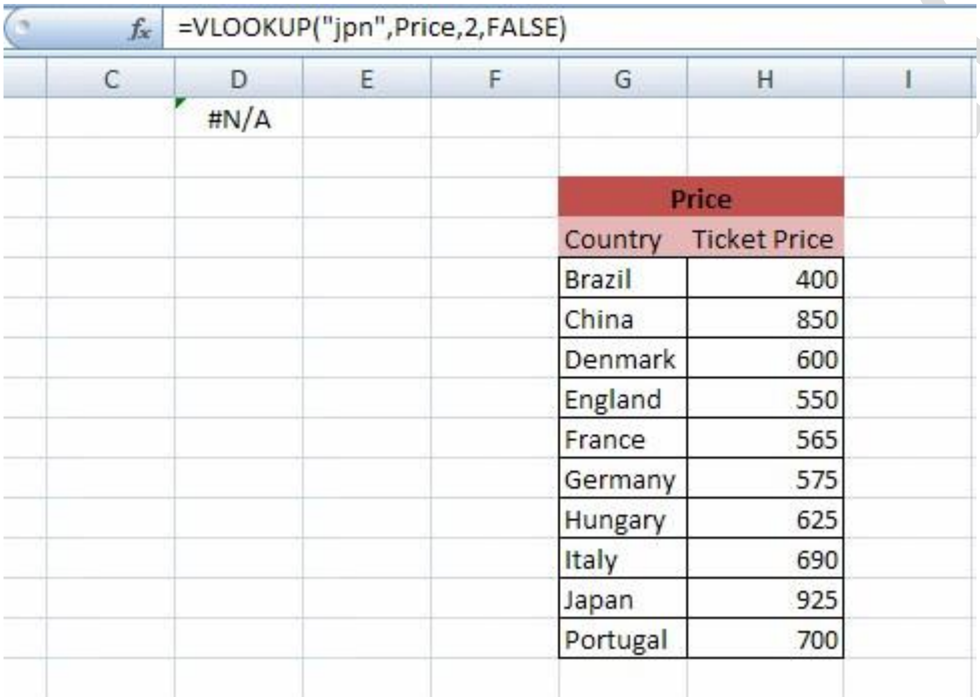
Formula result =

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.



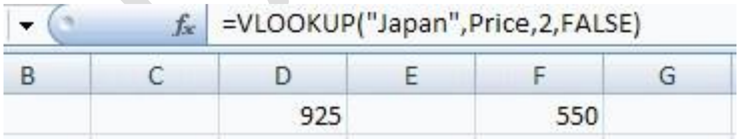
=VLOOKUP("jpn",Price,2,FALSE)

C	D	E	F	G	H	I
	#N/A					

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.



=VLOOKUP("Japan",Price,2,FALSE)

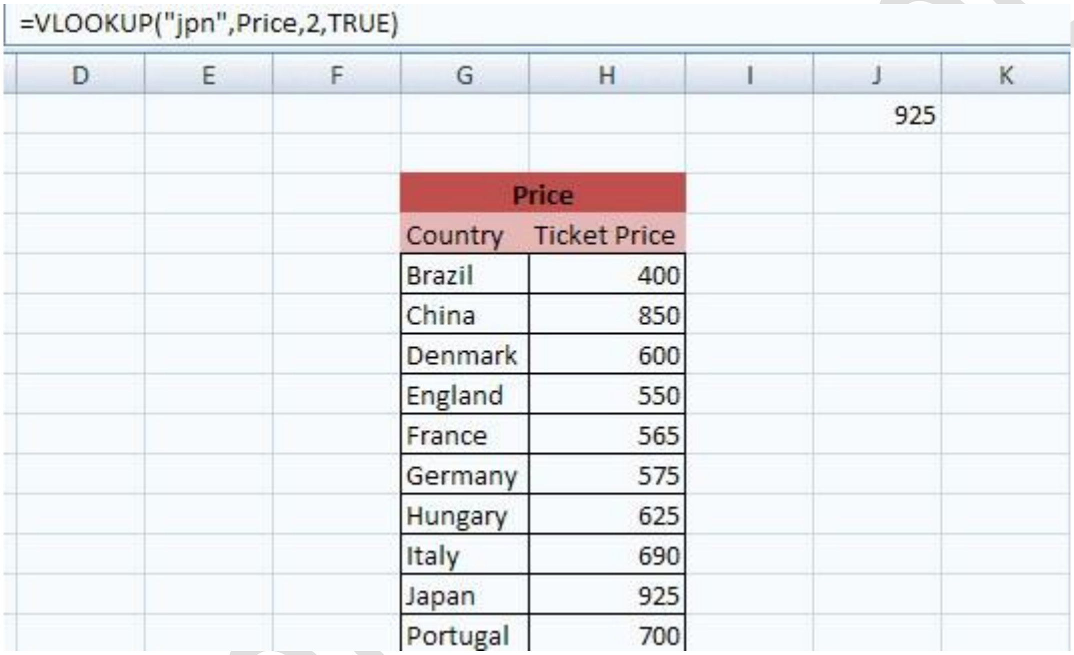
B	C	D	E	F	G
		925		550	

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.



```
=VLOOKUP("jpn",Price,2,TRUE)
```

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

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 as the final argument to your function (for an exact match), the lookup column does not have to be sorted.

Unit 3: Advanced List Management

In this unit, you will learn how to:

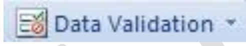
- Apply data validation settings to control data entry
- Identify and use database functions to perform calculations using multiple database fields

Validating Your Data

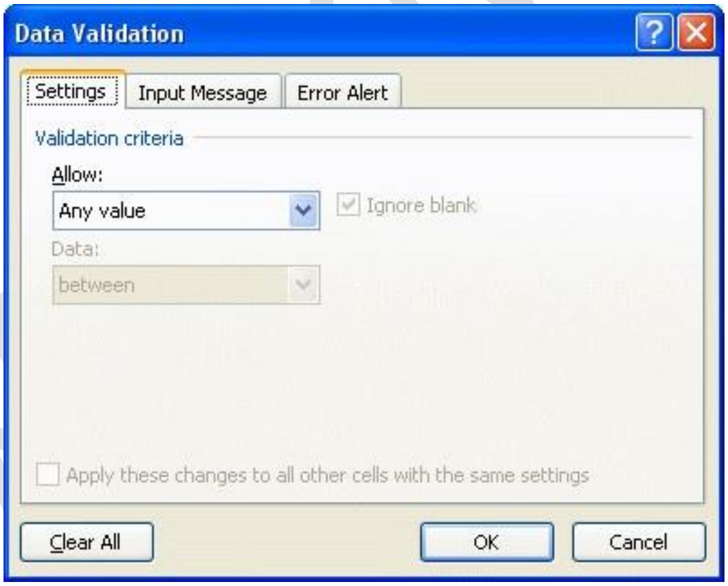
When you have a database set up in Excel, it is important to maintain its integrity. You can help ensure that only the right data goes into your database by setting up data validation rules.

Creating Data Validation Settings

To set up data validation for a cell, activate the cell of your choice and click the Data Validation button on the Data Ribbon.



This will display the Data Validation dialogue box.



Validation criteria

Allow: Any value

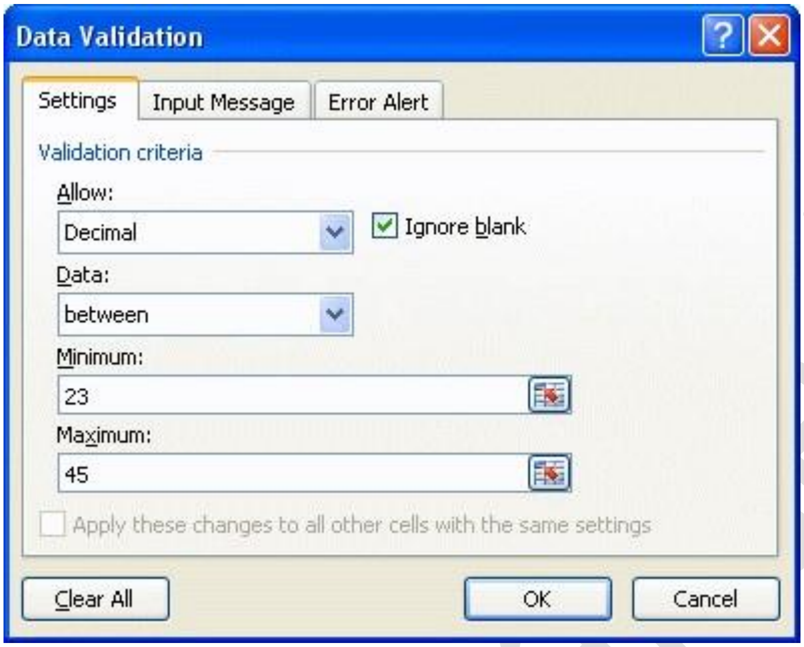
☑ Ignore blank

Data: between

□ Apply these changes to all other cells with the same settings

Under the Settings tab, you can use the Allow drop list to select the type of data value that will be permitted in the cell. Your choices are any value, whole number, decimal, list, date, time, text length, and custom. If you choose any option other than “any value,” you will be provided with further options to set criteria and or limits on the data values.

The following image shows an example where Decimal has been chosen as the allowed data type.



Allow: Decimal

☑ Ignore blank

Data: between

Minimum: 23

Maximum: 45

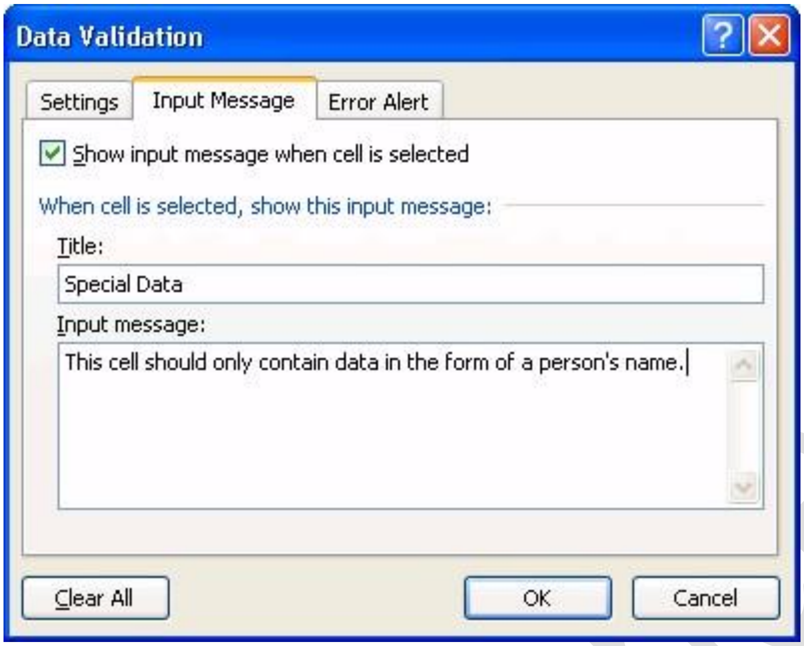
□ Apply these changes to all other cells with the same settings

Notice the other drop lists with options to set conditions and parameter values. You can set such conditions as: between, not between, equal to, not equal to, greater than, less than, greater than or equal to, and less than or equal to. With whatever condition you choose, you can specify the associated numerical parameters. (In this case only decimal values between 23 and 45 will be allowed in the cell.)

If there are other cells with the same validation settings, you can fill the bottom check box to apply the validation changes you have made to all of the cells.

Creating Entry and Error Messages

Under the Input Message tab, you will find options to create a message that will appear when the validated cell is selected.



☑ Show input message when cell is selected

When cell is selected, show this input message:

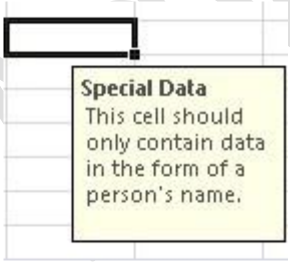
Title:

Special Data

Input message:

This cell should only contain data in the form of a person's name.

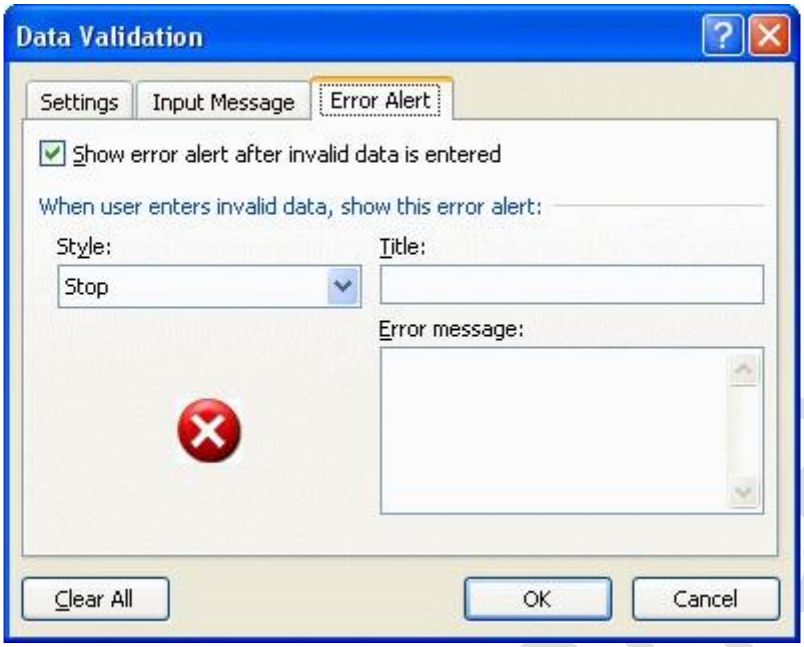
Here you can enter a title and a brief message with instructions or information about the type of data that should go in this cell. If you place a check in the checkbox at the top, the message will appear when the validated cell is selected.



Special Data

This cell should only contain data in the form of a person's name.

Under the Error Alert tab, you will find options to specify an error message if the wrong type of data is entered into the cell.



☑ Show error alert after invalid data is entered

Style: Stop

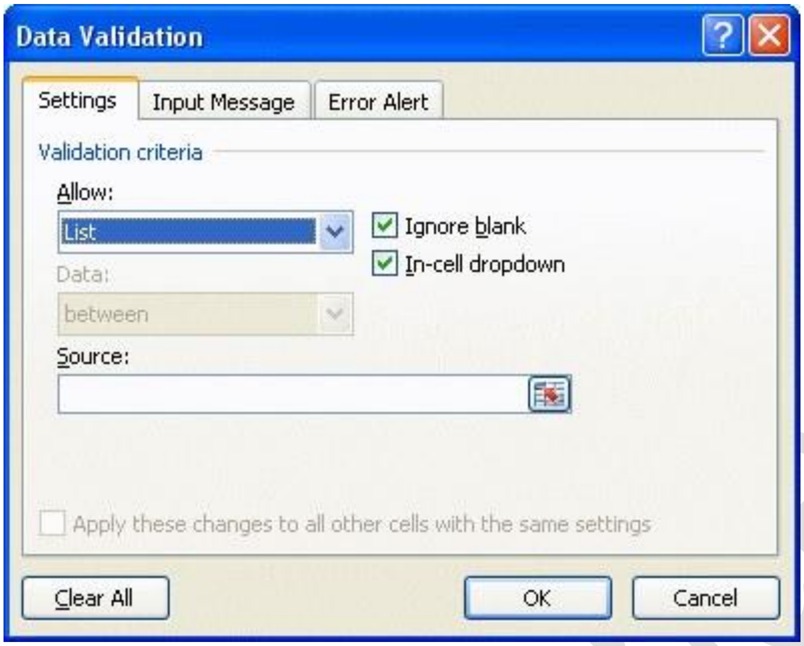
You can use these options to enter a title and message for a user that enters invalid data. You can also use the Style option list to specify what type of alert will appear.

A stop alert will not permit you to enter invalid data.

A warning alert can be used to warn of invalid data entry, but permit it if the user is sure.

An information alert is the most gentle, and can provide information to the user before he or she commits to entering the data.

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.



Data Validation

Settings | Input Message | Error Alert

Validation criteria

Allow: List

☑ Ignore blank

☑ In-cell dropdown

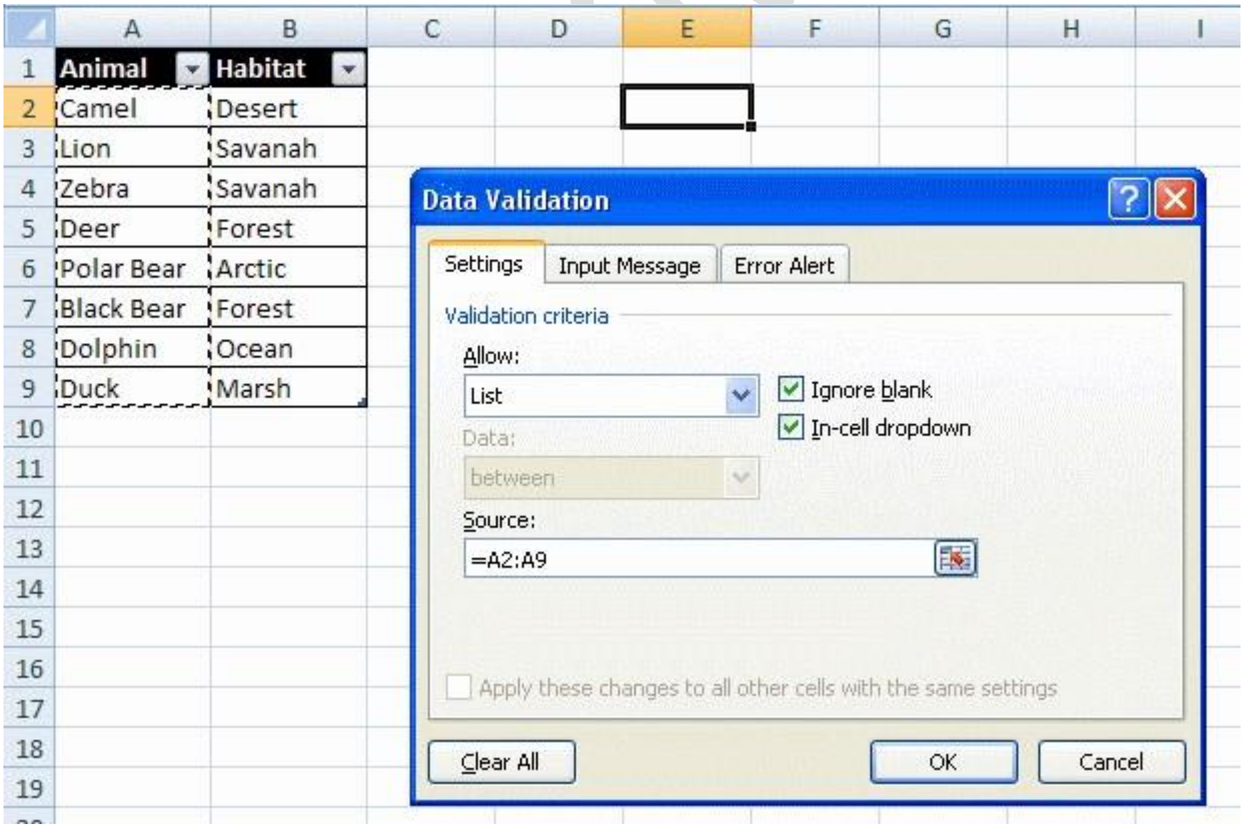
Data: between

Source:

□ Apply these changes to all other cells with the same settings

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.



	A	B
1	Animal	Habitat
2	Camel	Desert
3	Lion	Savanah
4	Zebra	Savanah
5	Deer	Forest
6	Polar Bear	Arctic
7	Black Bear	Forest
8	Dolphin	Ocean
9	Duck	Marsh

Data Validation

Settings | Input Message | Error Alert

Validation criteria

Allow: List

☑ Ignore blank

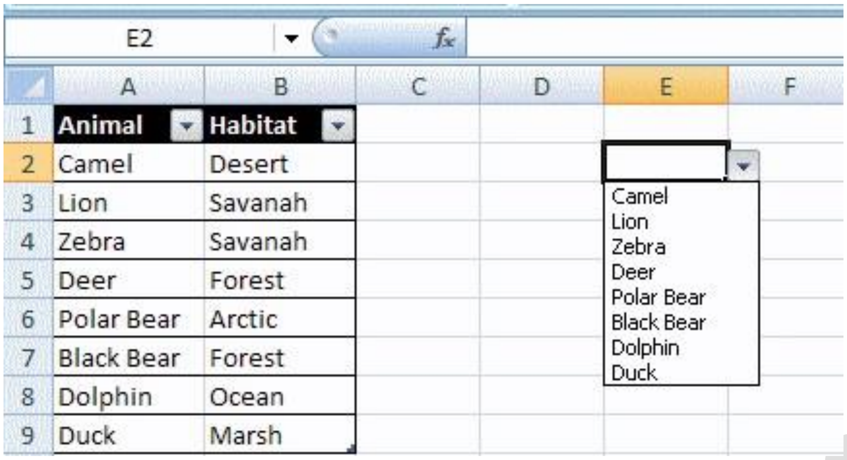
☑ In-cell dropdown

Data: between

Source: =A2:A9

□ Apply these changes to all other cells with the same settings

In this image, cell E2 is being set up so that items in cells A2 to A9 (surrounded by a dashed border) will be available as choices for data entry.

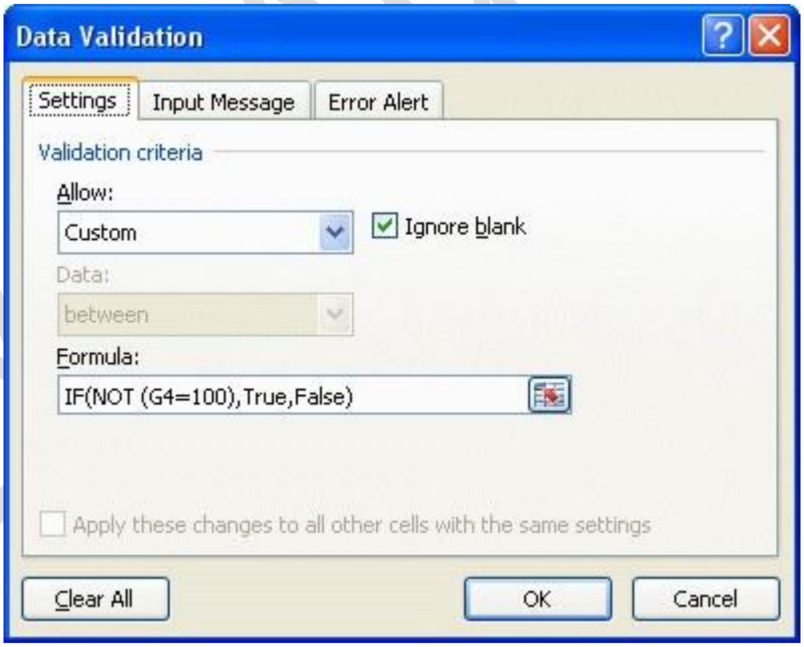


	Animal	Habitat
2	Camel	Desert
3	Lion	Savanah
4	Zebra	Savanah
5	Deer	Forest
6	Polar Bear	Arctic
7	Black Bear	Forest
8	Dolphin	Ocean
9	Duck	Marsh

Now, a user can choose from a validation drop list that is available for cell E2.

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.



Allow: Custom

☑ Ignore blank

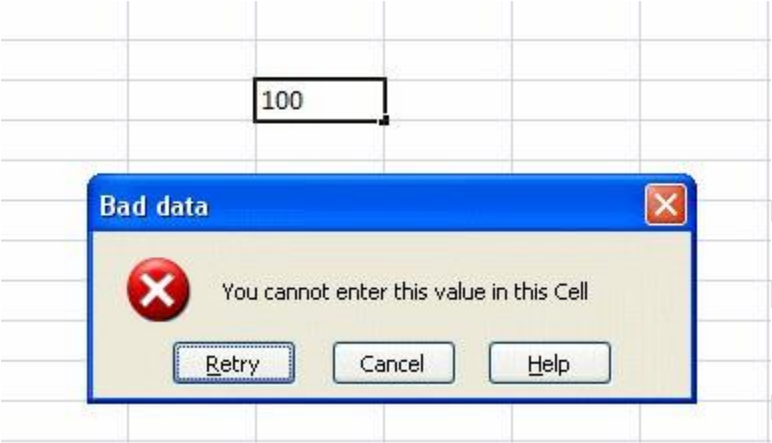
Data: between

Formula: IF(NOT (G4=100),True,False)

□ Apply these changes to all other cells with the same settings

Here the custom type has been chosen from the Allow drop list and a formula containing an IF function has been entered. The formula reads: IF(Not(G4=100),TRUE,FALSE).

This means that if the value entered into cell G4 (the cell that is being validated) is not 100, the function will return TRUE. If it is 100 the function will return FALSE. If the validation formula returns true, the data is permitted. If FALSE, the data is invalid. Basically, this formula allows any number other than 100 to be entered.



You cannot enter this value in this Cell

This image shows what happens if number 100 is entered into the cell. Remember, the type of alert and the message content can be specified when you are setting up the validation rules.

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 Return the largest and smallest values respectively from records that meet the DMIN specified conditions.

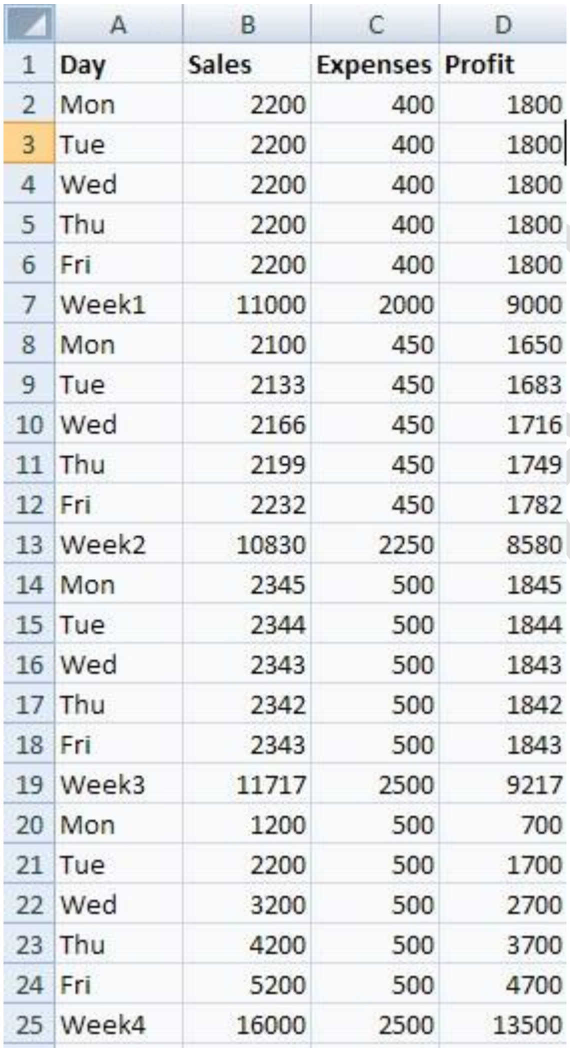
DPRODUCT Multiplies values in a field according to specific conditions

DGET Returns a single record value from a record that meets the specified conditions.

All of the database functions use the same argument format:

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

Take the following Excel table as an example.



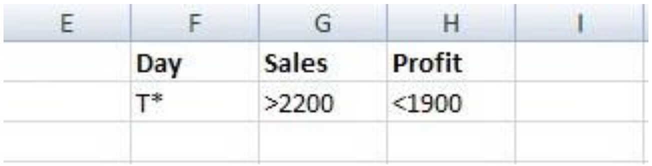
	A	B	C	D
1	Day	Sales	Expenses	Profit
2	Mon	2200	400	1800
3	Tue	2200	400	1800
4	Wed	2200	400	1800
5	Thu	2200	400	1800
6	Fri	2200	400	1800
7	Week1	11000	2000	9000
8	Mon	2100	450	1650
9	Tue	2133	450	1683
10	Wed	2166	450	1716
11	Thu	2199	450	1749
12	Fri	2232	450	1782
13	Week2	10830	2250	8580
14	Mon	2345	500	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.



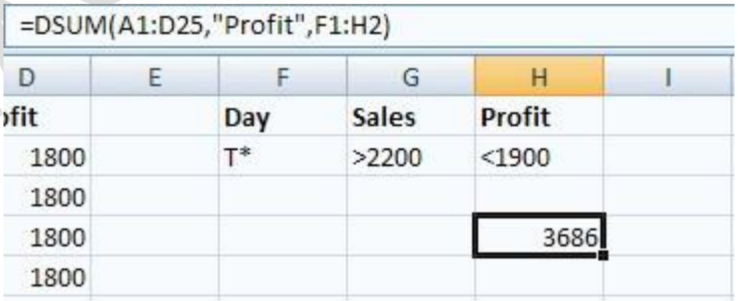
E	F	G	H	I
	Day	Sales	Profit	
	T*	>2200	<1900	

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.



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

D	E	F	G	H	I
›fit		Day	Sales	Profit	
1800		T*	>2200	<1900	
1800					
1800				3686	
1800					

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, and the criteria range is F1:H2.

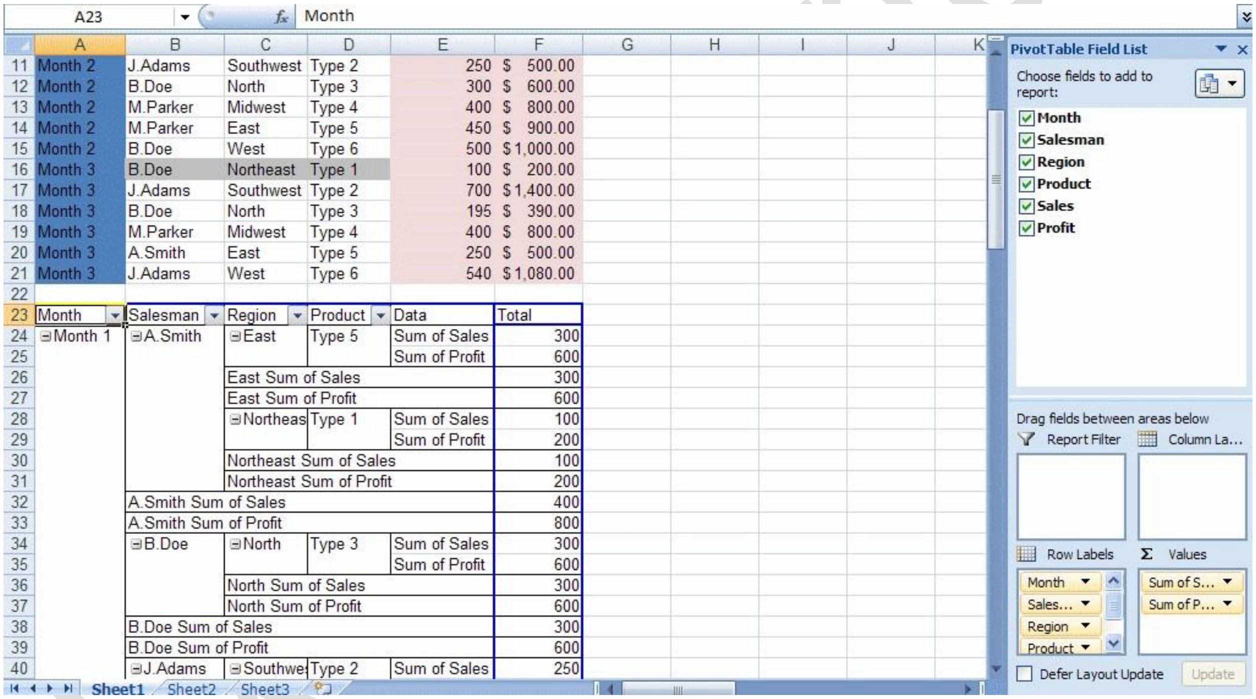
If you enter a database function directly into the formula bar, remember to keep your cell ranges accurate, and put the field that will be summed, averaged, or otherwise operated on, in quotation marks.

Unit 4: PivotTables and PivotCharts

In this unit, you will learn how to:

Create a PivotTable from a list of data
Modify a PivotTable to change data display
Create a PivotChart
Use external data to create a PivotTable

A major function of any spreadsheet program is to help you derive meaning from your data. An Excel PivotTable is a great tool for getting perspective on, and analysing relationships between, the columns and rows of your worksheet.



A23 | Month

	A	B	C	D	E	F
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
18	Month 3	B.Doe	North	Type 3	195	$ 390.00
19	Month 3	M.Parker	Midwest	Type 4	400	$ 800.00
20	Month 3	A.Smith	East	Type 5	250	$ 500.00
21	Month 3	J.Adams	West	Type 6	540	$ 1,080.00
22						
23	Month	Salesman	Region	Product	Data	Total
24	⊟Month 1	⊟A.Smith	⊟East	Type 5	Sum of Sales	300
25					Sum of Profit	600
26			East Sum of Sales			300
27			East Sum of Profit			600
28			⊟Northeas	Type 1	Sum of Sales	100
29					Sum of Profit	200
30			Northeast Sum of Sales			100
31			Northeast Sum of Profit			200
32		A.Smith Sum of Sales				400
33		A.Smith Sum of Profit				800
34		⊟B.Doe	⊟North	Type 3	Sum of Sales	300
35					Sum of Profit	600
36			North Sum of Sales			300
37			North Sum of Profit			600
38		B.Doe Sum of Sales				300
39		B.Doe Sum of Profit				600
40		⊟J.Adams	⊟Southwe	Type 2	Sum of Sales	250

Sheet1 Sheet2 Sheet3

PivotTable Field List

Choose fields to add to report:

☑ Month
☑ Salesman
☑ Region
☑ Product
☑ Sales
☑ Profit

Drag fields between areas below

Report Filter | Column La...

Row Labels: Month, Sales..., Region, Product

Values: Sum of S..., Sum of P...

□ Defer Layout Update | Update

What is a PivotTable?

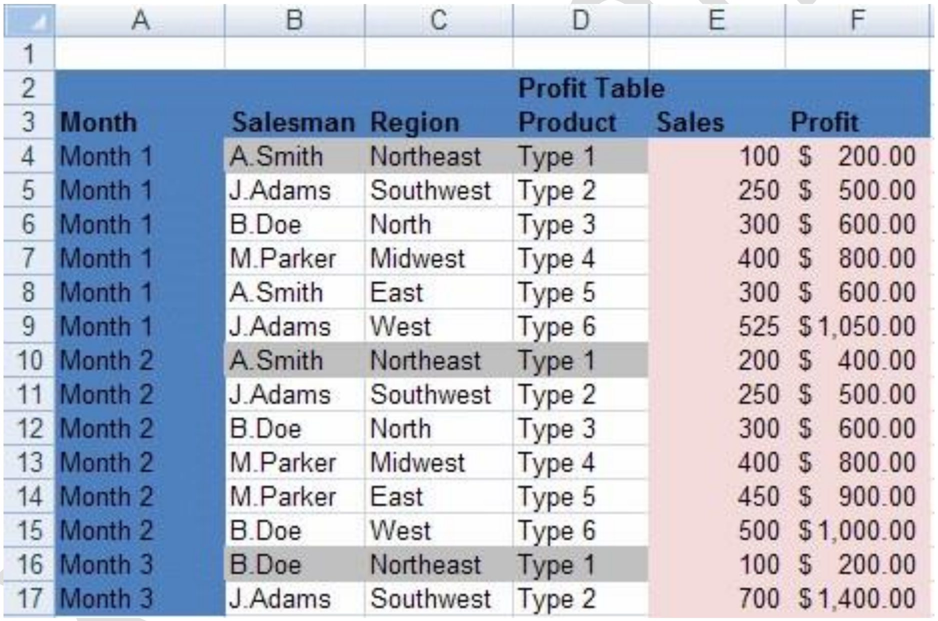
A PivotTable is a powerful tool for exploring and analysing information. A PivotTable helps you organise and manipulate the raw data in your spreadsheet, giving you insight into patterns or relationships that might not be obvious at first glance. PivotTables also give you the power to view your data 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.

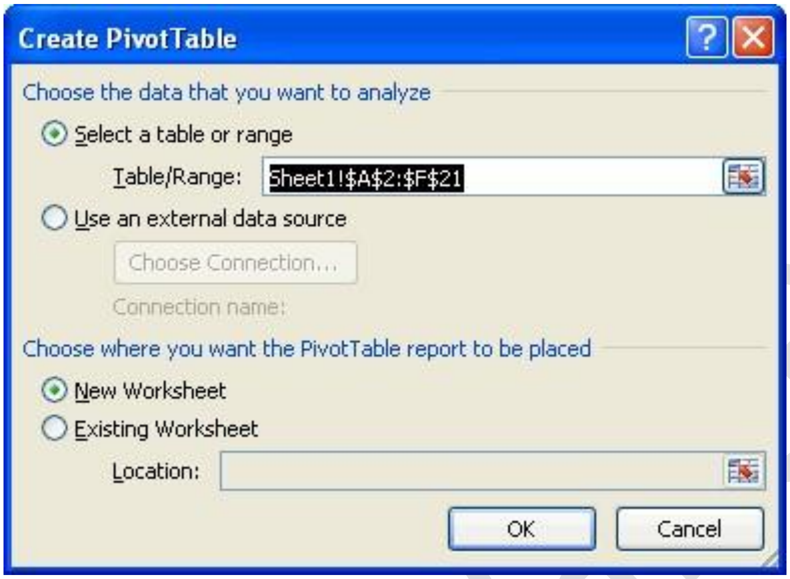


	A	B	C	D	E	F
1						
2				Profit Table		
3	Month	Salesman	Region	Product	Sales	Profit
4	Month 1	A.Smith	Northeast	Type 1	100	$ 200.00
5	Month 1	J.Adams	Southwest	Type 2	250	$ 500.00
6	Month 1	B.Doe	North	Type 3	300	$ 600.00
7	Month 1	M.Parker	Midwest	Type 4	400	$ 800.00
8	Month 1	A.Smith	East	Type 5	300	$ 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 2013 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.



Create PivotTable

Choose the data that you want to analyze

● Select a table or range

Table/Range: Sheet1!$A$2:$F$21

○ Use an external data source

Choose Connection...

Connection name:

Choose where you want the PivotTable report to be placed

● New Worksheet

○ Existing Worksheet

Location:

OK Cancel

Selecting PivotTable Data

When the Create PivotTable dialogue box appears, you should notice that you are allowed to select data from an Excel table or range or from an external data source. If you choose the table or range option, the PivotTable will be based on the Excel table or range you select.

You can select a table or range by dragging with your mouse before clicking the PivotTable button (before you invoke the dialogue), or by specifying a range with the keyboard or mouse after you invoke the dialogue. (When you select your data source, make sure that the column headings are included.)

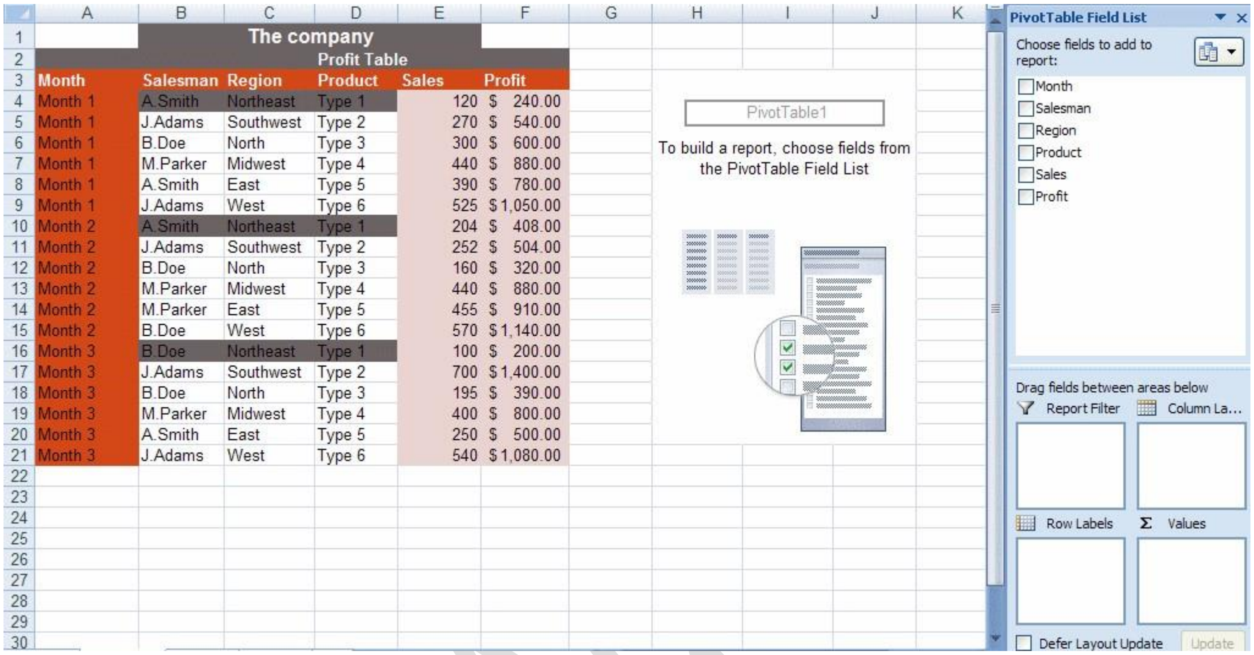
If you choose the External Data Source option, you can base your PivotTable on data outside your current workbook (such as another workbook or perhaps an external database). If you select the Use an External Data Source radio button, you will be able to display a drop list of existing connections in 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.

Selecting PivotTable Location

Once you select your data source, you can then choose to locate your PivotTable in an Existing Worksheet or a New Worksheet. If you choose to locate it in an existing worksheet, you can specify the location for the upper left corner of the PivotTable by entering it directly into the Location data field (as a cell reference) or by clicking the target cell with your mouse.

If you choose the new worksheet option, your PivotTable will be located in the upper left corner of a new worksheet that will be added to your workbook.

In summary, when using the dialogue box, make sure that the range of data that you want to use is visible in the Table/Range field. Use the radio buttons near the bottom of the dialogue to choose where in the workbook you want the PivotTable to be located. Once you are ready, click the OK button to create your PivotTable. When you create your table, a PivotTable field List will be placed in the same worksheet as your PivotTable.



	A	B	C	D	E	F
1		The company				
2				Profit Table		
3	Month	Salesman	Region	Product	Sales	Profit
4	Month 1	A.Smith	Northeast	Type 1	120	$ 240.00
5	Month 1	J.Adams	Southwest	Type 2	270	$ 540.00
6	Month 1	B.Doe	North	Type 3	300	$ 600.00
7	Month 1	M.Parker	Midwest	Type 4	440	$ 880.00
8	Month 1	A.Smith	East	Type 5	390	$ 780.00
9	Month 1	J.Adams	West	Type 6	525	$ 1,050.00
10	Month 2	A.Smith	Northeast	Type 1	204	$ 408.00
11	Month 2	J.Adams	Southwest	Type 2	252	$ 504.00
12	Month 2	B.Doe	North	Type 3	160	$ 320.00
13	Month 2	M.Parker	Midwest	Type 4	440	$ 880.00
14	Month 2	M.Parker	East	Type 5	455	$ 910.00
15	Month 2	B.Doe	West	Type 6	570	$ 1,140.00
16	Month 3	B.Doe	Northeast	Type 1	100	$ 200.00
17	Month 3	J.Adams	Southwest	Type 2	700	$ 1,400.00
18	Month 3	B.Doe	North	Type 3	195	$ 390.00
19	Month 3	M.Parker	Midwest	Type 4	400	$ 800.00
20	Month 3	A.Smith	East	Type 5	250	$ 500.00
21	Month 3	J.Adams	West	Type 6	540	$ 1,080.00

PivotTable1

To build a report, choose fields from the PivotTable Field List

PivotTable Field List

Choose fields to add to report:

□ Month
□ Salesman
□ Region
□ Product
□ Sales
□ Profit

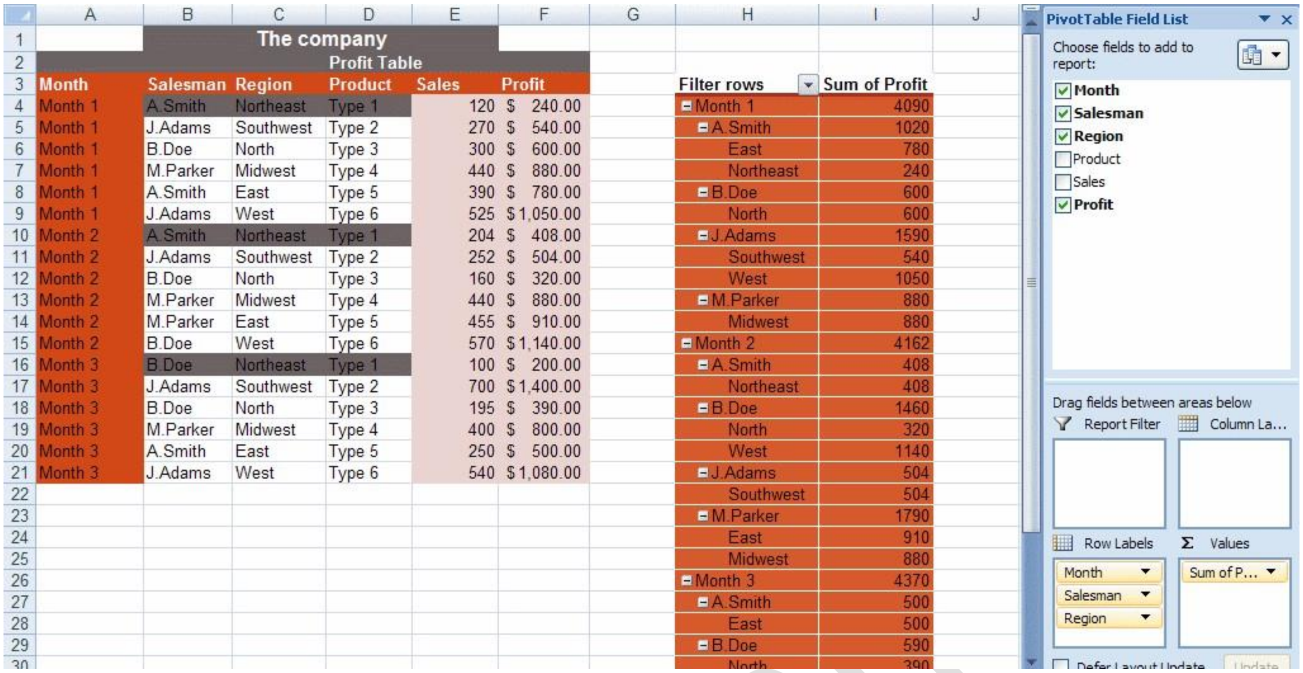
Drag fields between areas below

□ Defer Layout Update

Here you can see a new PivotTable area, and the corresponding PivotTable Field List, placed in the existing worksheet with the source data.

Adding Fields to the PivotTable

Once your PivotTable appears, you can add information to it by placing checks in the boxes next to the headings in the PivotTable field list. For this example, checks will be placed next to the Month field heading, the Salesman field heading, the Region field heading, and the Profit field Heading.



	A	B	C	D	E	F
1		The company				
2				Profit Table		
3	Month	Salesman	Region	Product	Sales	Profit
4	Month 1	A.Smith	Northeast	Type 1	120	$ 240.00
5	Month 1	J.Adams	Southwest	Type 2	270	$ 540.00
6	Month 1	B.Doe	North	Type 3	300	$ 600.00
7	Month 1	M.Parker	Midwest	Type 4	440	$ 880.00
8	Month 1	A.Smith	East	Type 5	390	$ 780.00
9	Month 1	J.Adams	West	Type 6	525	$ 1,050.00
10	Month 2	A.Smith	Northeast	Type 1	204	$ 408.00
11	Month 2	J.Adams	Southwest	Type 2	252	$ 504.00
12	Month 2	B.Doe	North	Type 3	160	$ 320.00
13	Month 2	M.Parker	Midwest	Type 4	440	$ 880.00
14	Month 2	M.Parker	East	Type 5	455	$ 910.00
15	Month 2	B.Doe	West	Type 6	570	$ 1,140.00
16	Month 3	B.Doe	Northeast	Type 1	100	$ 200.00
17	Month 3	J.Adams	Southwest	Type 2	700	$ 1,400.00
18	Month 3	B.Doe	North	Type 3	195	$ 390.00
19	Month 3	M.Parker	Midwest	Type 4	400	$ 800.00
20	Month 3	A.Smith	East	Type 5	250	$ 500.00
21	Month 3	J.Adams	West	Type 6	540	$ 1,080.00

Filter rows	Sum of Profit
⊟ Month 1	4090
⊟ A.Smith	1020
East	780
Northeast	240
⊟ B.Doe	600
North	600
⊟ J.Adams	1590
Southwest	540
West	1050
⊟ M.Parker	880
Midwest	880
⊟ Month 2	4162
⊟ A.Smith	408
Northeast	408
⊟ B.Doe	1460
North	320
West	1140
⊟ J.Adams	504
Southwest	504
⊟ M.Parker	1790
East	910
Midwest	880
⊟ Month 3	4370
⊟ A.Smith	500
East	500
⊟ B.Doe	590
North	390

PivotTable Field List

Choose fields to add to report:

☑ Month
☑ Salesman
☑ Region
☐ Product
☐ Sales
☑ Profit

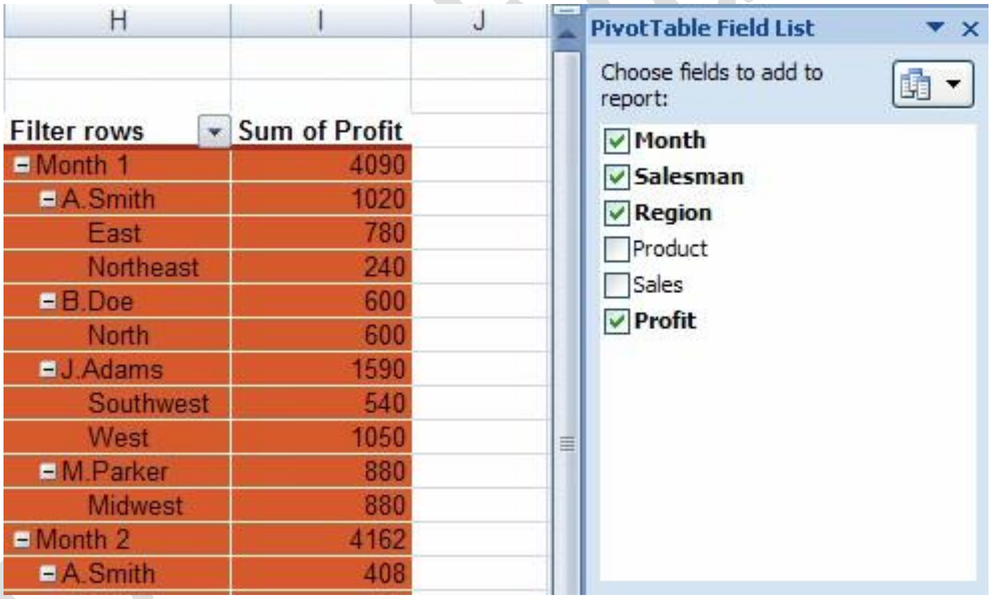
Drag fields between areas below

Report Filter | Column La...

Row Labels: Month, Salesman, Region

Values: Sum of P...

The PivotTable will now be populated with data.



Filter rows	Sum of Profit
⊟ Month 1	4090
⊟ A.Smith	1020
East	780
Northeast	240
⊟ B.Doe	600
North	600
⊟ J.Adams	1590
Southwest	540
West	1050
⊟ M.Parker	880
Midwest	880
⊟ Month 2	4162
⊟ A.Smith	408

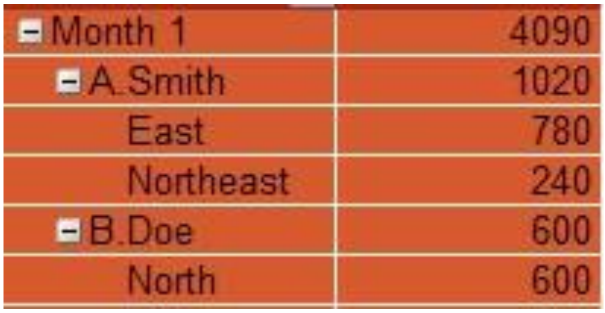
PivotTable Field List

Choose fields to add to report:

☑ Month
☑ Salesman
☑ Region
☐ Product
☐ Sales
☑ Profit

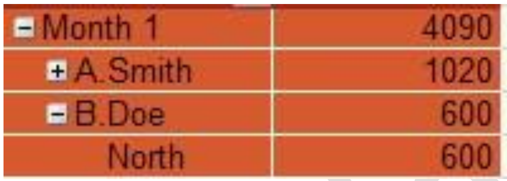
As you can see in this image, the profit has been organised by Month with a total profit for the month at the top of the Sum of Profit column heading. It has also been organised by Salesman, with a total profit for each Salesman being shown in the Sum of Profit column. Because Region has been checked in the PivotTable field list, you can also see a profit breakdown by region for each salesman.

The following “close up” view of the table tells us that the total profit for Month 1 is 4090. The Salesman A. Smith generated a total of 1020 in profit with 780 from the East region, and 240 from the Northeast region.



⊟ Month 1	4090
⊟ A.Smith	1020
East	780
Northeast	240
⊟ B.Doe	600
North	600

The salesman B. Doe generated a total of 600 in profit from the North region. If you click on the minus (-) sign preceding a salesman's name, the data for that specific salesman will be collapsed, and you will only see the total profit for that person as shown in the following example.



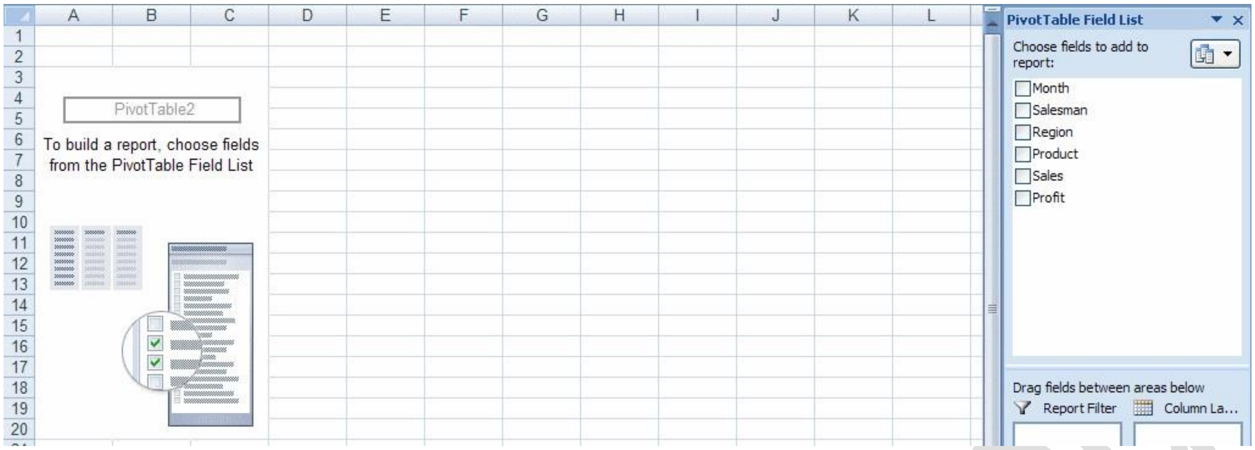
⊟ Month 1	4090
⊞ A.Smith	1020
⊟ B.Doe	600
North	600

If you click on the (+) sign in front of a name, the data will be expanded again. This holds true for the (-) sign in front of the Month headings as well.

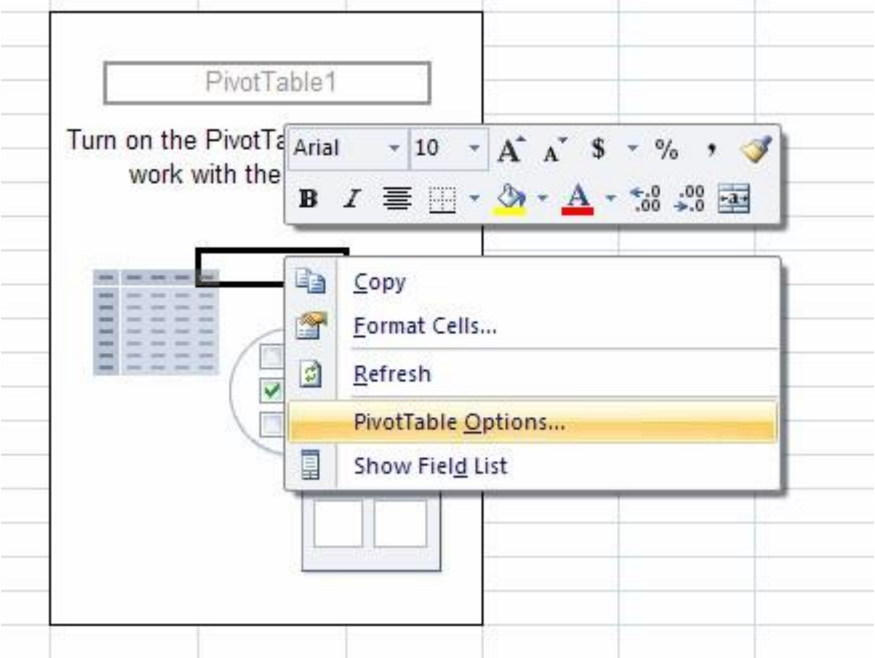
Creating a PivotTable Frame (Classic PivotTable Layout)

You can also create a PivotTable by creating an empty table frame (also known as a classic layout) and adding and arranging data in the table by dropping and dragging column headings onto the frame.

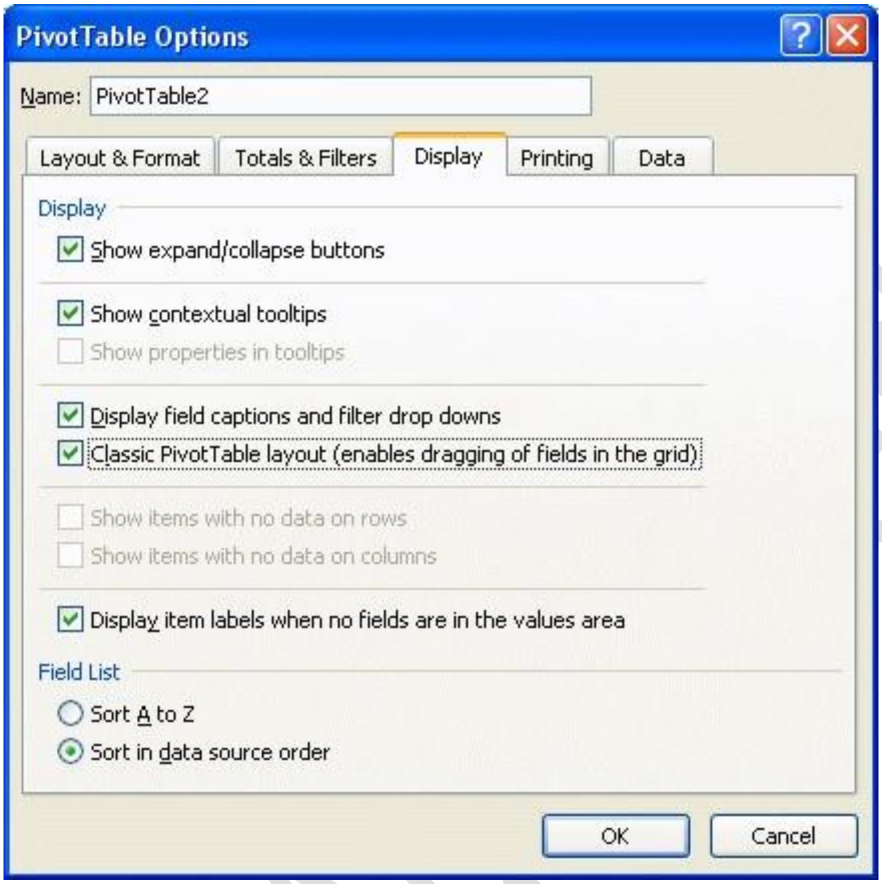
To create a PivotTable frame, click the PivotTable button on the Insert Ribbon to display the PivotTable dialogue box. From this point, you essentially follow the same process as described before for creating an Excel 2013 PivotTable. You select a range of data, choose a location for the table, and click the OK button when you are ready.



This time, before adding data to the PivotTable table, right click on it and choose Table Options from the pop up menu.



This will display the PivotTable options dialogue box.



Name: PivotTable2

Display

☑ Show expand/collapse buttons

☑ Show contextual tooltips

□ Show properties in tooltips

☑ Display field captions and filter drop downs

☑ Classic PivotTable layout (enables dragging of fields in the grid)

□ Show items with no data on rows

□ Show items with no data on columns

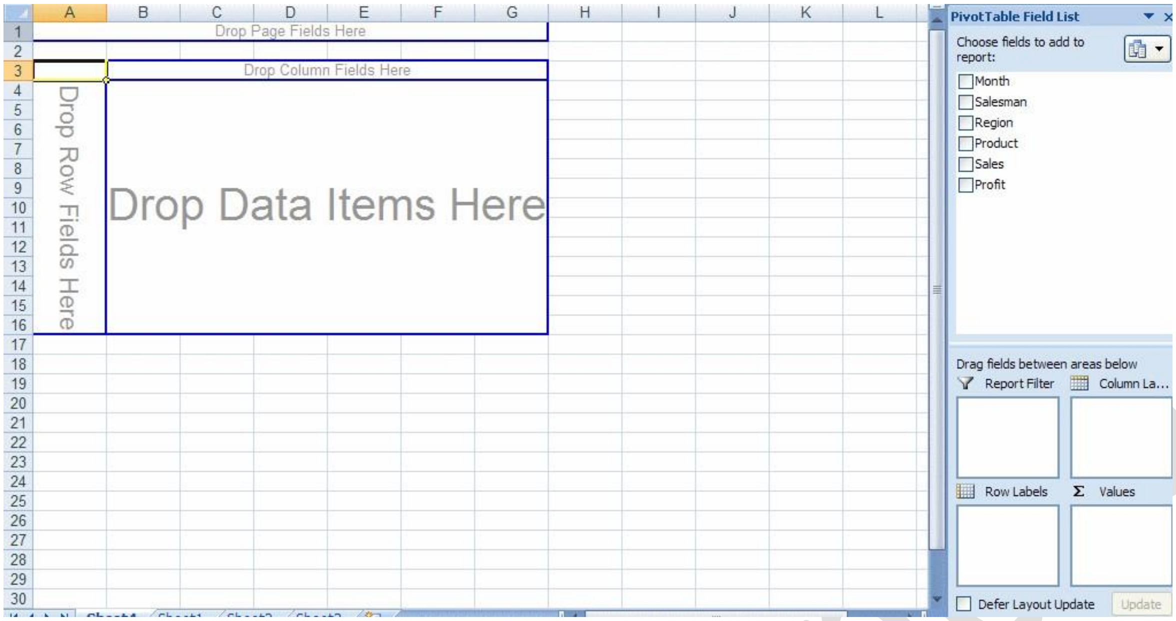
☑ Display item labels when no fields are in the values area

Field List

○ Sort A to Z

● Sort in data source order

In the options box, choose the Display tab and then put a checkmark in the Classic PivotTable Layout box. This will change the Excel 2013 PivotTable into a classic PivotTable frame. (If you are already working with a classic PivotTable frame, you can clear this checkbox to convert it to an Excel 2013 style PivotTable.)



Here you can see an empty PivotTable frame enhanced with blue borders. Beside it to the right, you can see the PivotTable Field List. You can easily add any of the data available in the PivotTable Field List to your PivotTable by dragging it to the table with your mouse.

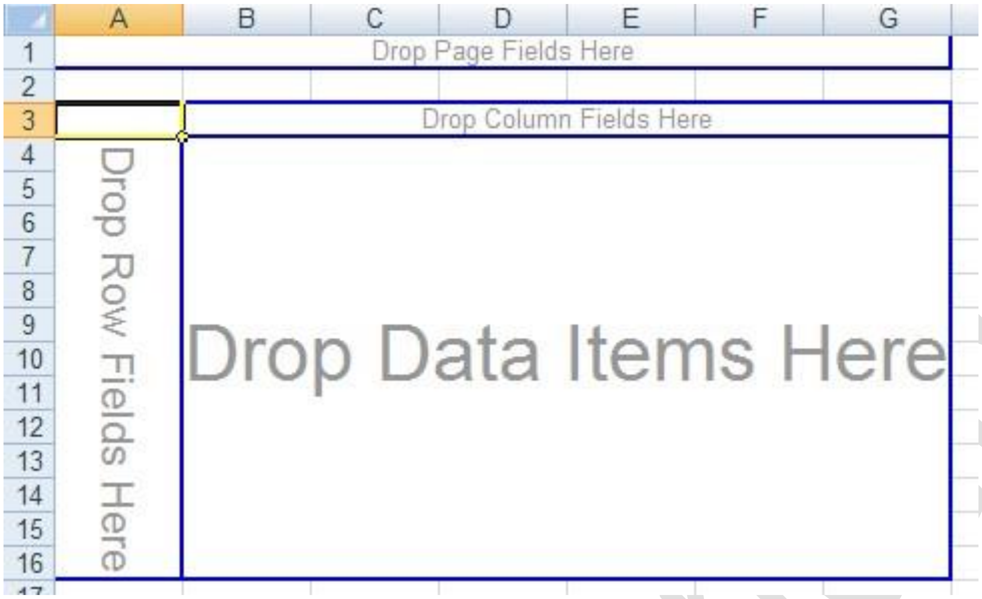
Modifying a PivotTable using Classic PivotTable Layout

PivotTables are designed to make working with your data flexible, so that the information can be arranged in numerous ways, depending on how you wish the data to be presented or analysed

PivotTable areas

To make the best use of your PivotTable, you should understand what the various areas in a PivotTable are for.

You can see by examining an empty PivotTable frame that a PivotTable is broken into four main areas: Page, Data, Row, and Column.

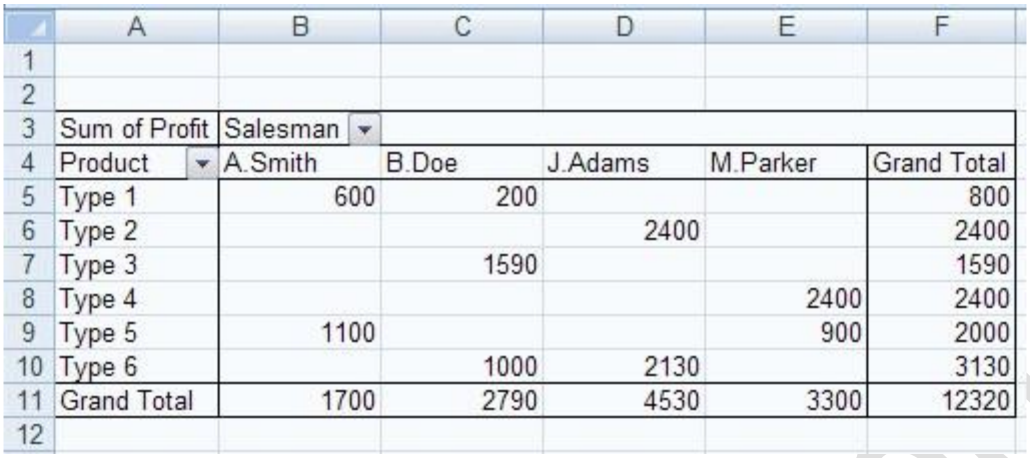


The Page Field area is ideally suited to column headings (fields) that are used to identify periodic or organizational groupings of the data in your other columns. For example, you may have column headings for sales, profit, and expenses, which are grouped according to the specific month or year over which the sales, profits, and expenses have occurred. In this case, the month or the year column heading would be placed in the page area of the PivotTable. This would allow you to populate the table with the data corresponding to whatever page field (year or month) you select. You can select a given page (year or month) and the table will display the data for that year or month.

The Data area provides the underlying context for the rest of the PivotTable. The column heading you choose for the data area of the table normally has numeric values associated with it (like units sold, profit, or expenses). This is the kind of data that can be measured and totaled to reveal trends or indicate relationships between non numeric data.

The Row and Column fields are used to categorize the data you want to examine. When you choose column headings to be placed in the row and column fields, you can see how your choices relate to each other in the context of the Data being examined.

As an example, suppose you select a heading like Product Type for the row area, Salesman for the column area, and Profit for the data area. The resulting PivotTable will show the profit for different product types across sales people.



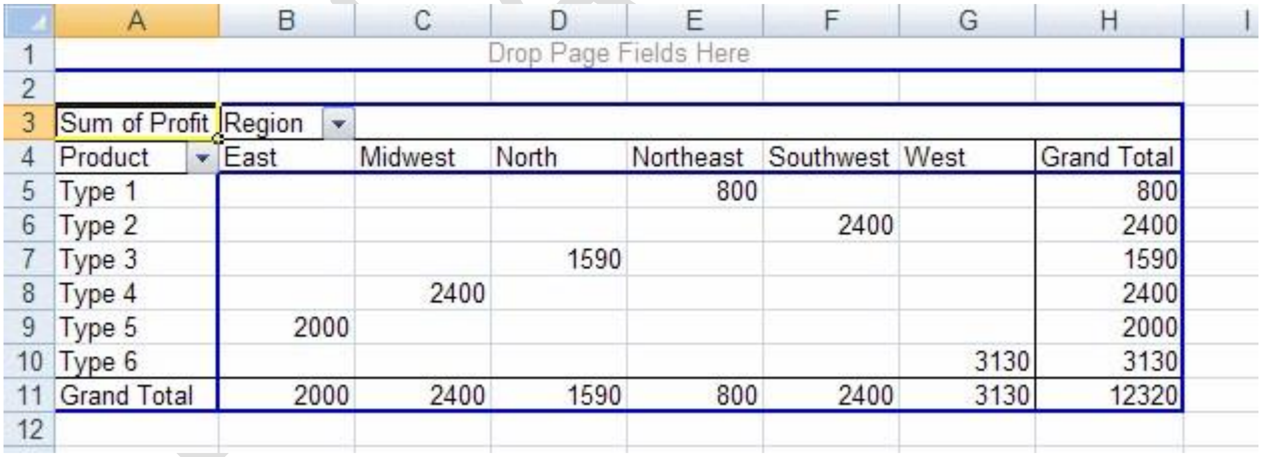
	A	B	C	D	E	F
1						
2						
3	Sum of Profit	Salesman				
4	Product	A.Smith	B.Doe	J.Adams	M.Parker	Grand Total
5	Type 1	600	200			800
6	Type 2			2400		2400
7	Type 3		1590			1590
8	Type 4				2400	2400
9	Type 5	1100			900	2000
10	Type 6		1000	2130		3130
11	Grand Total	1700	2790	4530	3300	12320
12						

Now that you have an idea of how the parts of a PivotTable relate, it is easy to specify data for the table. You just use your mouse to drag and drop items from the PivotTable field list to the appropriate areas of the table.

Once again, you should specify a column heading (field) that represents numeric values for the data area. For the row and column areas of the table, drag and drop items that you want to analyze with respect to the chosen item for the data area.

Rearranging PivotTable Data

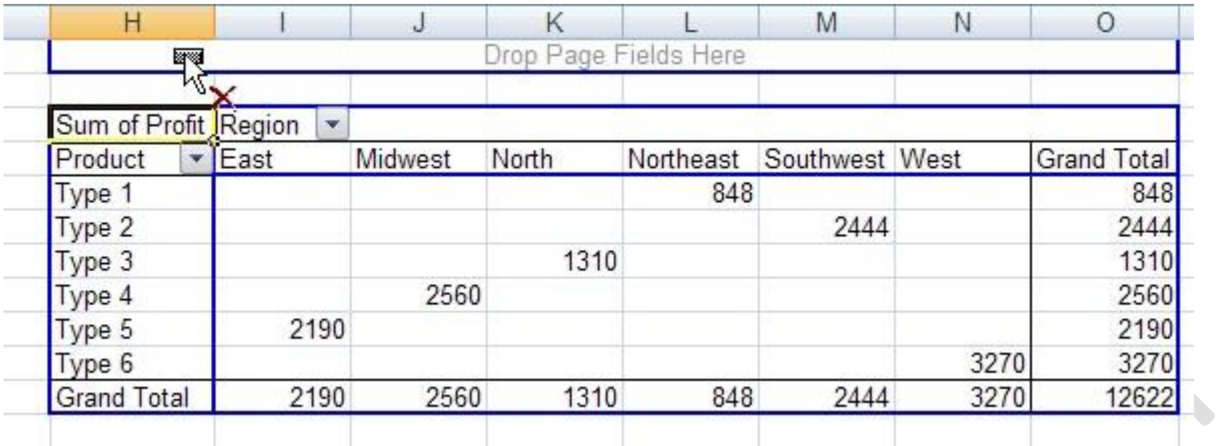
Once a PivotTable has been created, it is easy to rearrange the data if necessary. Note that you must be using the Classic Layout (as described in the last lesson) to rearrange data.



	A	B	C	D	E	F	G	H
1	Drop Page Fields Here							
2								
3	Sum of Profit	Region						
4	Product	East	Midwest	North	Northeast	Southwest	West	Grand Total
5	Type 1				800			800
6	Type 2					2400		2400
7	Type 3			1590				1590
8	Type 4		2400					2400
9	Type 5	2000						2000
10	Type 6						3130	3130
11	Grand Total	2000	2400	1590	800	2400	3130	12320
12								

Here we have a table showing profit by product type across regions.

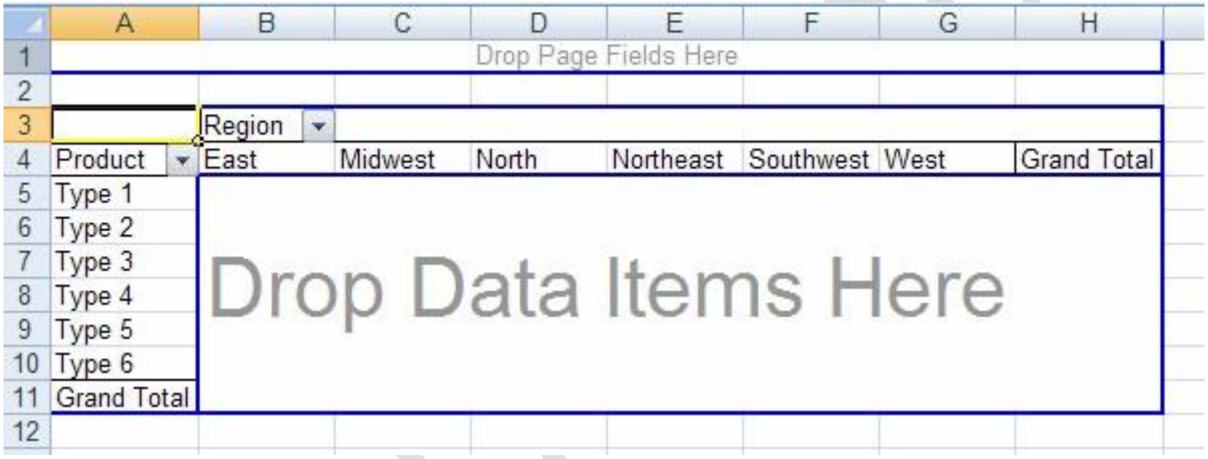
If we wanted to show sales figures instead of profit figures, you can remove the profit information from the data area by dragging the Sum of Profit heading in the upper left corner of the table out of the table area to drop it anywhere outside the blue table borders.



Drop Page Fields Here

Sum of Profit	Region						
Product	East	Midwest	North	Northeast	Southwest	West	Grand Total
Type 1				848			848
Type 2					2444		2444
Type 3			1310				1310
Type 4		2560					2560
Type 5	2190						2190
Type 6						3270	3270
Grand Total	2190	2560	1310	848	2444	3270	12622

This will empty the data area.

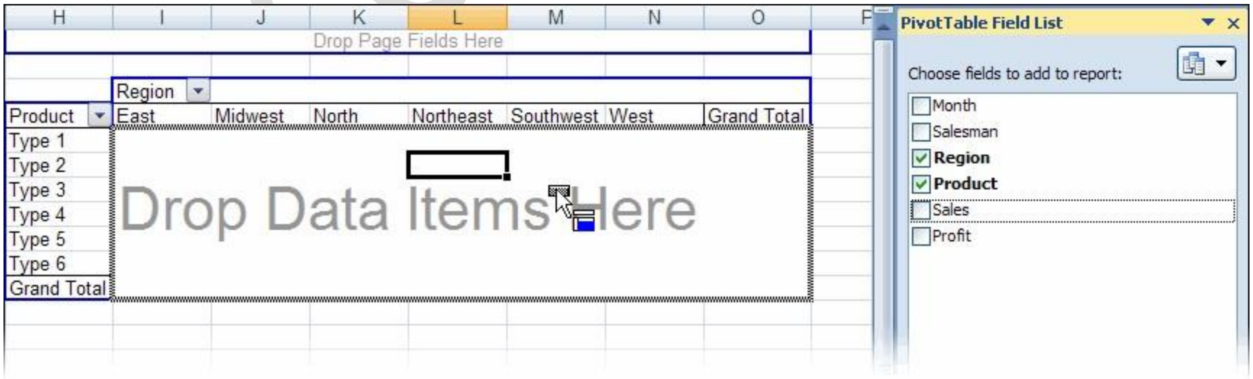


Drop Page Fields Here

	Region						
Product	East	Midwest	North	Northeast	Southwest	West	Grand Total
Type 1							
Type 2							
Type 3							
Type 4							
Type 5							
Type 6							
Grand Total							

Drop Data Items Here

Once the data area has been cleared, it is just a matter of dragging and dropping the Sales field from the PivotTable Field List to the Drop Data Items Here area of the table.



Drop Page Fields Here

	Region						
Product	East	Midwest	North	Northeast	Southwest	West	Grand Total
Type 1							
Type 2							
Type 3							
Type 4							
Type 5							
Type 6							
Grand Total							

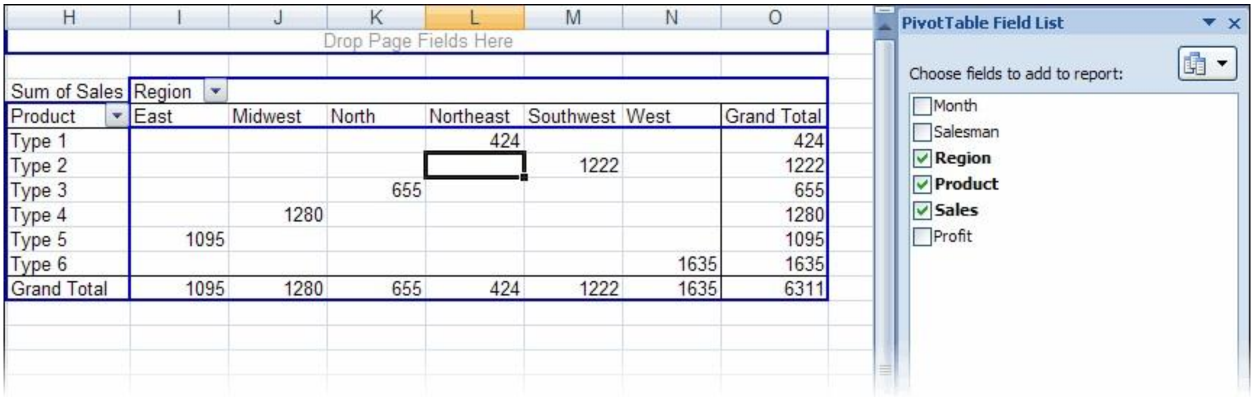
Drop Data Items Here

PivotTable Field List

Choose fields to add to report:

- ☐ Month
- ☐ Salesman
- ☑ **Region**
- ☑ **Product**
- ☐ Sales
- ☐ Profit

In this image, the Sales heading is being dragged from the field list to the Data area of the table. Once the move is complete, this is what the table will look like:



Drop Page Fields Here

H	I	J	K	L	M	N	O
Sum of Sales	Region ▾						
Product ▾	East	Midwest	North	Northeast	Southwest	West	Grand Total
Type 1				424			424
Type 2					1222		1222
Type 3			655				655
Type 4		1280					1280
Type 5	1095						1095
Type 6						1635	1635
Grand Total	1095	1280	655	424	1222	1635	6311

PivotTable Field List

Choose fields to add to report:

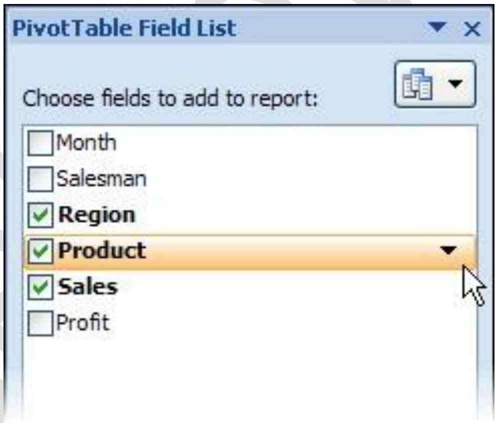
□ Month
□ Salesman
☑ **Region**
☑ **Product**
☑ **Sales**
□ Profit

Notice that Sum of Sales is in the upper left corner and the Sales heading has a check next to it in the PivotTable field list.

You can rearrange any of the categories of data in your table by following this process of dragging headings out of the table and then replacing them with headings from the PivotTable field list.

Hiding and Showing Field Data

You will also notice that the row, column, and page areas of the PivotTable field list have drop lists associated with them, indicated by small downward pointing triangles.

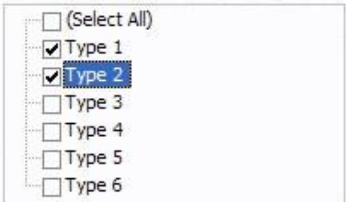


PivotTable Field List

Choose fields to add to report:

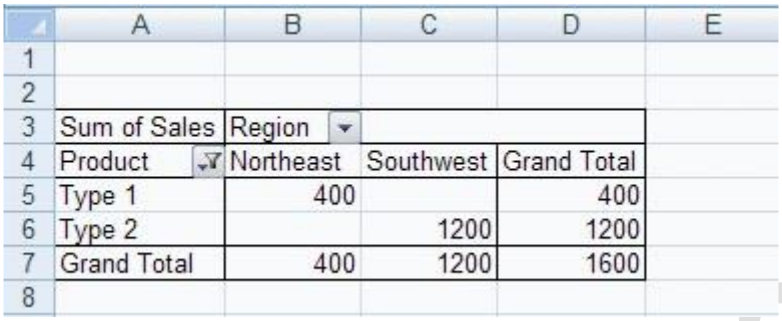
□ Month
□ Salesman
☑ **Region**
☑ **Product** ▾
☑ **Sales**
□ Profit

You can use these drop lists to rearrange your data according to specific elements in a given category. The product drop list, for example, will let you see data for only Type 1, or Type 2, or for a specific combination of types, depending on how you configure the check boxes.



□ (Select All)
☑ Type 1
☑ Type 2
□ Type 3
□ Type 4
□ Type 5
□ Type 6

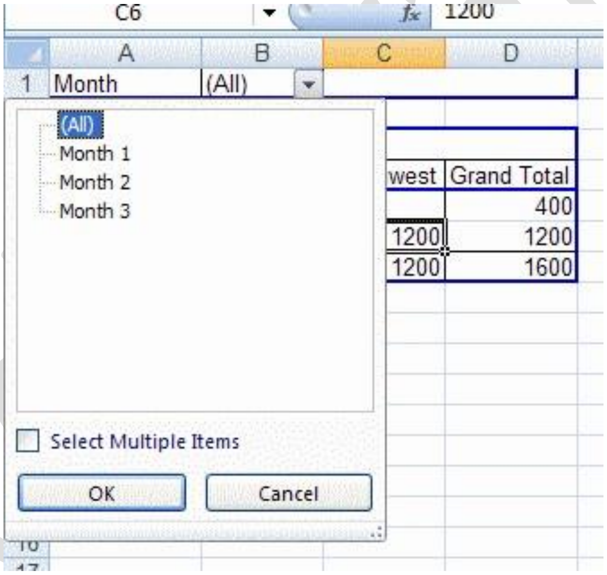
This is the drop list for the Product heading. (Any of the headings in the PivotTable that have drop lists can be configured in the same way.)



	A	B	C	D	E
1					
2					
3	Sum of Sales	Region			
4	Product	Northeast	Southwest	Grand Total	
5	Type 1	400		400	
6	Type 2		1200	1200	
7	Grand Total	400	1200	1600	
8					

This table only shows data for the products of Type 1 and Type 2, as specified by in the Product drop list. (By default, the drop lists are configured to show all.)

In the following image, the Month field has been dragged to the Page area of the table. The drop list for the Month field can be used to show the data for a specific month, or all of the months.



C6 fx 1200

	A	B	C	D
1	Month	(All)		

(All)
Month 1
Month 2
Month 3

☐ Select Multiple Items

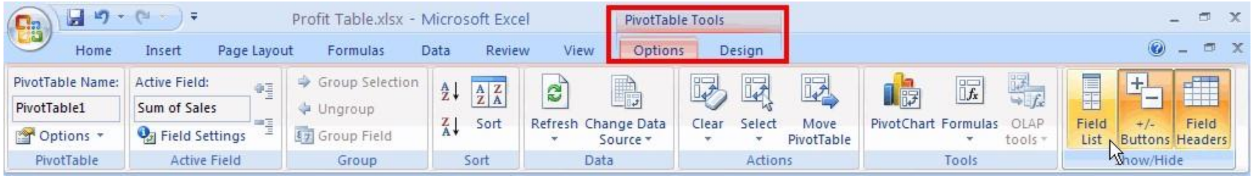
OK Cancel

west	Grand Total
	400
1200	1200
1200	1600

As you now know, you can specify and rearrange data by dragging headings from the PivotTable Field List to the PivotTable itself.

The PivotTable Tools Ribbon

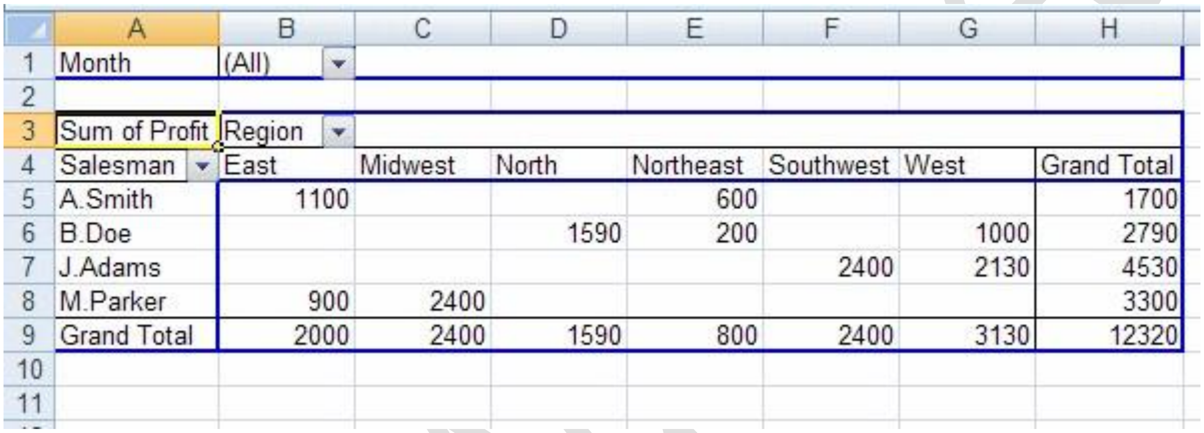
If you close the PivotTable Field List, you can always get it back by clicking the Field List button on the Options Ribbon.



The Options Ribbon and the Design Ribbon will become available whenever you click inside the borders of your PivotTable.

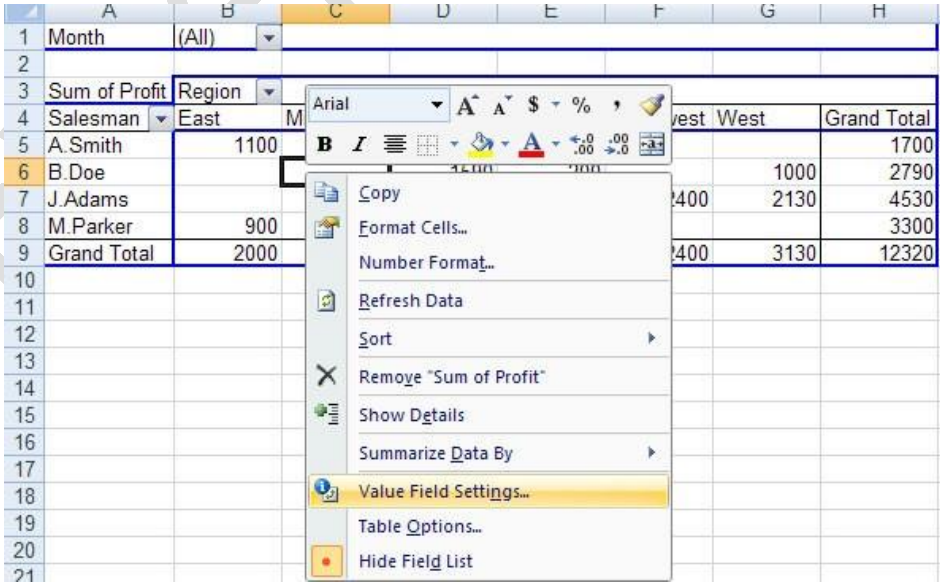
Modifying Calculations and Data Area Display

The default setting for PivotTable calculations is SUM. This means that the totals shown in the Grand Total areas of the PivotTable will be derived from summing the values in the rows and columns.

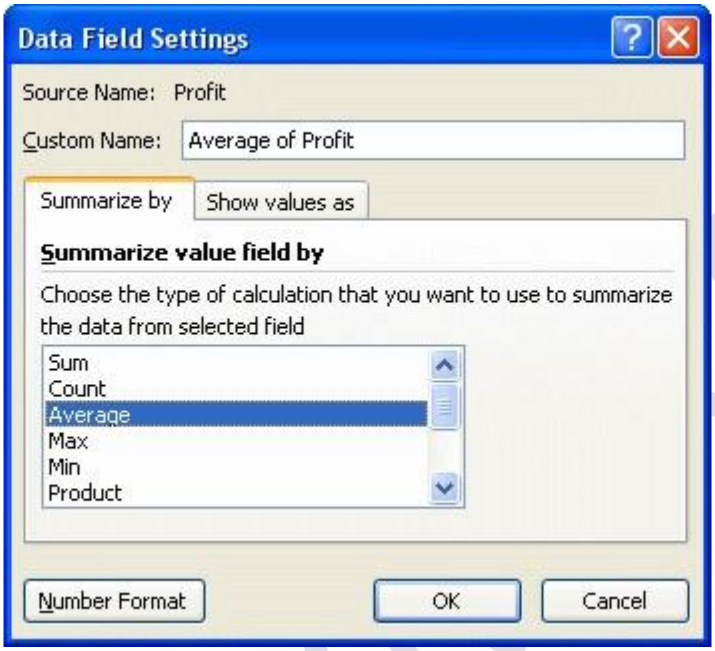


	A	B	C	D	E	F	G	H
1	Month	(All)						
2								
3	Sum of Profit	Region						
4	Salesman	East	Midwest	North	Northeast	Southwest	West	Grand Total
5	A.Smith	1100			600			1700
6	B.Doe			1590	200		1000	2790
7	J.Adams					2400	2130	4530
8	M.Parker	900	2400					3300
9	Grand Total	2000	2400	1590	800	2400	3130	12320
10								
11								

To modify the way these totals are calculated, right click on the Sum of 'X' field in the upper left corner of the PivotTable. This field represents the data area of the PivotTable (you could also right click on any cell in the data area of the PivotTable). When you right click on a cell in the data area or on the cell in the upper left (Sum of 'X'), you will see a pop up menu with several options.



Near the bottom of the menu (third from the bottom in this image), you will see an option called Value Field Settings. If you click on this option, you will display the PivotTable Data Field Settings dialogue box.



Data Field Settings

Source Name: Profit

Custom Name: Average of Profit

Summarize value field by

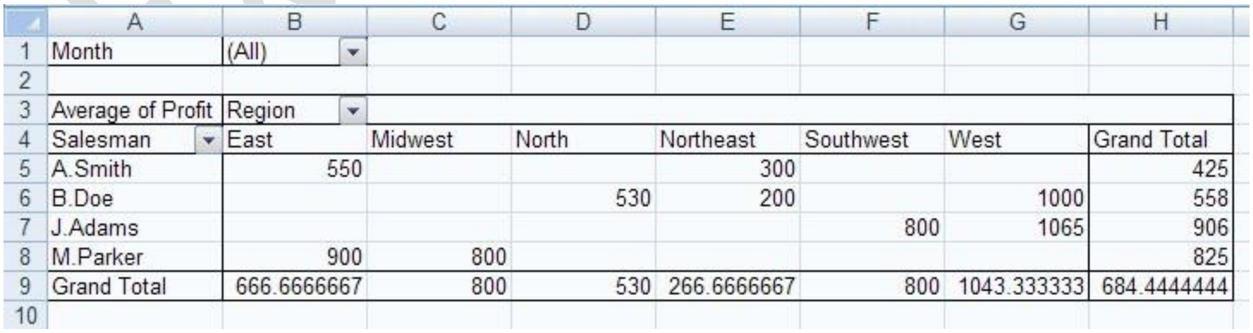
Choose the type of calculation that you want to use to summarize the data from selected field

In this dialogue box, you can switch the type of calculation to Count, Average, Max or Min, Product, Standard deviation, or Variance. For this example, Average has been selected.

If you click the Number Format button, you can choose a format like currency, accounting, or text for the values in the data area of the table.

Under the Show Values As button, you can modify the data area values further by expressing them as percentages or differences.

Clicking OK will incorporate the changes into the PivotTable.



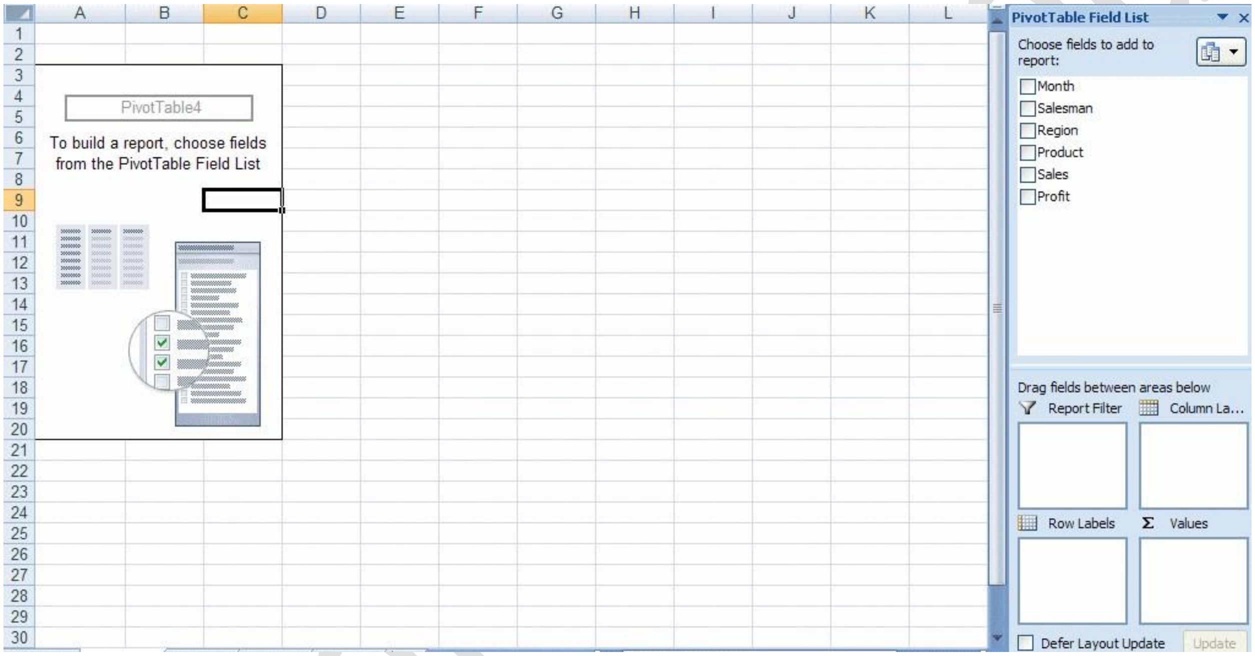
	A	B	C	D	E	F	G	H
1	Month	(All)						
2								
3	Average of Profit	Region						
4	Salesman	East	Midwest	North	Northeast	Southwest	West	Grand Total
5	A.Smith	550			300			425
6	B.Doe			530	200		1000	558
7	J.Adams					800	1065	906
8	M.Parker	900	800					825
9	Grand Total	666.6666667	800	530	266.6666667	800	1043.333333	684.4444444
10								

Here you can see that the data field label in the upper left of the PivotTable now says Average of Profit, and the values in the grand total row and column are averages rather than sums. You can modify any field that you place in the data area of the table in this same way.

Modifying a PivotTable using Excel 2013 PivotTable Layout

The previous discussion has dealt primarily with PivotTable frames, also known as classic PivotTable layouts. Now you will learn how to arrange data in the default Excel 2013 PivotTable layout.

You will remember from the earlier discussion that when you create a standard PivotTable in Excel 2013, it looks like this:


PivotTable4

To build a report, choose fields from the PivotTable Field List

PivotTable Field List

Choose fields to add to report:

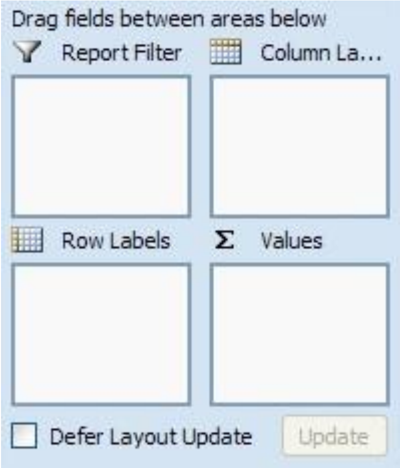
□ Month
□ Salesman
□ Region
□ Product
□ Sales
□ Profit

Drag fields between areas below

□ Defer Layout Update

Here you can see the empty PivotTable on the left and the PivotTable field List on the right.

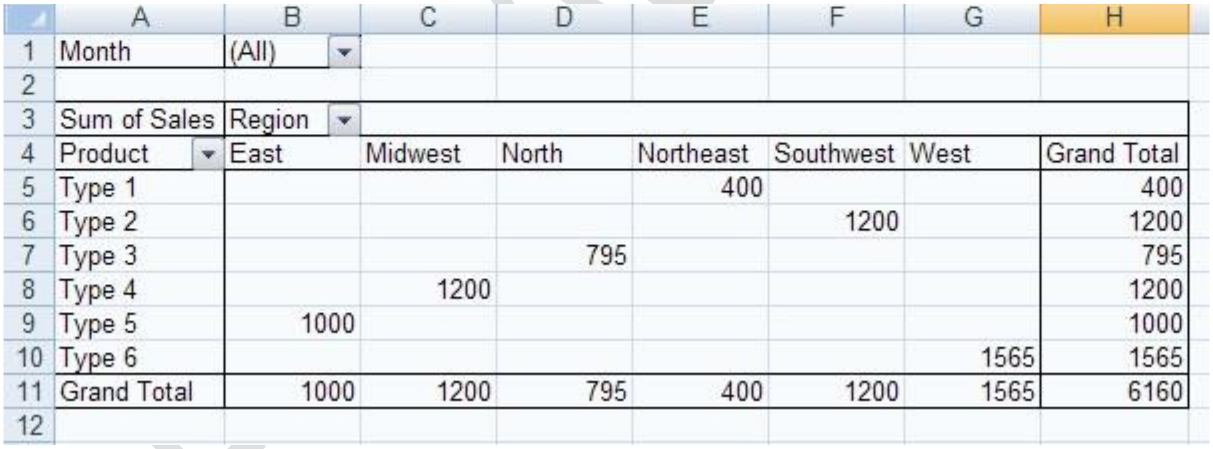
At the bottom of the PivotTable field list, you will notice a panel of four data boxes.



These boxes are named Report Filter, Column Labels, Row Labels, and ∑ Values. They correspond to the Page fields, the Column fields, the Row fields, and the Data Item areas of a classic PivotTable frame.

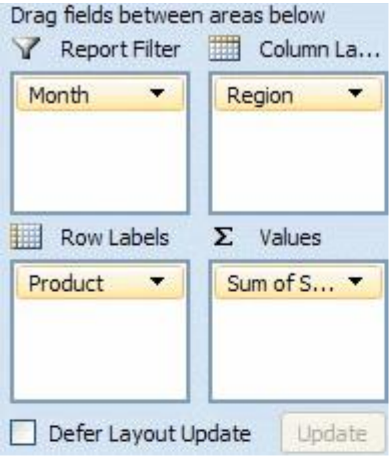
Dragging an item from the PivotTable field list to the ∑ Values data box produces similar results as dragging the item to the Data items area of a classic PivotTable frame. This holds true for the other data boxes as well.

As an example, if we drag Month from the field list to the Report filter box, Region from the field list to the Column Labels box, Product from the field list to the Row Labels box, and Sales from the field list to the ∑ Values box, our resulting PivotTable will look like the following.

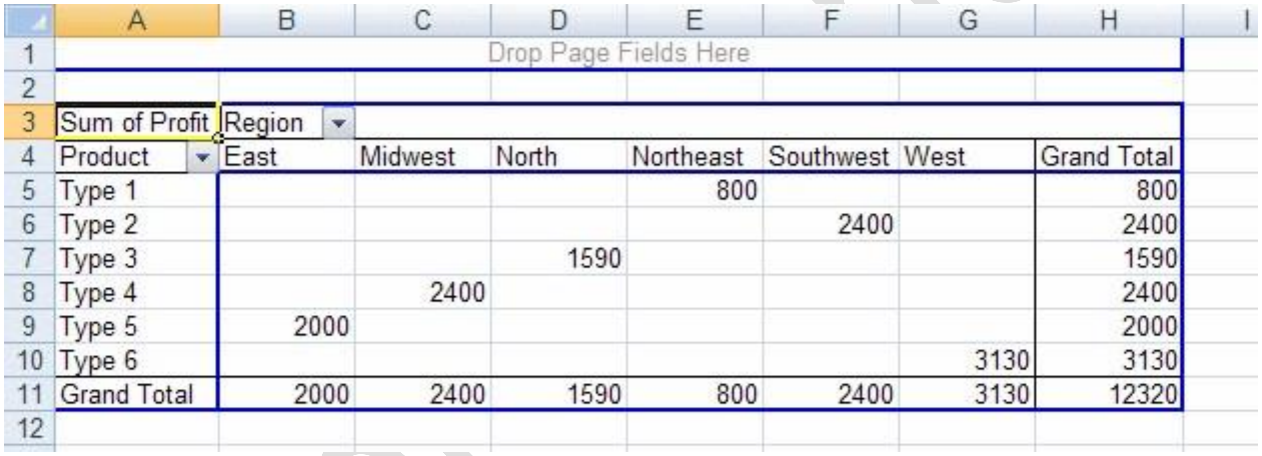


	A	B	C	D	E	F	G	H
1	Month	(All)						
2								
3	Sum of Sales	Region						
4	Product	East	Midwest	North	Northeast	Southwest	West	Grand Total
5	Type 1				400			400
6	Type 2					1200		1200
7	Type 3			795				795
8	Type 4		1200					1200
9	Type 5	1000						1000
10	Type 6						1565	1565
11	Grand Total	1000	1200	795	400	1200	1565	6160
12								

The corresponding panel of data boxes will look like this.



You can easily drag field headings to and from the data boxes in the panel to rearrange the layout of the PivotTable.

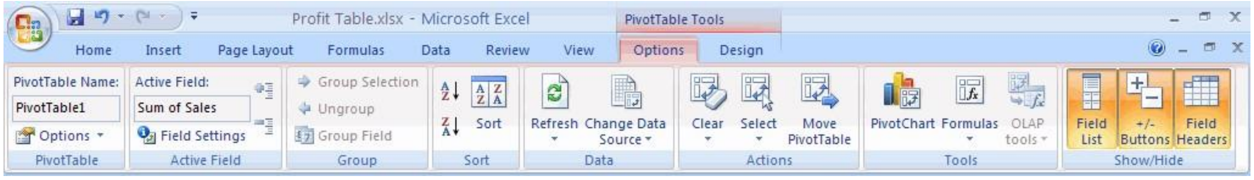


	A	B	C	D	E	F	G	H
1	Drop Page Fields Here							
2								
3	Sum of Profit	Region						
4	Product	East	Midwest	North	Northeast	Southwest	West	Grand Total
5	Type 1				800			800
6	Type 2					2400		2400
7	Type 3			1590				1590
8	Type 4		2400					2400
9	Type 5	2000						2000
10	Type 6						3130	3130
11	Grand Total	2000	2400	1590	800	2400	3130	12320
12								

Here is a classic PivotTable showing profit per product, across regions. Notice that the classic table has thick blue borders.

Formatting a PivotTable

To format a PivotTable, first click on any cell in the table to display the Options tab near the top of your Excel screen. When you click the Options tab, the Options Ribbon will appear.



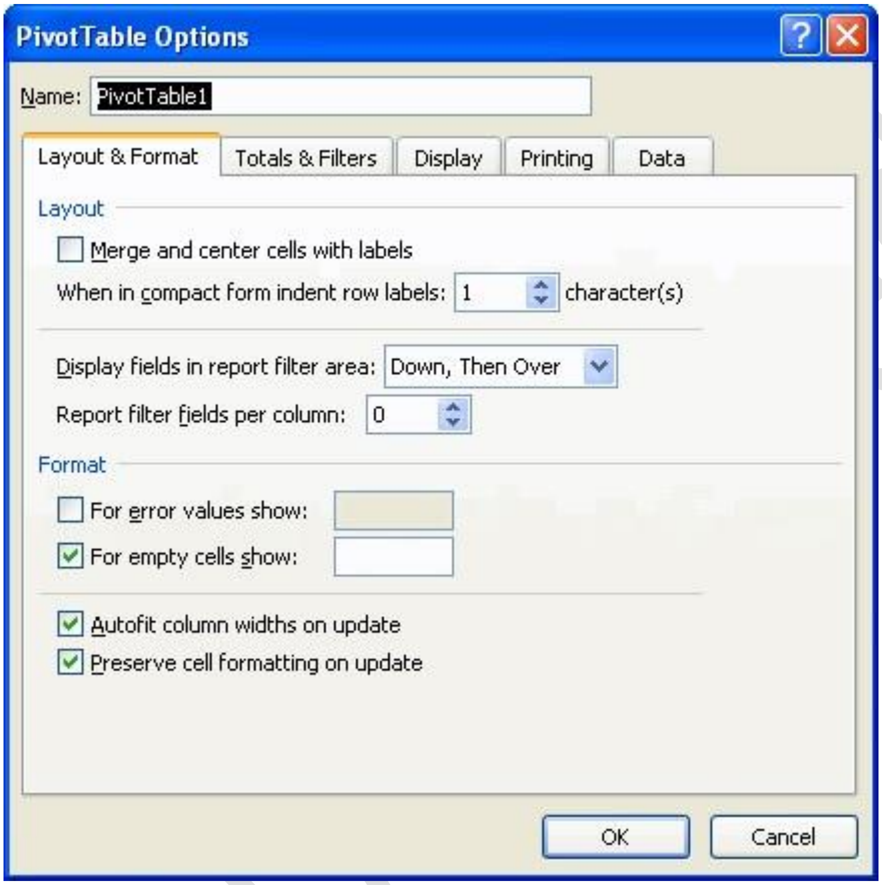
Profit Table.xlsx - Microsoft Excel

PivotTable Name: PivotTable1

Active Field: Sum of Sales

Using PivotTables Options Ribbon

At the left of the Options Ribbon, in the PivotTable group, you will find an Options button. Clicking this button will display the PivotTable Options dialogue box.



PivotTable Options

Name: PivotTable1

Layout

□ Merge and center cells with labels

When in compact form indent row labels: 1 character(s)

Display fields in report filter area: Down, Then Over

Report filter fields per column: 0

Format

□ For error values show:

☑ For empty cells show:

☑ Autofit column widths on update

☑ Preserve cell formatting on update

Under the Layout & Format tab, make sure the “Preserve cell formatting on update” box is checked (it should be by default). This will ensure that any formatting you apply will be retained if the table is modified or refreshed.

Under the Totals & Filters tab, you can specify if grand totals for rows or columns will be shown.

Under the Display tab, you can configure how filter drop downs, field captions, expand/collapse buttons, and contextual tool tips are shown.

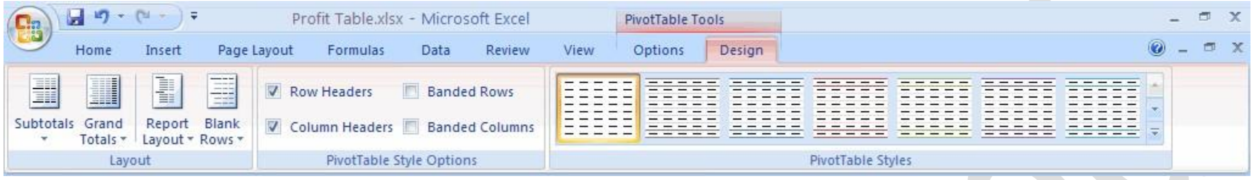
Under the Printing tab you can alter the layout of a PivotTable printout.

Finally, under the Data tab you can specify PivotTable data options, like having the table refreshed when the workbook opens.

Click OK when you are finished with the PivotTable options dialogue.

Using PivotTable Design Ribbon

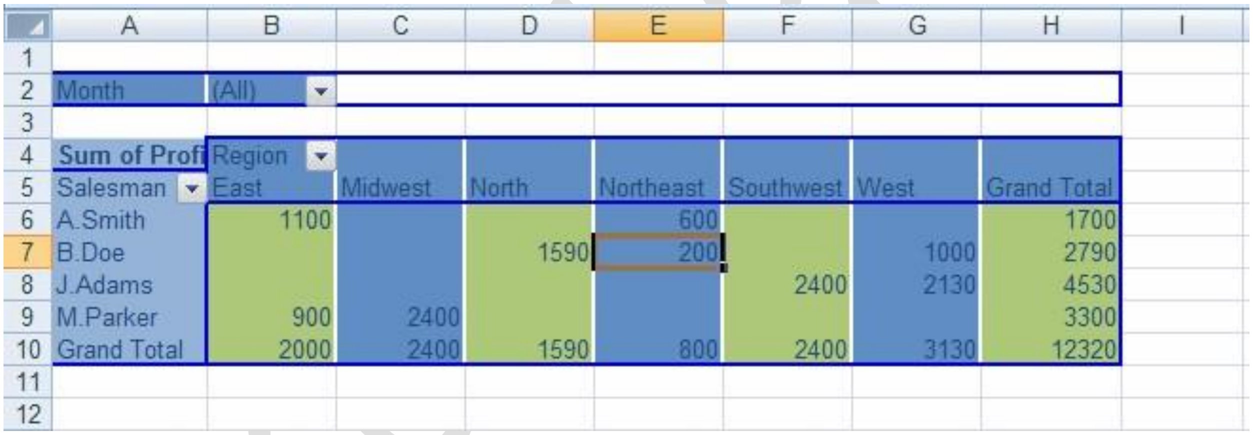
In Excel 2013, the easiest way to format your PivotTable is to use the new PivotTable Styles that are provided on the Design Ribbon. When you click on a cell in your PivotTable, an Options tab and a Design tab will appear near the top of the Excel screen. Clicking on the Design tab will reveal the Design Ribbon.



☑ Row Headers □ Banded Rows
☑ Column Headers □ Banded Columns

On the Design Ribbon, you can click on one of the preset styles in the PivotTable Styles group to quickly apply preset formatting to the table. There are also checkboxes in the PivotTable Style Options section of the ribbon that will let you specify banded columns or rows, row headers, or column headers. At the left of the ribbon are controls that you can use to view or hide subtotals and grand totals and to specify a PivotTable report layout type.

Here is a style with banded rows applied to an existing PivotTable.



	A	B	C	D	E	F	G	H	I
1									
2	Month	(All)							
3									
4	Sum of Profi	Region							
5	Salesman	East	Midwest	North	Northeast	Southwest	West	Grand Total	
6	A.Smith	1100			600			1700	
7	B.Doe			1590	200		1000	2790	
8	J.Adams					2400	2130	4530	
9	M.Parker	900	2400					3300	
10	Grand Total	2000	2400	1590	800	2400	3130	12320	
11									
12									

Using the Home Ribbon

If you would like to apply specific formatting to a table that is not available as a style, you can always select a range of cells from the table and use the buttons on the Home Ribbon to format the selected range. Another method is to select a range of cells from the table, right click in the selected range, and choose Format Cells from the drop down menu. You can then choose from a series of tabs in the format cells dialogue box that will let you select number formats, font styles, borders, and protection options.

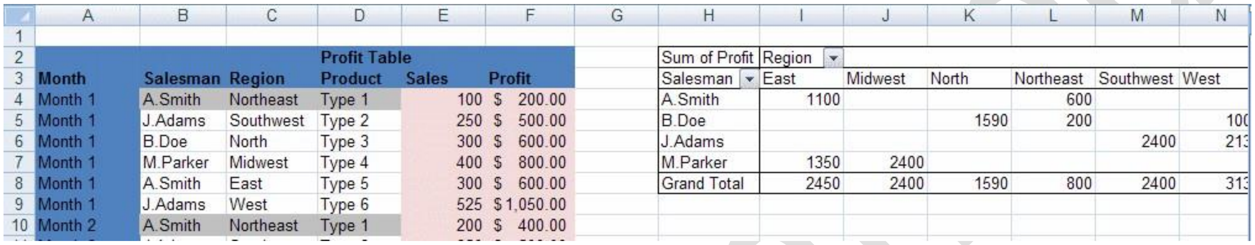
Refreshing a PivotTable

The data in a PivotTable is not linked directly to the source table or range. Instead, the PivotTable is based on a hidden copy of the source data that is kept in memory by Excel. This means that changes to the original source list or database will not be automatically updated in the PivotTable. If you make changes in the source data, you must refresh the PivotTable to update it.

You can refresh a PivotTable by clicking the Refresh button on the Data Ribbon or on the Options Ribbon.

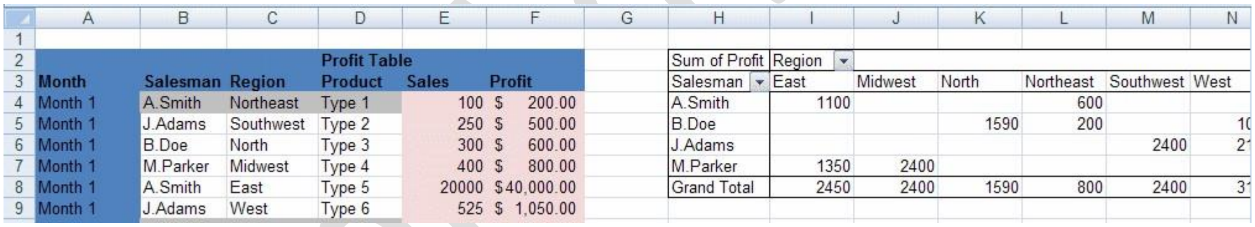
You could also right click on a cell in the data area of the table and then click the Refresh Data option from the pop up menu.

Here is a range of data and a PivotTable that uses the data range as its source.



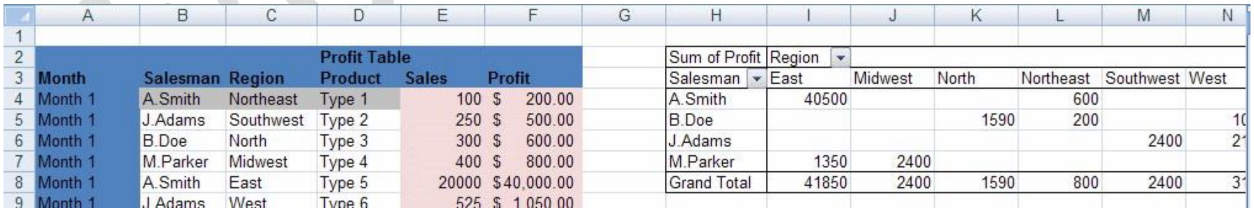
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				Profit Table				Sum of Profit	Region ▾					
3	Month	Salesman	Region	Product	Sales	Profit		Salesman ▾	East	Midwest	North	Northeast	Southwest	West
4	Month 1	A.Smith	Northeast	Type 1	100	$ 200.00		A.Smith	1100			600		
5	Month 1	J.Adams	Southwest	Type 2	250	$ 500.00		B.Doe			1590	200		10
6	Month 1	B.Doe	North	Type 3	300	$ 600.00		J.Adams					2400	21
7	Month 1	M.Parker	Midwest	Type 4	400	$ 800.00		M.Parker	1350	2400				
8	Month 1	A.Smith	East	Type 5	300	$ 600.00		Grand Total	2450	2400	1590	800	2400	31
9	Month 1	J.Adams	West	Type 6	525	$1,050.00								
10	Month 2	A.Smith	Northeast	Type 1	200	$ 400.00								

If we change the sales for A Smith in the East, (in the source data range), there will be no change in the PivotTable. (The data in cell E8 has been changed to 20000 and the corresponding profit in cell F8 is now 40000.)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				Profit Table				Sum of Profit	Region ▾					
3	Month	Salesman	Region	Product	Sales	Profit		Salesman ▾	East	Midwest	North	Northeast	Southwest	West
4	Month 1	A.Smith	Northeast	Type 1	100	$ 200.00		A.Smith	1100			600		
5	Month 1	J.Adams	Southwest	Type 2	250	$ 500.00		B.Doe			1590	200		1
6	Month 1	B.Doe	North	Type 3	300	$ 600.00		J.Adams					2400	2
7	Month 1	M.Parker	Midwest	Type 4	400	$ 800.00		M.Parker	1350	2400				
8	Month 1	A.Smith	East	Type 5	20000	$40,000.00		Grand Total	2450	2400	1590	800	2400	3
9	Month 1	J.Adams	West	Type 6	525	$ 1,050.00								

Here is the same data and PivotTable after the PivotTable has been refreshed. Notice that cell I4 now has a value of 40500.



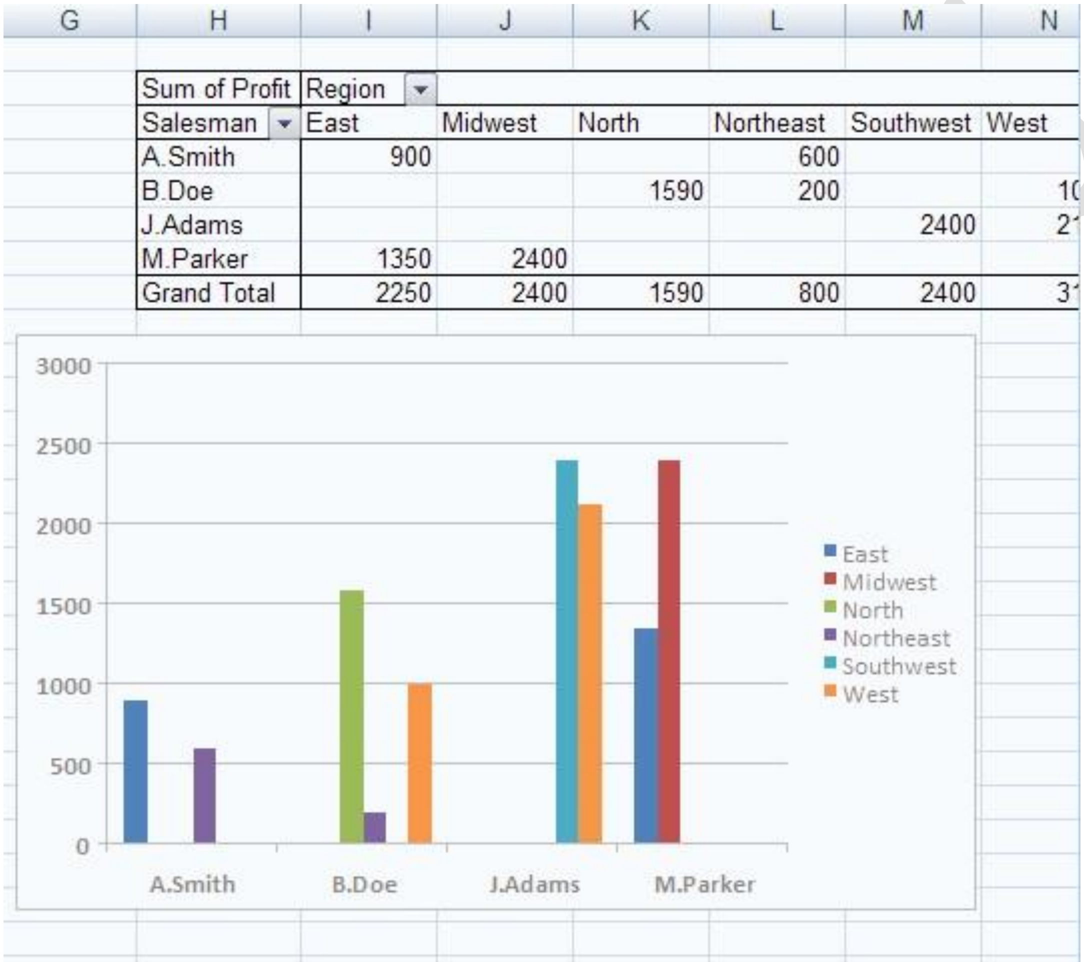
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				Profit Table				Sum of Profit	Region ▾					
3	Month	Salesman	Region	Product	Sales	Profit		Salesman ▾	East	Midwest	North	Northeast	Southwest	West
4	Month 1	A.Smith	Northeast	Type 1	100	$ 200.00		A.Smith	40500			600		
5	Month 1	J.Adams	Southwest	Type 2	250	$ 500.00		B.Doe			1590	200		1
6	Month 1	B.Doe	North	Type 3	300	$ 600.00		J.Adams					2400	2
7	Month 1	M.Parker	Midwest	Type 4	400	$ 800.00		M.Parker	1350	2400				
8	Month 1	A.Smith	East	Type 5	20000	$40,000.00		Grand Total	41850	2400	1590	800	2400	3
9	Month 1	J.Adams	West	Type 6	525	$ 1,050.00								

Creating a PivotChart

In Excel 2013, you can create a PivotChart from scratch, or you can create a PivotChart based on an existing PivotTable

From an Existing PivotTable

To create a chart based on an existing PivotTable, simply click a cell in the table and then click on a chart type from the Charts area on the Insert Ribbon. You can select any chart type except XY Scatter, Bubble, or Stock charts.



Sum of Profit	Region ▾					
Salesman ▾	East	Midwest	North	Northeast	Southwest	West
A.Smith	900			600		
B.Doe			1590	200		10
J.Adams					2400	21
M.Parker	1350	2400				
Grand Total	2250	2400	1590	800	2400	31

In the image above, a basic column chart has been selected and created from an existing PivotTable. This particular chart shows profits generated by each salesman broken down over regions.

Any changes or rearranging of field values in the PivotTable will be automatically reflected in the PivotChart.

If there are changes made to the source data for the PivotTable, any of the Refresh options previously mentioned will update both the PivotTable and the PivotChart. This makes the PivotChart every bit as malleable as the PivotTable upon which it is based.

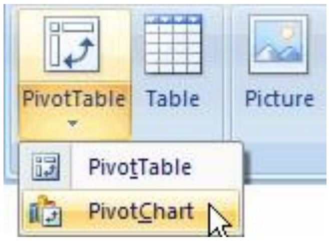
Once you create your chart, you can click on your chart area and then display the Design

Ribbon which will allow you to apply a preset format to your chart via the Chart Styles buttons.

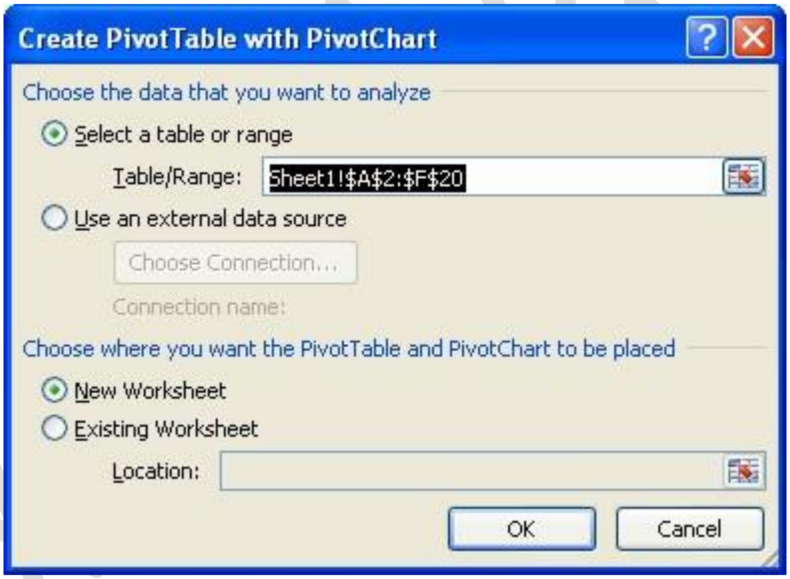
Keep in mind that all chart types may not be equally suited for displaying your PivotTable data. Pick the chart type that most clearly illustrates the point you want to put forward.

Without an Existing PivotTable

You can also create a PivotTable/PivotChart combination from scratch by choosing the PivotChart option from the PivotTable button's pop up menu. Remember, the PivotTable button is located on the Insert Ribbon.



When you click the PivotChart option in the PivotTable button menu, you will see the following PivotTable with PivotChart dialogue box.



Create PivotTable with PivotChart

Choose the data that you want to analyze

● Select a table or range

Table/Range: Sheet1!$A$2:$F$20

○ Use an external data source

Choose Connection...

Connection name:

Choose where you want the PivotTable and PivotChart to be placed

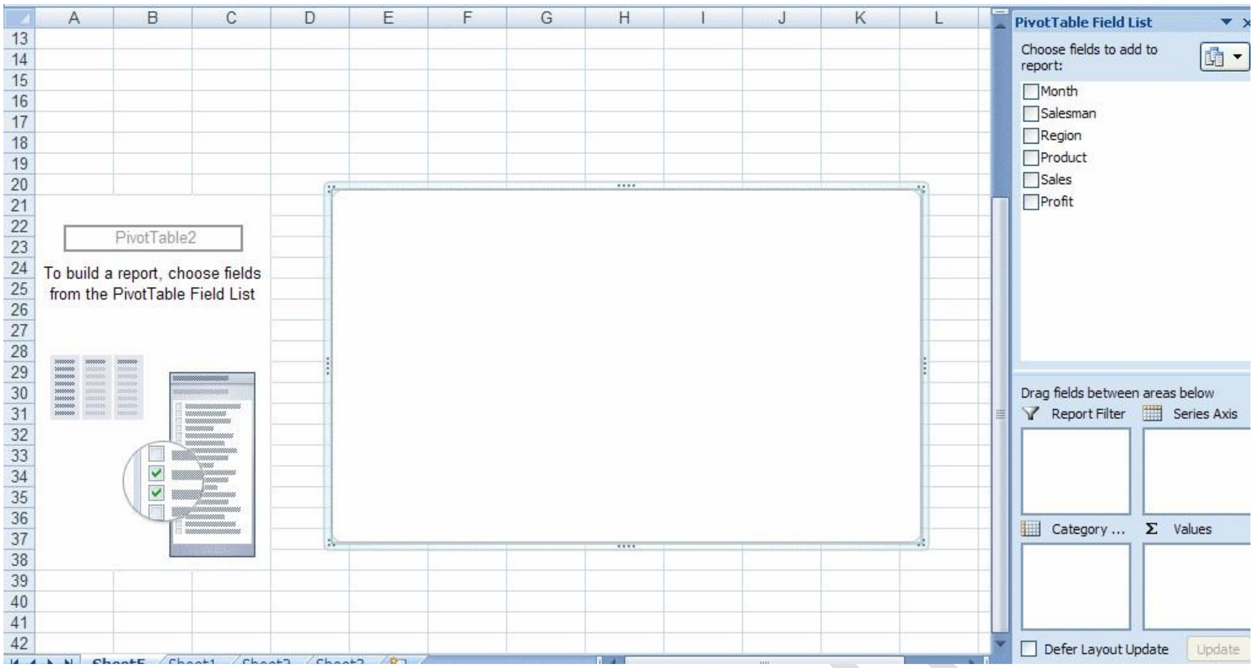
● New Worksheet

○ Existing Worksheet

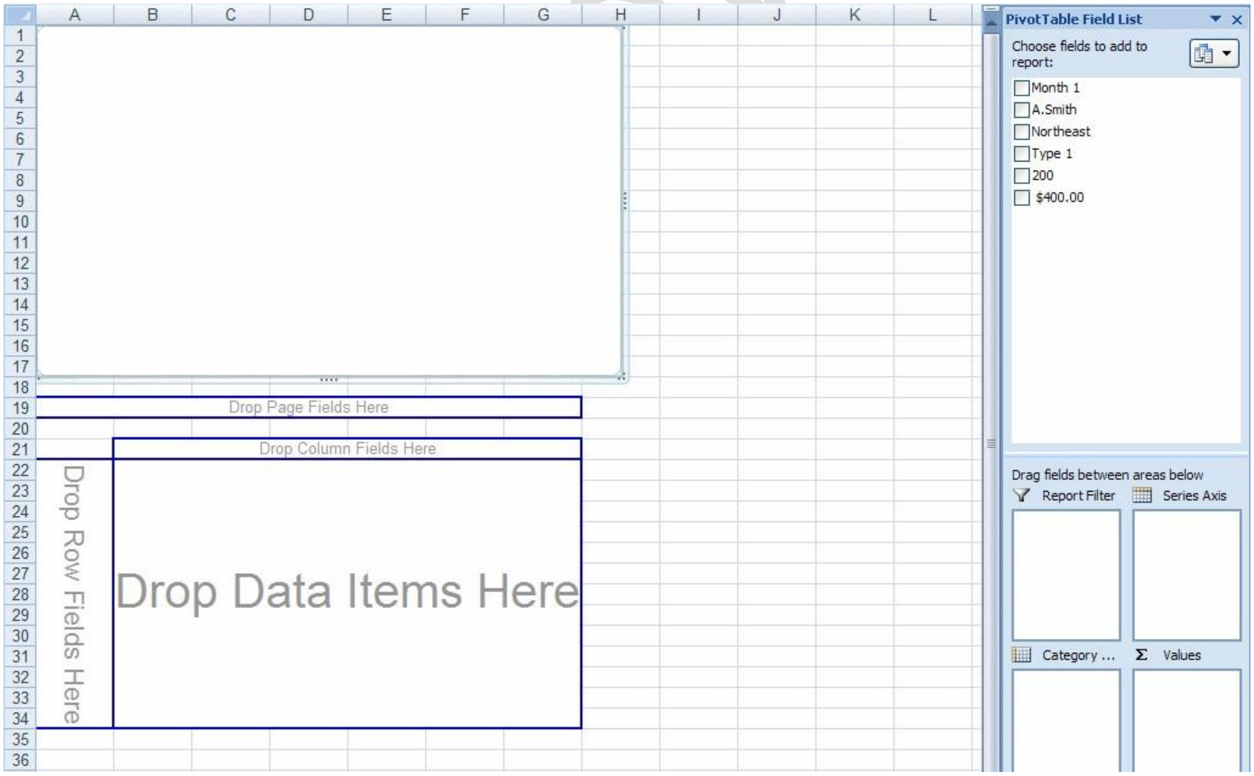
Location:

OK Cancel

In this dialogue, click in the Table/Range field to place a cursor inside and then select the range of cells that you want to use from the spreadsheet. You can also select the range before invoking the dialogue if you wish. Once the range or table has been selected, you can choose to place the new table/chart on a new worksheet, or on the existing worksheet, by selecting the appropriate radio button. In the image above, the New Worksheet option has been selected. As soon as you are ready, click the OK button to create the new PivotTable/PivotChart combination.



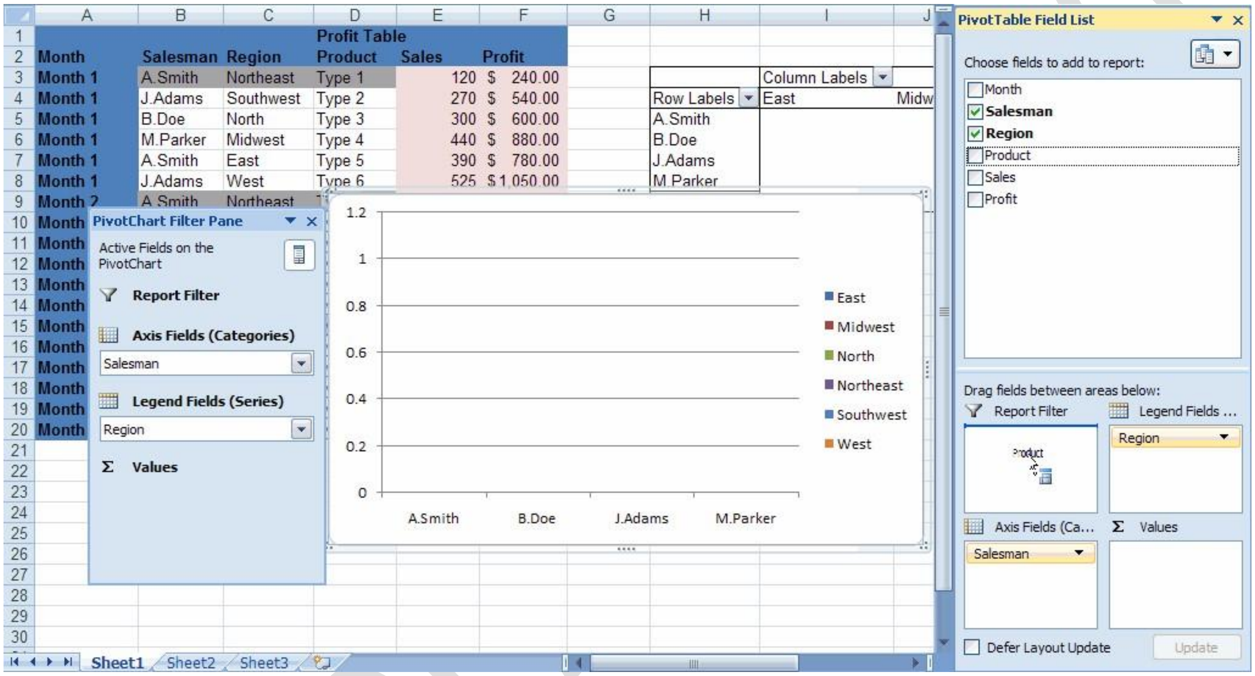
If you wish, you can right click on the PivotTable area, select the PivotTable Options item from the Pop up menu, and then choose the Display tab and check the Classic PivotTable Layout option to create a classic frame.



Here you can see a classic PivotTable frame, empty PivotChart, and PivotTable field list.

Your PivotChart will first appear empty, very much like your basic empty PivotTable. When you drag field headings from the PivotTable field list over to your Table frame, the data from the fields will also begin to populate the chart. In a sense, the layout that you create for the actual PivotTable will be mirrored automatically in the PivotChart.

If you prefer to build your PivotTable without switching to the classic layout, the chart will be created automatically as you drag your field headings to the panel of data boxes at the bottom of the PivotTable field list.



Profit Table

Month	Salesman	Region	Product	Sales	Profit
Month 1	A.Smith	Northeast	Type 1	120	$ 240.00
Month 1	J.Adams	Southwest	Type 2	270	$ 540.00
Month 1	B.Doe	North	Type 3	300	$ 600.00
Month 1	M.Parker	Midwest	Type 4	440	$ 880.00
Month 1	A.Smith	East	Type 5	390	$ 780.00
Month 1	J.Adams	West	Type 6	525	$ 1,050.00
Month 2	A.Smith	Northeast	T		

	Column Labels	
Row Labels	East	Midw
A.Smith		
B.Doe		
J.Adams		
M.Parker		

PivotChart Filter Pane

Active Fields on the PivotChart

Report Filter

Axis Fields (Categories): Salesman

Legend Fields (Series): Region

Values

PivotTable Field List

Choose fields to add to report:

☐ Month
☑ Salesman
☑ Region
☐ Product
☐ Sales
☐ Profit

Drag fields between areas below:

Report Filter: Product

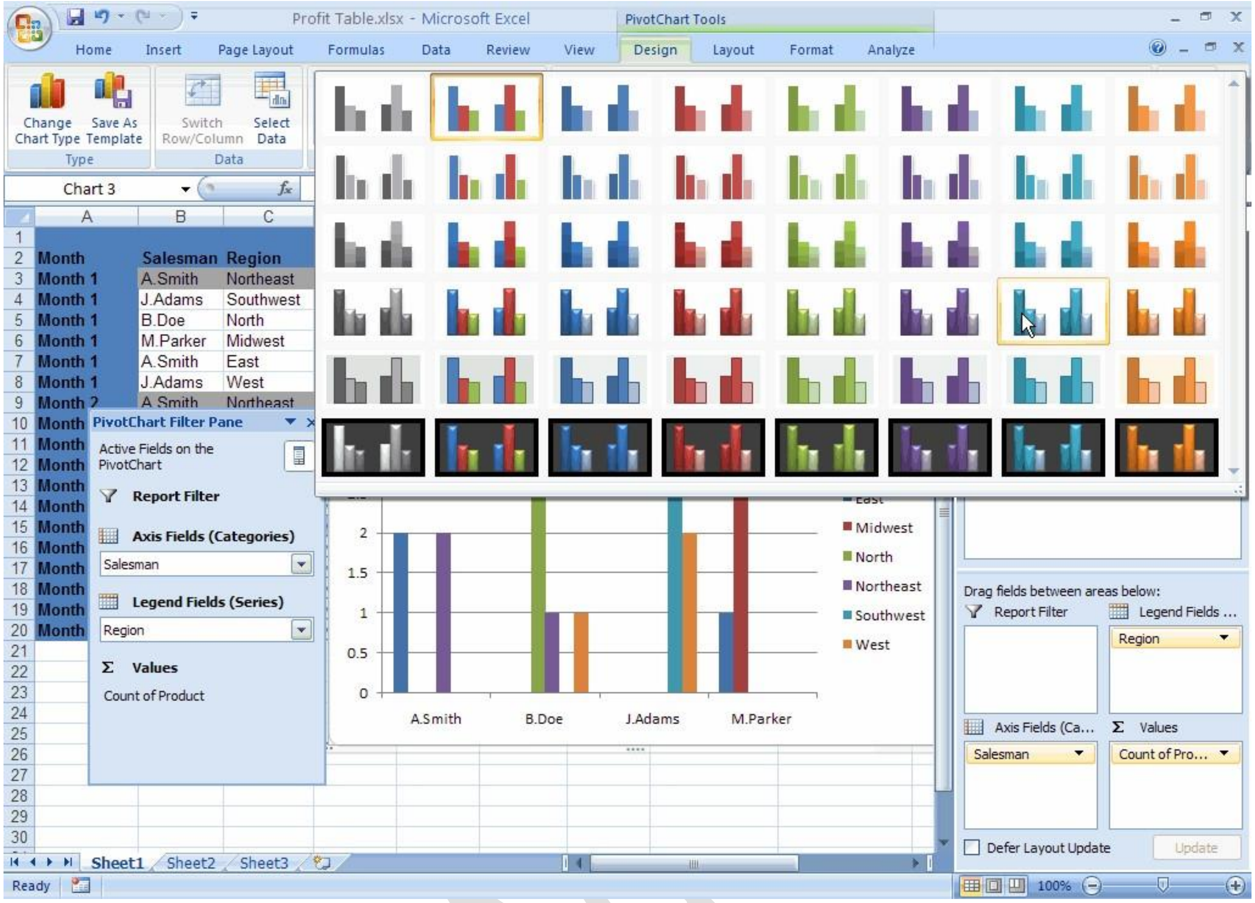
Legend Fields: Region

Axis Fields (Ca...: Salesman

Values

☐ Defer Layout Update

Here is the chart and PivotTable after dragging field headings from the PivotTable field list to the appropriate table locations. You can easily format the PivotChart by clicking on the chart area, and then using the Styles on the Design Ribbon to choose a preset design.



	A	B	C
1			
2	Month	Salesman	Region
3	Month 1	A.Smith	Northeast
4	Month 1	J.Adams	Southwest
5	Month 1	B.Doe	North
6	Month 1	M.Parker	Midwest
7	Month 1	A.Smith	East
8	Month 1	J.Adams	West
9	Month 2	A.Smith	Northeast

Axis Fields (Categories): Salesman

Legend Fields (Series): Region

Values: Count of Product

Region

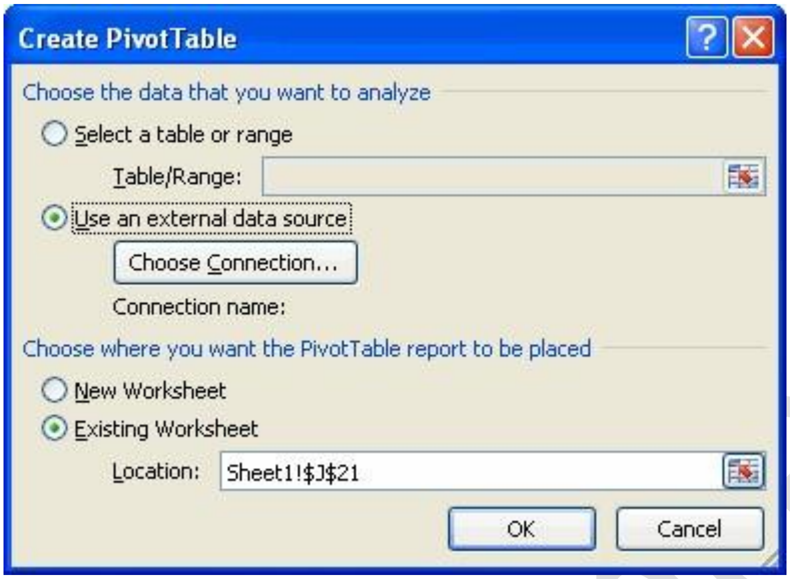
Salesman

Count of Pro...

(Note the new ribbons available for formatting and editing PivotCharts.)

Creating a PivotTable Based on External Data

When you select the Insert Ribbon, PivotTable button, you can select the Use an External Data Source radio button to create a PivotTable from external data.



Create PivotTable

Choose the data that you want to analyze

○ Select a table or range

Table/Range:

● Use an external data source

Choose Connection...

Connection name:

Choose where you want the PivotTable report to be placed

○ New Worksheet

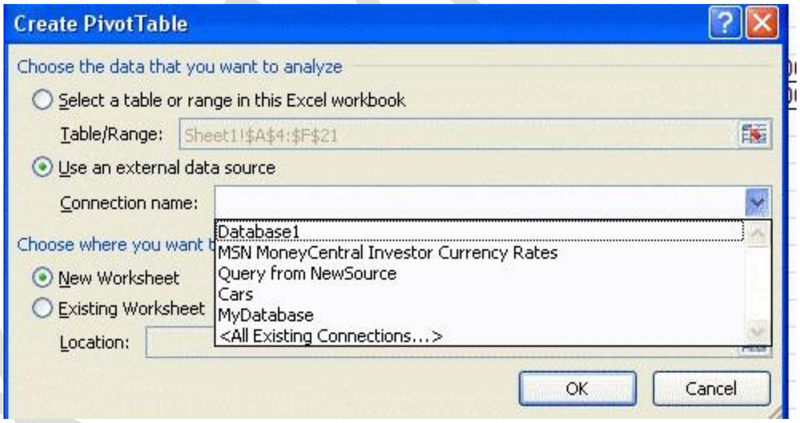
● Existing Worksheet

Location: Sheet1!$J$21

OK Cancel

Connecting to External Data

When you do this, you will be able to select from a list of existing connections (if there are any) to use as a data source. (If there are no existing connections, you will see the Choose Connection button as shown in the figure above.)



Create PivotTable

Choose the data that you want to analyze

○ Select a table or range in this Excel workbook

Table/Range: Sheet1!$A$4:$F$21

● Use an external data source

Connection name:

- Database1
- MSN MoneyCentral Investor Currency Rates
- Query from NewSource
- Cars
- MyDatabase
- <All Existing Connections...>

Choose where you want t

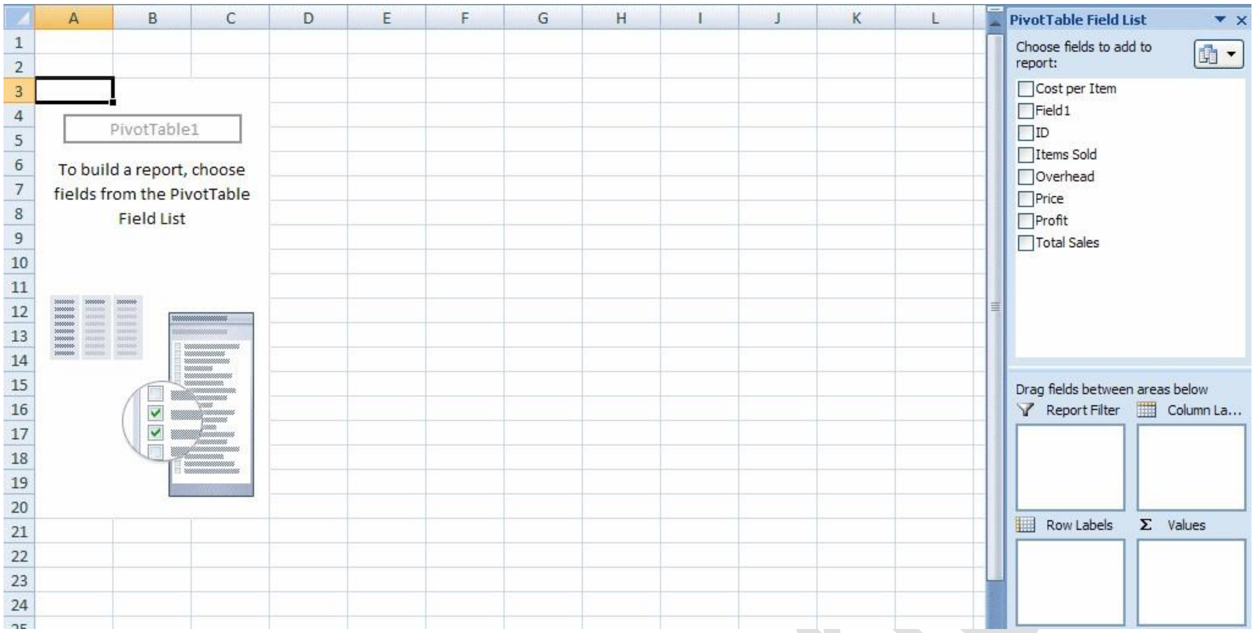
● New Worksheet

○ Existing Worksheet

Location:

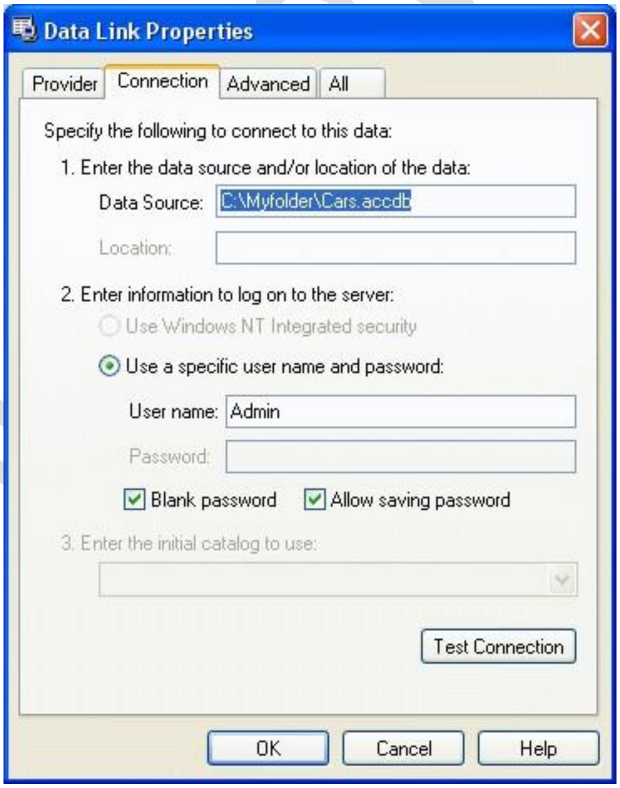
OK Cancel

When you select an item from the list and click OK, you will see the PivotTable area and PivotTable field list arrangement as before, the difference being that now the field headings in the PivotTable field list will be from the external data source that you selected.



You can now drag and drop the field headings into the panel of data boxes to build your table.

Depending on the external data source you choose, it is possible that you may have to enter a user name/password combination to get access to the data (as shown below).


Data Link Properties

Provider | Connection | Advanced | All

Specify the following to connect to this data:

1. Enter the data source and/or location of the data:

Data Source: C:\Myfolder\Cars.accdb

Location:

2. Enter information to log on to the server:

○ Use Windows NT Integrated security

● Use a specific user name and password:

User name: Admin

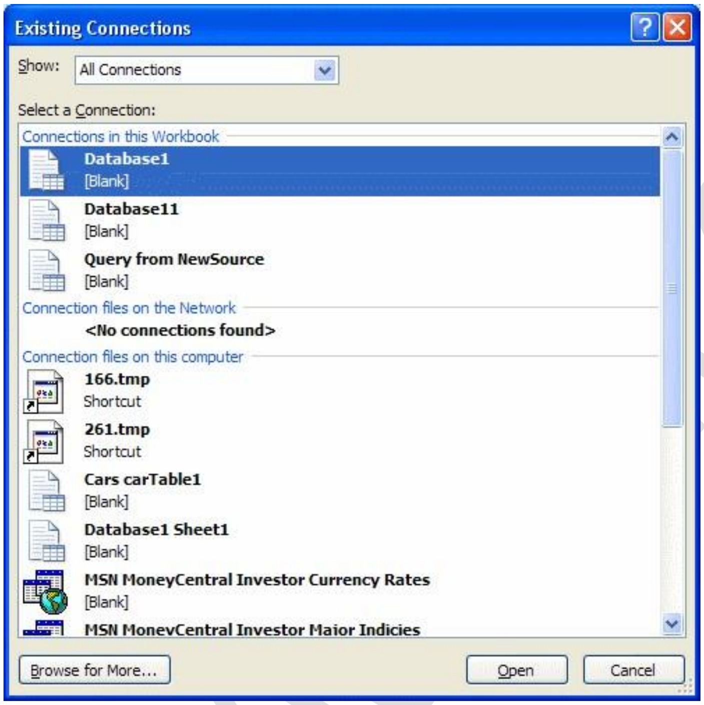
Password:

☑ Blank password ☑ Allow saving password

3. Enter the initial catalog to use:

Once you have entered the appropriate information in the fields provided, click the OK button to begin creating your PivotTable.

You can also click on the Existing Connections button in the Data Ribbon and then select a data source from the dialogue box that appears.



Existing Connections

Show: All Connections

Select a Connection:

Connections in this Workbook

- Database1 [Blank]
- Database11 [Blank]
- Query from NewSource [Blank]

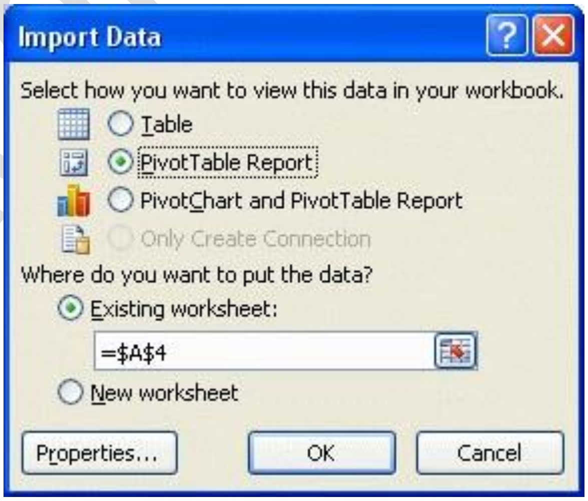
Connection files on the Network

- <No connections found>

Connection files on this computer

- 166.tmp Shortcut
- 261.tmp Shortcut
- Cars carTable1 [Blank]
- Database1 Sheet1 [Blank]
- MSN MoneyCentral Investor Currency Rates [Blank]
- MSN MoneyCentral Investor Major Indicies

Once you select one of the data source connections, click the Open button in the lower right to reveal the Import Data panel.



Import Data

Select how you want to view this data in your workbook.

○ Table
● PivotTable Report
○ PivotChart and PivotTable Report
○ Only Create Connection

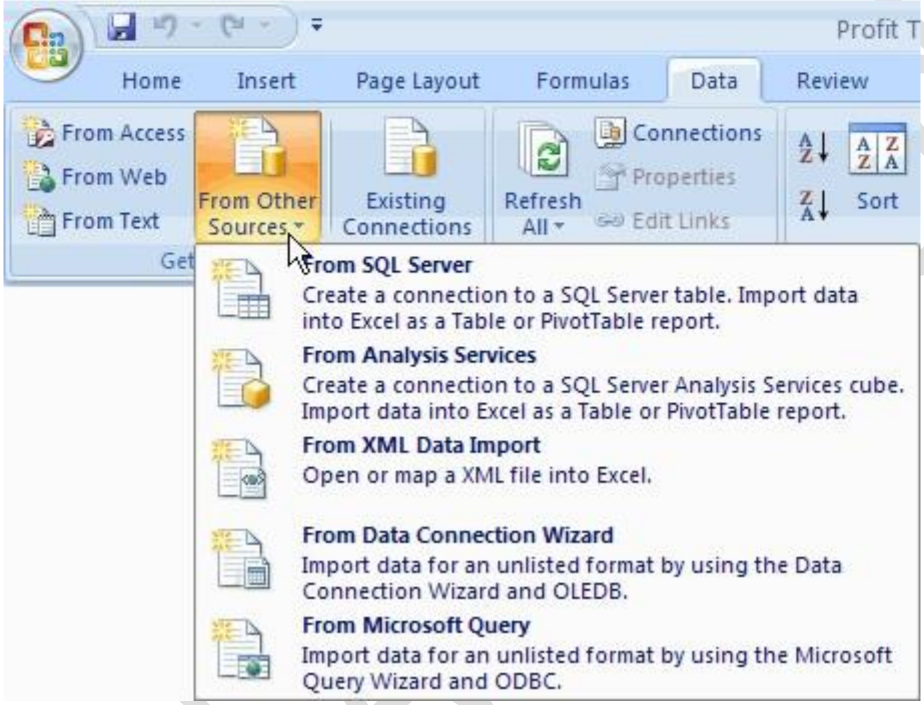
Where do you want to put the data?

● Existing worksheet:
=$A$4
○ New worksheet

Once you see this panel of controls, you can specify a Table, a PivotTable Report, or a

PivotChart and PivotTable report, for your imported data. You can also specify a new or existing worksheet for your PivotTable location. Once you are finished making your selections, click the OK button to create your PivotTable frame and PivotTable field list.

If you do not have any existing connections, you can create a new one as long as you have access to an external data source. To create a new connection, use the From Other Sources button on the Data Ribbon.



From SQL Server
Create a connection to a SQL Server table. Import data into Excel as a Table or PivotTable report.

From Analysis Services
Create a connection to a SQL Server Analysis Services cube. Import data into Excel as a Table or PivotTable report.

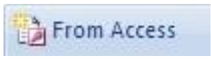
From XML Data Import
Open or map a XML file into Excel.

From Data Connection Wizard
Import data for an unlisted format by using the Data Connection Wizard and OLEDB.

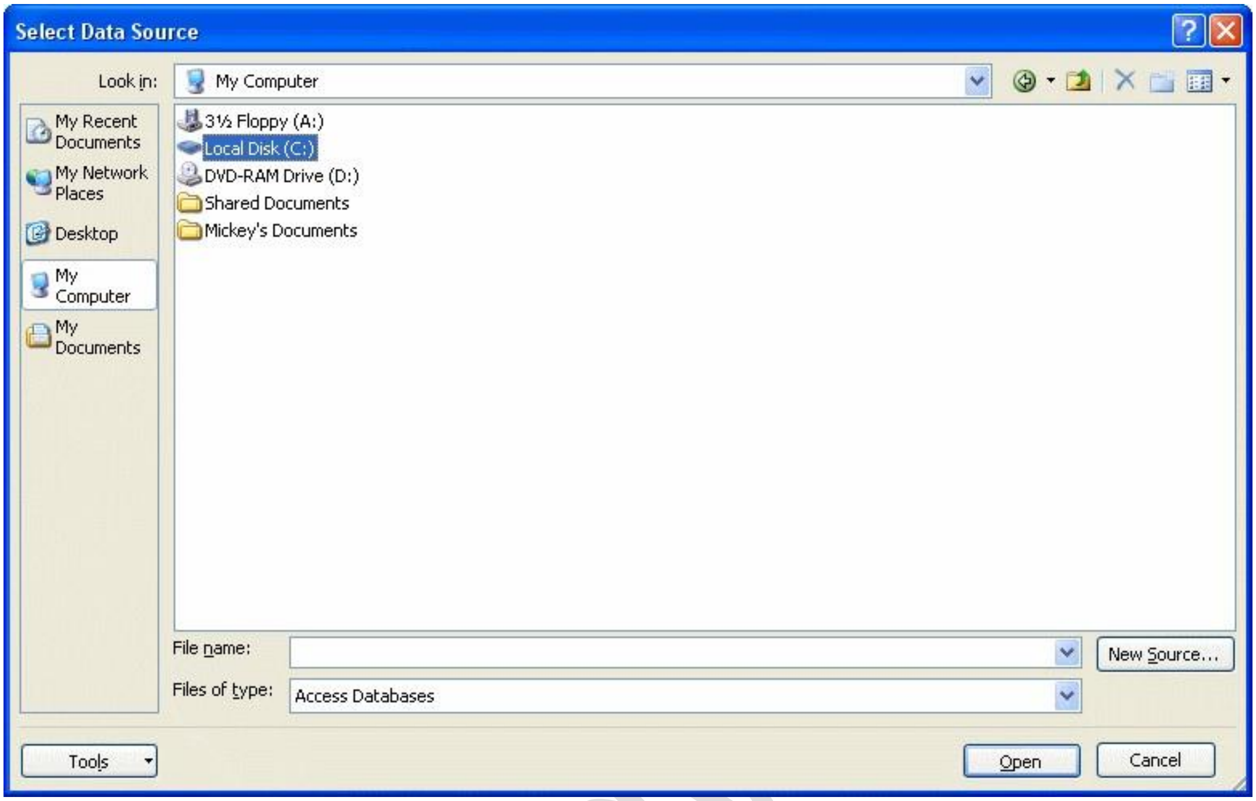
From Microsoft Query
Import data for an unlisted format by using the Microsoft Query Wizard and ODBC.

You can create a connection to an SQL server, an Analysis Services cube, or with Microsoft Query if you wish. Once you have created a connection to an external data source, you can proceed to create your PivotTable by following the same procedure used for an existing connection.

You can import data directly from an Access database by using the From Access button on the Data Ribbon.



When you click this button, a Select Data Source dialogue box will appear, allowing you to navigate to a given Access database.



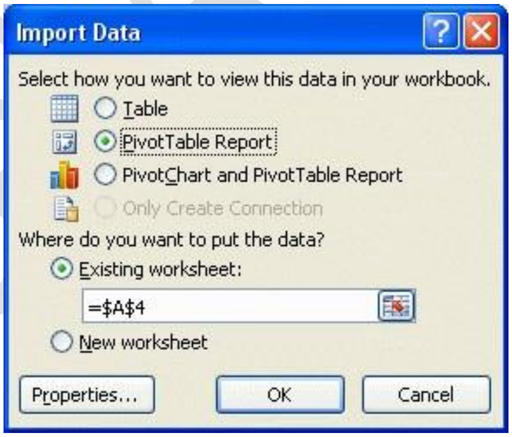
Select Data Source

Look in: My Computer

File name:

Files of type: Access Databases

When you select a database and click the open button in the lower right of the dialogue, you will be presented with the familiar Import Data panel.



Import Data

Select how you want to view this data in your workbook.

○ Table
● PivotTable Report
○ PivotChart and PivotTable Report
○ Only Create Connection

Where do you want to put the data?

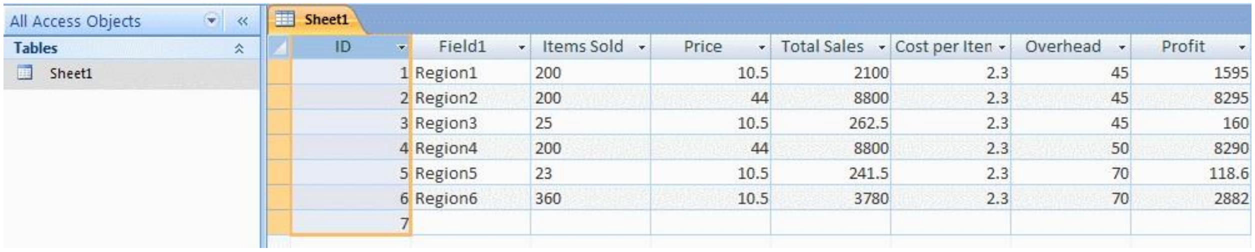
● Existing worksheet:

=$A$4

○ New worksheet

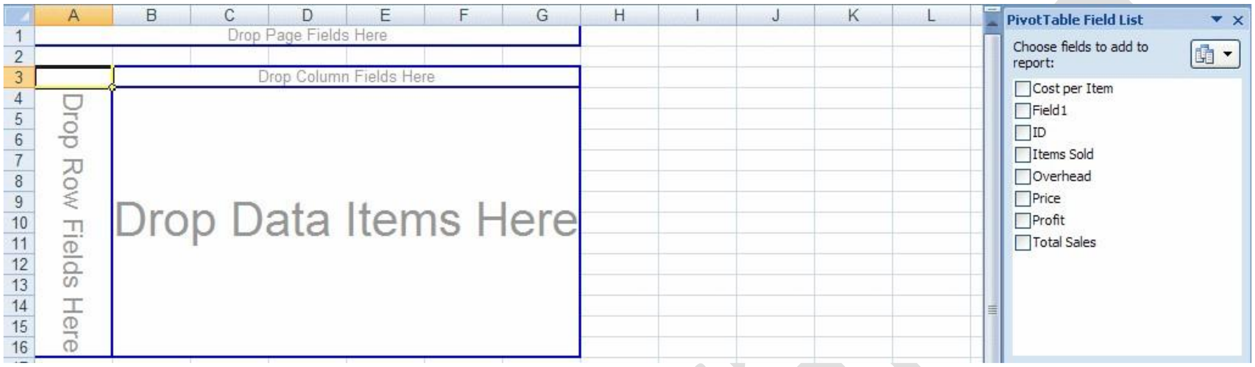
Once again, you can use the options in the import data panel to specify the type and location of the table you want to build from the imported data.

Here is an Access database table containing the source data.



ID	Field1	Items Sold	Price	Total Sales	Cost per Iten	Overhead	Profit
1	Region1	200	10.5	2100	2.3	45	1595
2	Region2	200	44	8800	2.3	45	8295
3	Region3	25	10.5	262.5	2.3	45	160
4	Region4	200	44	8800	2.3	50	8290
5	Region5	23	10.5	241.5	2.3	70	118.6
6	Region6	360	10.5	3780	2.3	70	2882
7							

Here is a classic PivotTable frame and field list created by importing the data from Access.

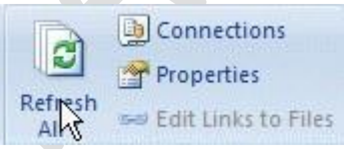


Choose fields to add to report:

☐ Cost per Item
☐ Field1
☐ ID
☐ Items Sold
☐ Overhead
☐ Price
☐ Profit
☐ Total Sales

Refreshing PivotTable Data

Once you build a PivotTable using external data, you can refresh the table by clicking the Refresh button on the Data Ribbon. This will update the PivotTable with any changes made to pertinent data in the source database.



You can also refresh the PivotTable by right clicking a cell in the table and selecting the Refresh option from the pop up menu.

Unit 5: Using 'What-If' Analysis Tools

In this unit, you will learn how to:

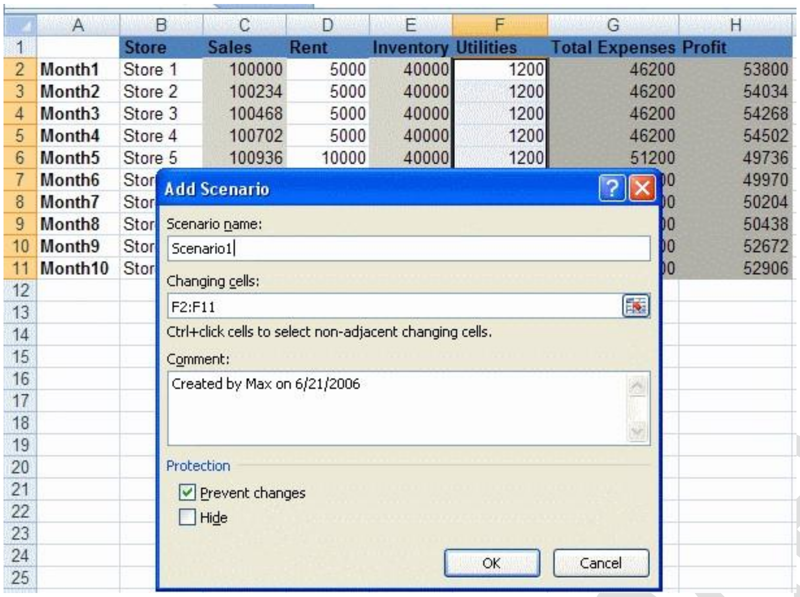
Create and view different scenarios
Use the Goal Seek utility
Use the Solver utility
Create one and two input data tables

Using Scenarios in Excel

In business, it is often beneficial to explore multiple scenarios for a given situation to see what combination of factors gives the best results. With Excel 2013 you can easily explore multiple scenarios based on the same worksheet layout.

In Excel, a scenario can be described as a set of cell values that is saved and substituted into your worksheet as required. If you have multiple scenarios saved, you can load different scenarios into your worksheet and compare and contrast them to see which one gives the best results. Since the worksheet will be fully calculated according to the given set of scenario data being used, you can compare and contrast the results of one set of data (scenario) with another by simply switching between different scenarios. The end result is that scenarios provide a great way of performing "What if?" analysis.

You might use scenarios to represent different budget options, evaluate different financial forecasts, or to compare different data projections based on a number of factors. The data that makes up your scenario is saved with your workbook, but remains hidden until you use the Scenario Manager to load the scenario data. All of the scenarios you create are just versions of the same worksheet layout, with each different scenario having different values for certain cells.



	A	B	C	D	E	F	G	H
1		Store	Sales	Rent	Inventory	Utilities	Total Expenses	Profit
2	Month1	Store 1	100000	5000	40000	1200	46200	53800
3	Month2	Store 2	100234	5000	40000	1200	46200	54034
4	Month3	Store 3	100468	5000	40000	1200	46200	54268
5	Month4	Store 4	100702	5000	40000	1200	46200	54502
6	Month5	Store 5	100936	10000	40000	1200	51200	49736
7	Month6	Stor					0	49970
8	Month7	Stor					0	50204
9	Month8	Stor					0	50438
10	Month9	Stor					0	52672
11	Month10	Stor					0	52906

Add Scenario

Scenario name: Scenario1

Changing cells: F2:F11

Ctrl+click cells to select non-adjacent changing cells.

Comment: Created by Max on 6/21/2006

Protection

☑ Prevent changes

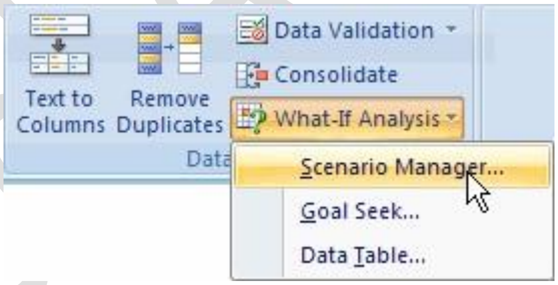
□ Hide

OK Cancel

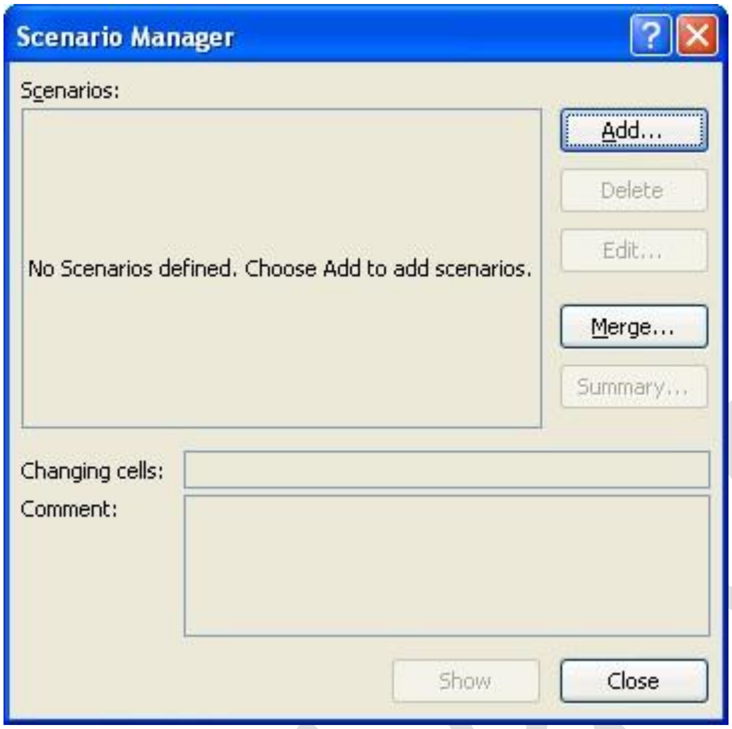
Creating a Scenario

When you are creating scenarios for a worksheet, it is a good idea to create a base scenario with the actual or current data for the worksheet and name it appropriately. This way, when you are exploring the scenarios you have created, you will always be able to return to the original (or actual) data.

To create a scenario in Excel, begin with the worksheet that you want to add scenarios to and click the Scenario Manager option from the What–If-Analysis button on the Data Ribbon.

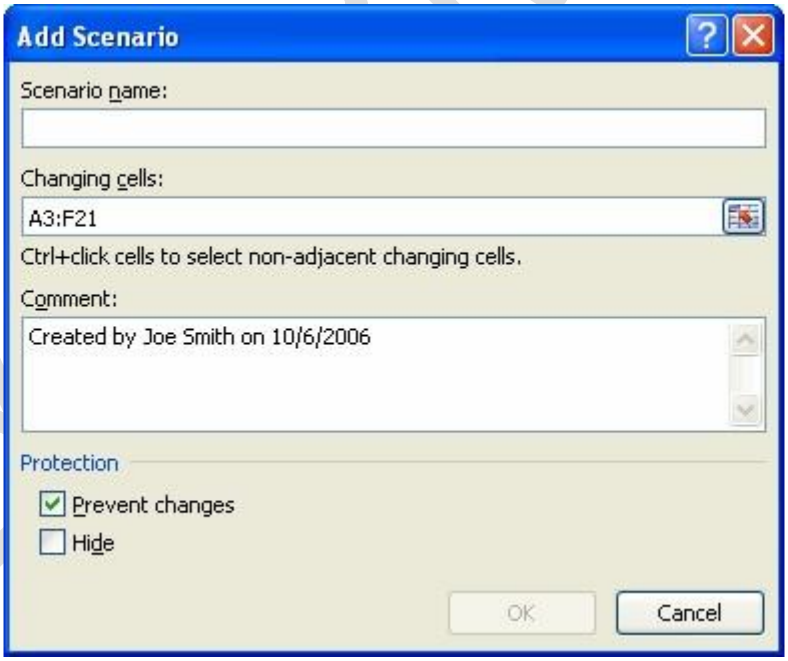


When you click on this option, the Scenario Manager dialogue will appear.



No Scenarios defined. Choose Add to add scenarios.

To create a scenario, click the Add button to display the Add Scenario dialogue box.



Changing cells: A3:F21

Ctrl+click cells to select non-adjacent changing cells.

Comment: Created by Joe Smith on 10/6/2006

Protection

☑ Prevent changes

□ Hide

The Add Scenario dialogue is used to specify a name for the given scenario and to enter the cells that will have value changes according to this scenario.

To enter a name, make sure that your cursor is in the Scenario Name text box and type a name that describes the scenario you are creating.

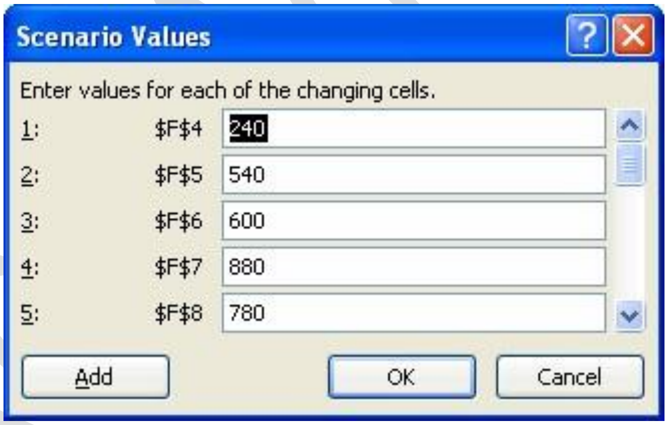
Next, decide which cells you want to change with this scenario, and then put your cursor in the Changing Cells text box. Select the cells that will be changing with your mouse, and remember to use the Ctrl button for non adjacent selections.

If the Add Scenario dialogue box gets in the way when you are selecting cells, you can drag it to one side with your mouse, or collapse it by clicking the Collapse Dialogue button.

You can also add some remarks describing the scenario in the Comment area of the dialogue box if you wish.

You will see two check boxes at the bottom of the add scenario dialogue that are labeled Prevent Changes and Hide. If you check the Prevent Changes box, changes to this scenario will not be permitted if the corresponding worksheet is protected. If you check the Hide box, the scenario will be hidden if the corresponding worksheet is protected.

Once you enter a name, select the cells to be changed, and add some comments; click the OK button to show the Scenario Values box.


Scenario Values

Enter values for each of the changing cells.

1: $F$4 240

2: $F$5 540

3: $F$6 600

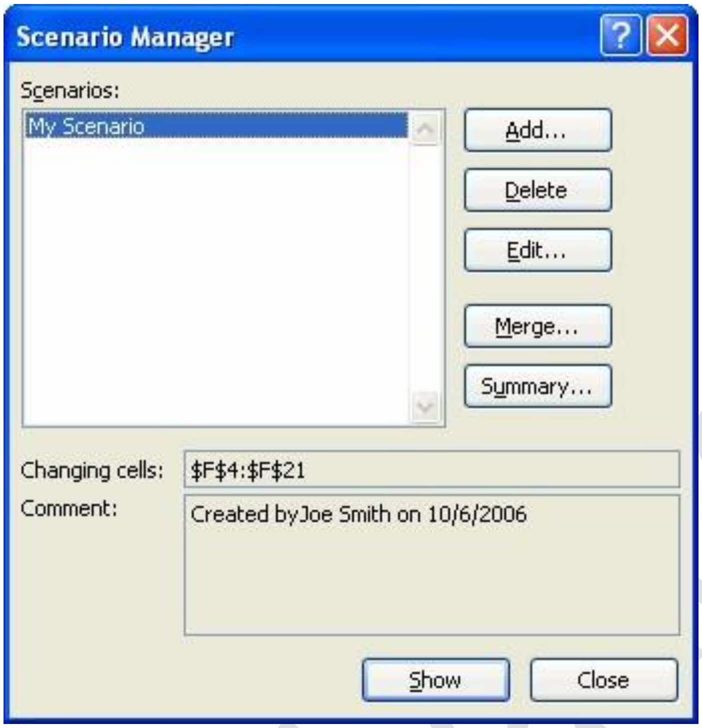
4: $F$7 880

5: $F$8 780

Here you will see a series of text fields labeled with the cell references for the corresponding changing cells. In each text field, you should enter the appropriate value for this scenario.

When you use this scenario, the values you enter here will be loaded into the corresponding worksheet. Remember, you can move between text boxes by pressing the Tab key.

Click the OK button to create the scenario, and you will then see the Scenario Manager box with your newly created scenario available in the Scenarios area.



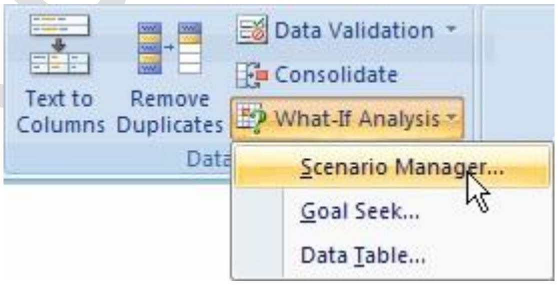
Changing cells: $F$4:$F$21

Comment: Created byJoe Smith on 10/6/2006

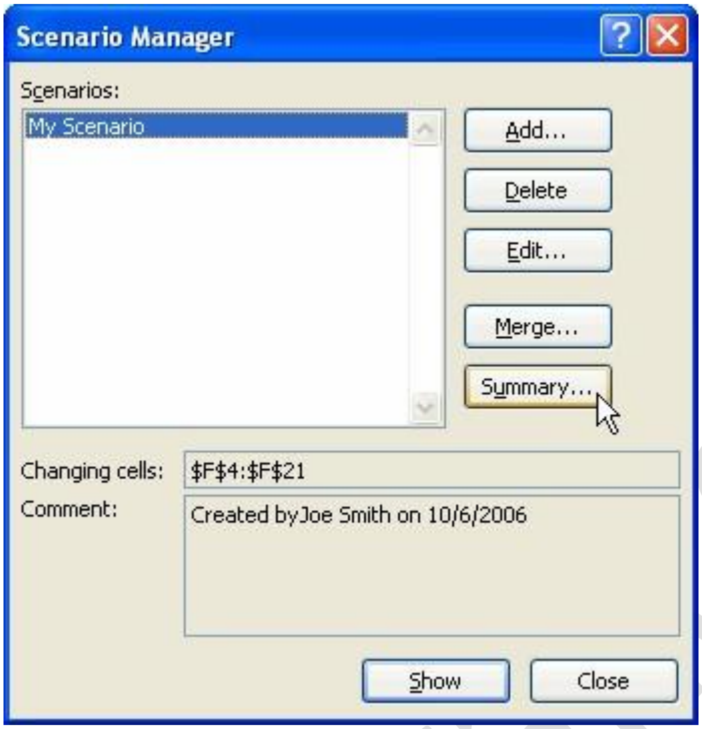
In this dialogue box, you can see the changing cells in the Changing Cells area and any comments you entered for this scenario are visible in the Comment area. If there is an assortment of scenarios in the Scenario Manager, just select one and then click the Show button to see the results of the given scenario in the spreadsheet.

Creating a Scenario Summary Report

To create a summary report for a given Scenario, click the Scenario Manager option under the What–If Analysis button on the Data ribbon.



When the Scenario Manager dialogue appears, click the Summary button.



Scenario Manager

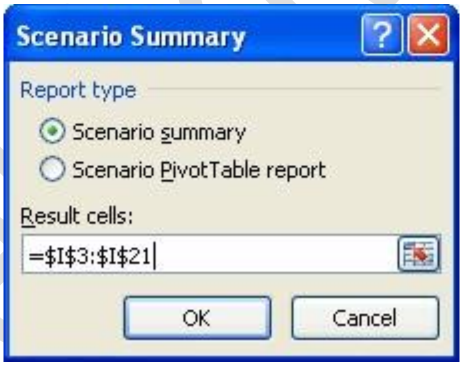
Scenarios:

My Scenario

Changing cells: $F$4:$F$21

Comment: Created byJoe Smith on 10/6/2006

This will display the Scenario Summary box. Here you can choose to create a PivotTable (based on the scenario data) or a scenario summary by selecting the appropriate radio button.



Scenario Summary

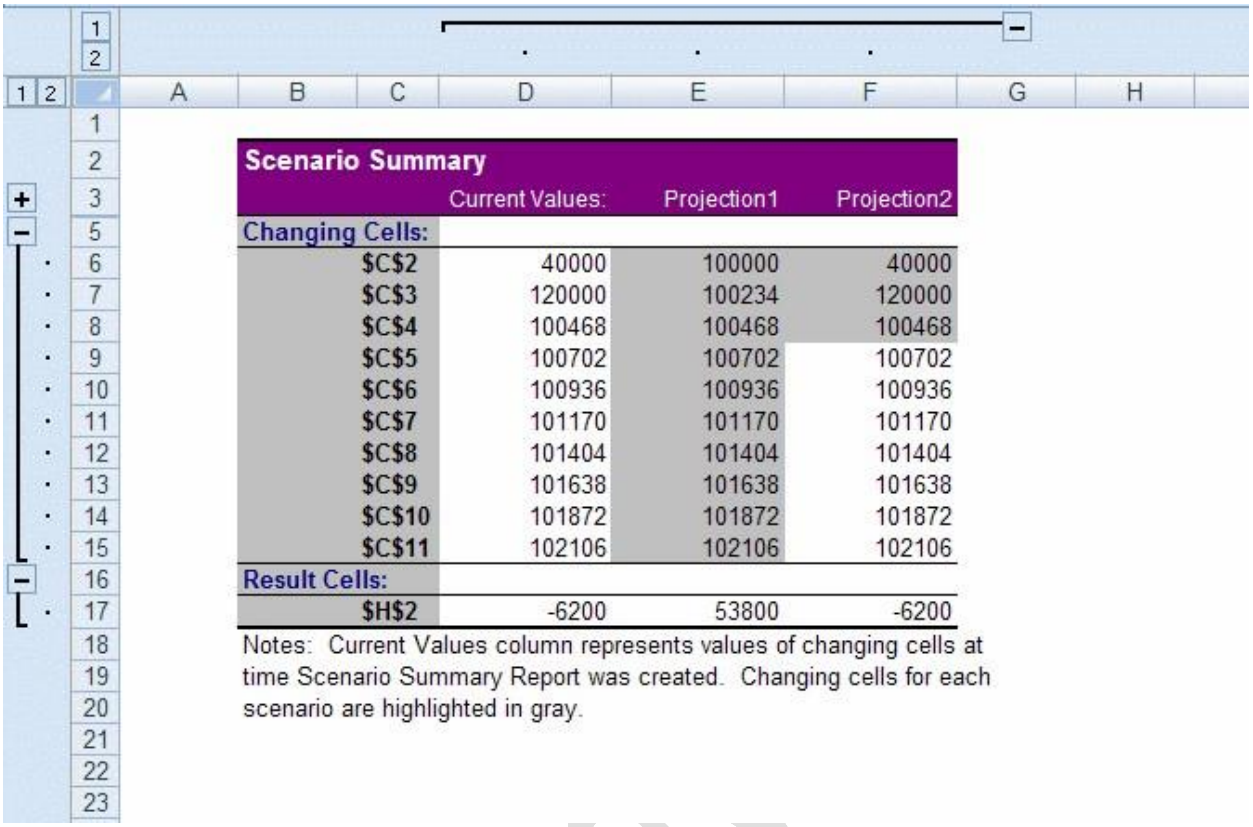
Report type

● Scenario summary

○ Scenario PivotTable report

Result cells: =$I$3:$I$21

In this example, the Scenario Summary button is selected. The next step is to select the result cells that you want to be shown in the summary report. Finally, click OK to create the summary.



Scenario Summary

	Current Values:	Projection1	Projection2
Changing Cells:			
$C$2	40000	100000	40000
$C$3	120000	100234	120000
$C$4	100468	100468	100468
$C$5	100702	100702	100702
$C$6	100936	100936	100936
$C$7	101170	101170	101170
$C$8	101404	101404	101404
$C$9	101638	101638	101638
$C$10	101872	101872	101872
$C$11	102106	102106	102106
Result Cells:			
$H$2	-6200	53800	-6200

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.

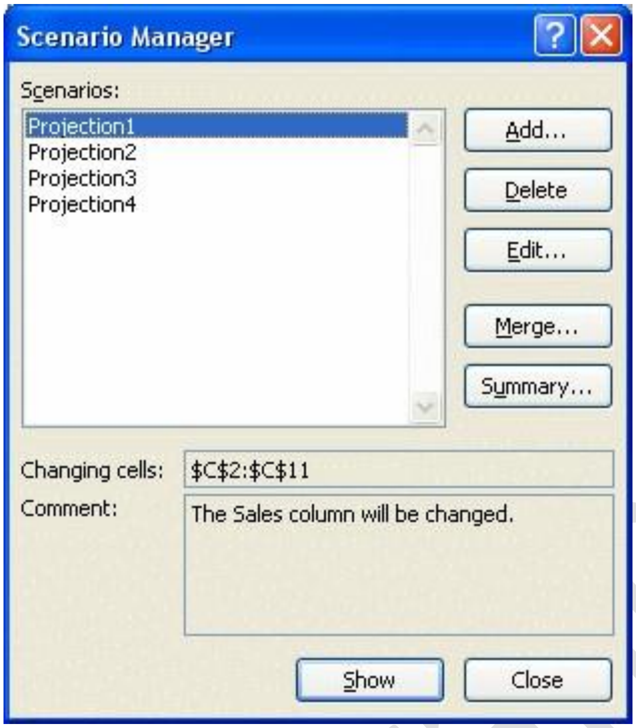
The summary report will be created and inserted into a new worksheet. The report contains a list of cell references corresponding to the changing cells in your various scenarios. It also contains a column of current values for the changing cells, as well as columns of changed values corresponding to each of the available scenarios (Current values, Projection1, Projection2).

There is also a row of scenario names across the top of the columns to identify which scenario a given column of values belongs to. (The scenarios in the report shown above are: Current values, Projection1, and Projection2.) At the bottom of the report, you can see the values for the results cells that correspond to each scenario.

Saving Multiple Scenarios

To save multiple scenarios for a worksheet, choose the Scenario Manager option from the What If Analysis button to display the Scenario Manager dialogue.

Use the Add button in the Scenario Manager to create as many scenarios as you require. All of the scenarios you create for a given worksheet will be available in the scenarios list in the Scenario Manager dialogue.



Scenarios:
Projection1
Projection2
Projection3
Projection4

Changing cells: $C$2:$C$11

Comment: The Sales column will be changed.

In this example, you can see four scenarios in the scenario list. These scenarios will be saved with this workbook when the workbook itself is saved. When you open the workbook, you can see the available scenarios by displaying the Scenario Manager.

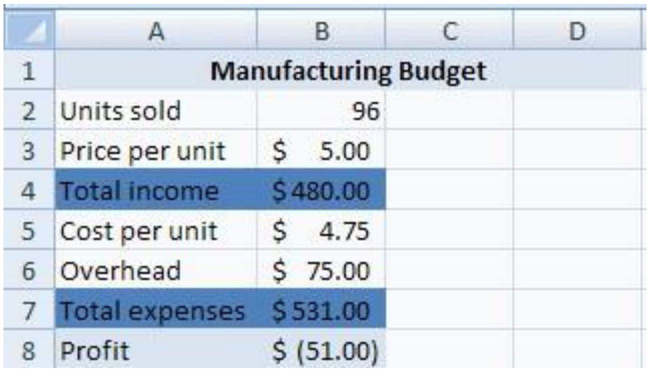
To load a scenario into the worksheet, select it from the Scenarios list and click the Show button. To remove a scenario, select it and click Delete. To make changes to a scenario, select it and click the Edit button.

You can have a different set of scenarios for each worksheet in your workbook. When you display the Scenario Manager, only scenarios for the currently active worksheet will be shown in the Scenarios list.

Using Goal Seek

Goal Seek is another useful what-if analysis tool provided by Excel. With Goal Seek, Excel will find a value for a specified cell that makes a given worksheet formula equal to a value that you define. In other words, you can set a formula to a value (goal) that you would like to attain, and then specify one of the cells that the formula references as a cell that Excel can adjust in order to reach the goal.

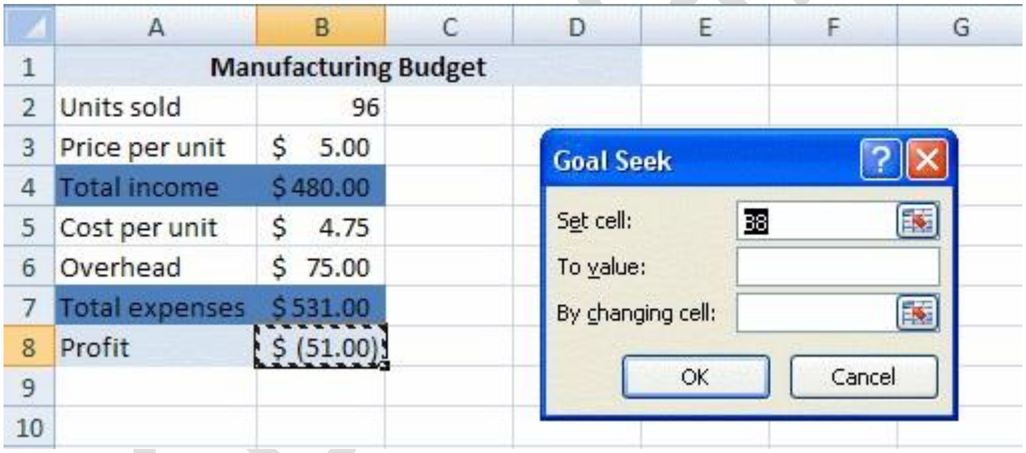
Take the following worksheet as an example.



	A	B	C	D
1	Manufacturing Budget			
2	Units sold	96		
3	Price per unit	$ 5.00		
4	Total income	$ 480.00		
5	Cost per unit	$ 4.75		
6	Overhead	$ 75.00		
7	Total expenses	$ 531.00		
8	Profit	$ (51.00)		

The worksheet clearly shows a negative profit (loss) for the current worksheet values. We can use Goal Seek to find a break even point based on changes to a cell that is referenced (directly or indirectly) by the profit formula in cell B8. The profit formula is Total Income - Total Expenses, or B4 - B7. Cell B2 contains the number of units sold. This will be the cell that we will adjust to break even.

First, select cell B8 (the cell with the profit formula), and then invoke Goal Seek by choosing the Goal Seek option from the What-If Analysis button on the Data Ribbon.



	A	B	C	D	E	F	G
1	Manufacturing Budget						
2	Units sold	96					
3	Price per unit	$ 5.00					
4	Total income	$ 480.00					
5	Cost per unit	$ 4.75					
6	Overhead	$ 75.00					
7	Total expenses	$ 531.00					
8	Profit	$ (51.00)					
9							
10							

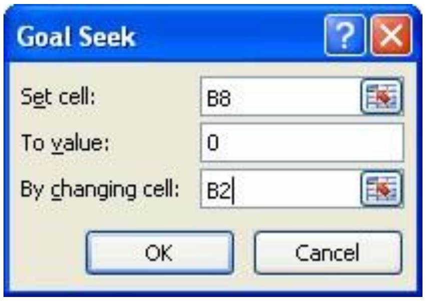
Goal Seek

Set cell: B8

To value:

By changing cell:

Cell B8 is entered into the Set Cell text field, because that is the cell that you selected just before invoking Goal Seek. In the To Value text field type 0 and in the By Changing Cell field enter B2 (by typing it in directly or by clicking with your mouse). The resulting Goal Seek box should look like the following.



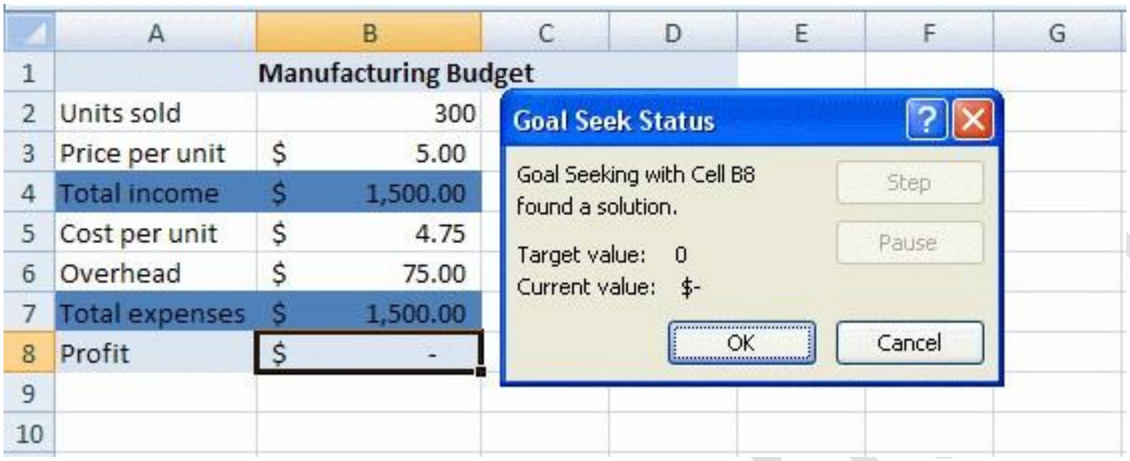
Goal Seek

Set cell: B8

To value: 0

By changing cell: B2

If you click the OK button, Goal Seek will find a value for Cell B2 (Units sold) that will make the profit equal to 0 (break even).



	A	B
1	Manufacturing Budget	
2	Units sold	300
3	Price per unit	$ 5.00
4	Total income	$ 1,500.00
5	Cost per unit	$ 4.75
6	Overhead	$ 75.00
7	Total expenses	$ 1,500.00
8	Profit	$ -
9		
10		

Goal Seek Status

Goal Seeking with Cell B8 found a solution.

Target value: 0

Current value: $-

The Goal Seek Status box reports that a solution has been found. You can see the value 300 in cell B2. This means that if all other variables remain unchanged, you must sell 300 units to break even. Clicking the Cancel button will restore the original worksheet values, and clicking OK will enter the Goal Seek solution values into the worksheet.

You can just as easily use Goal Seek to find the price per unit or the cost per unit that would make the worksheet break even.

Using Solver

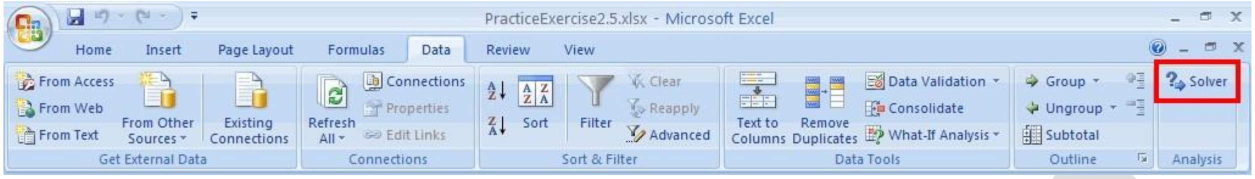
Sometimes, when dealing with more complex problems, Excel data tables or the Goal Seek feature cannot provide the kind of forecast or analysis you are looking for. In this type of situation, Excel 2013's Solver feature might be able to help.

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.

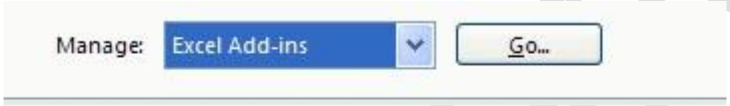
Checking Solver Installation

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.



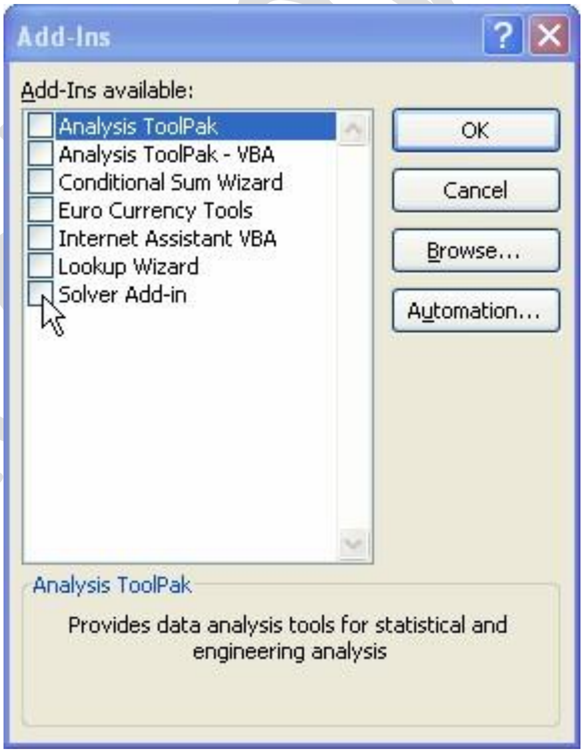
If there is no Solver option, the Solver has not yet been installed. To install the Solver, display the Office menu and click the Excel Options button at the bottom.

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.



Manage: Excel Add-ins

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



Add-Ins

Add-Ins available:

- □ Analysis ToolPak
- □ Analysis ToolPak - VBA
- □ Conditional Sum Wizard
- □ Euro Currency Tools
- □ Internet Assistant VBA
- □ Lookup Wizard
- □ Solver Add-in

Analysis ToolPak

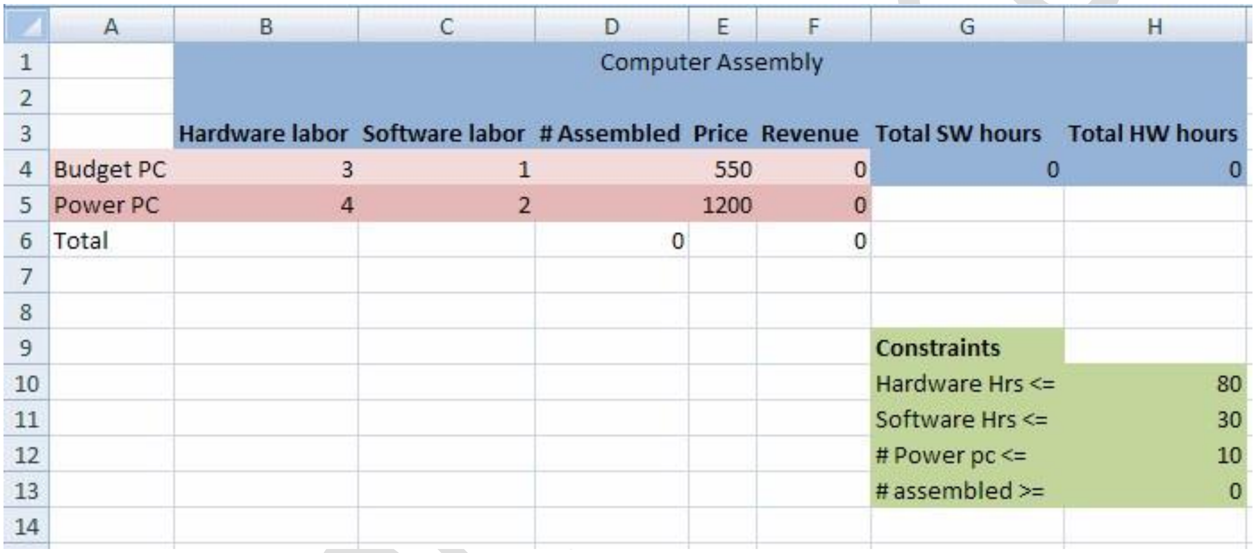
Provides data analysis tools for statistical and engineering analysis

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.



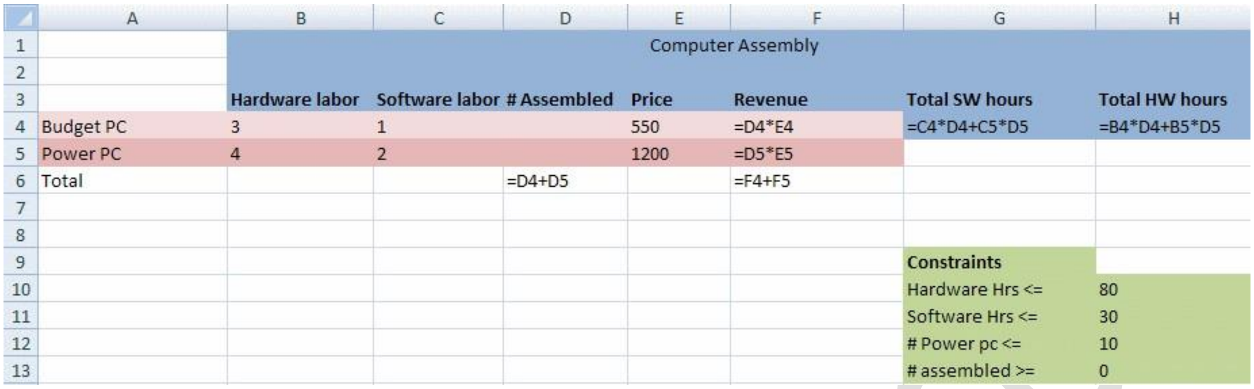
	A	B	C	D	E	F	G	H
1		Computer Assembly						
2								
3		**Hardware labor**	**Software labor**	**# Assembled**	**Price**	**Revenue**	**Total SW hours**	**Total HW hours**
4	Budget PC	3	1		550	0	0	0
5	Power PC	4	2		1200	0		
6	Total			0		0		
7								
8								
9							**Constraints**	
10							Hardware 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.

The following worksheet is the same as the previous one, except that the cell formulas are now visible. Take your time and examine the cell references in the formulas carefully.



	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.

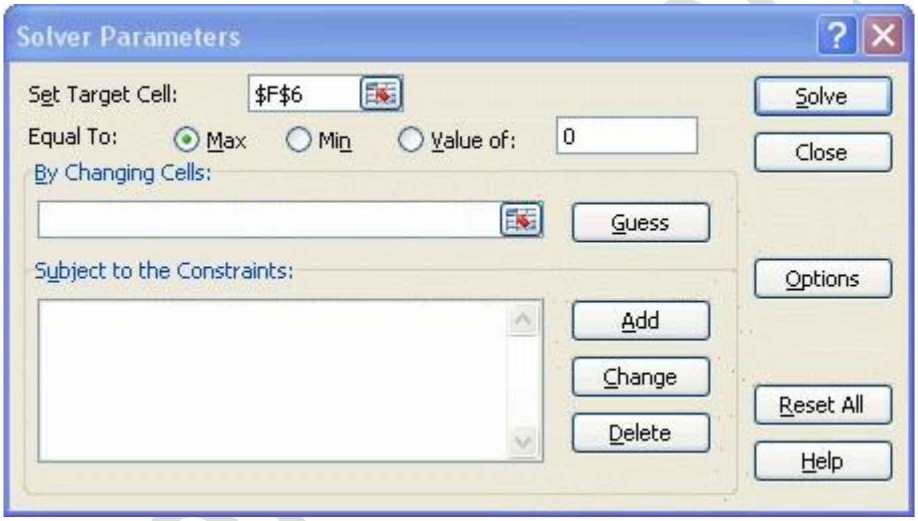
To summarize, in order to use Excel's Solver you must have a formula to optimise (called the objective formula) and you must have cells that can be changed to optimise the objective. The cells to be changed should be precedents to the objective formula; that is, the calculation of the objective formula 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.



Solver Parameters

Set Target Cell: $F$6

Equal To: ● Max ○ Min ○ Value of: 0

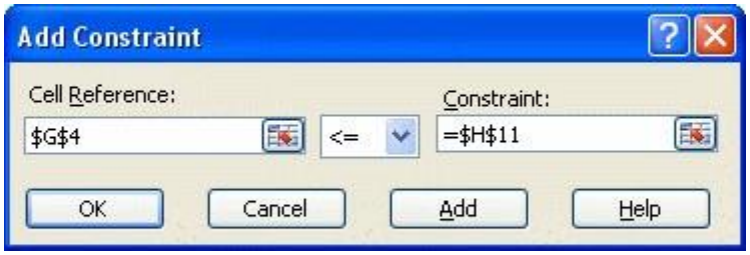
By Changing Cells:

Subject to the Constraints:

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.



Add Constraint

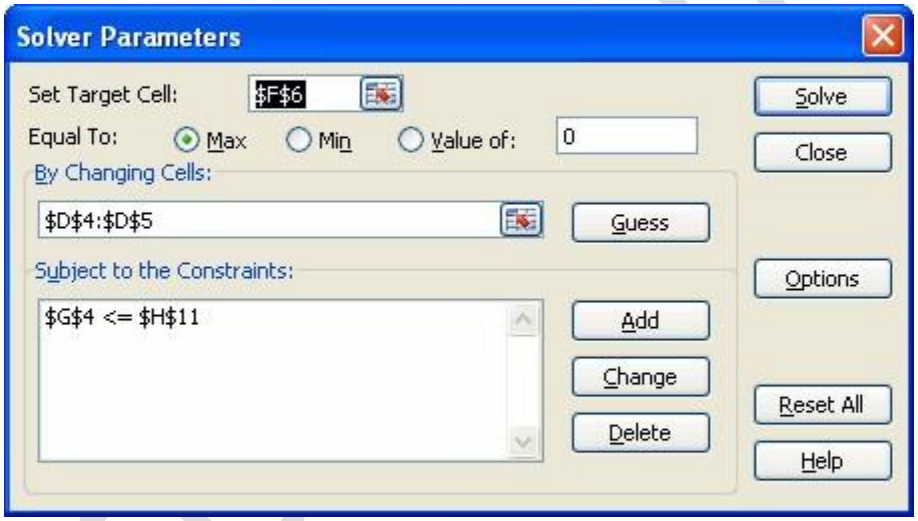
Cell Reference: $G$4

<=

Constraint: =$H$11

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, which now looks like this.



Solver Parameters

Set Target Cell: $F$6

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells: $D$4:$D$5

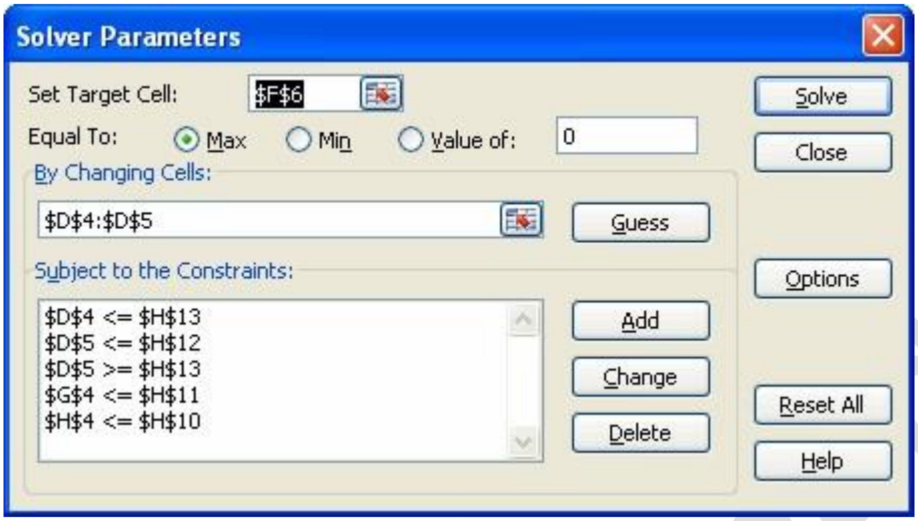
Subject to the Constraints:

$G$4 <= $H$11

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.



Solver Parameters

Set Target Cell: $F$6

Equal To: ● Max ○ Min ○ Value of: 0

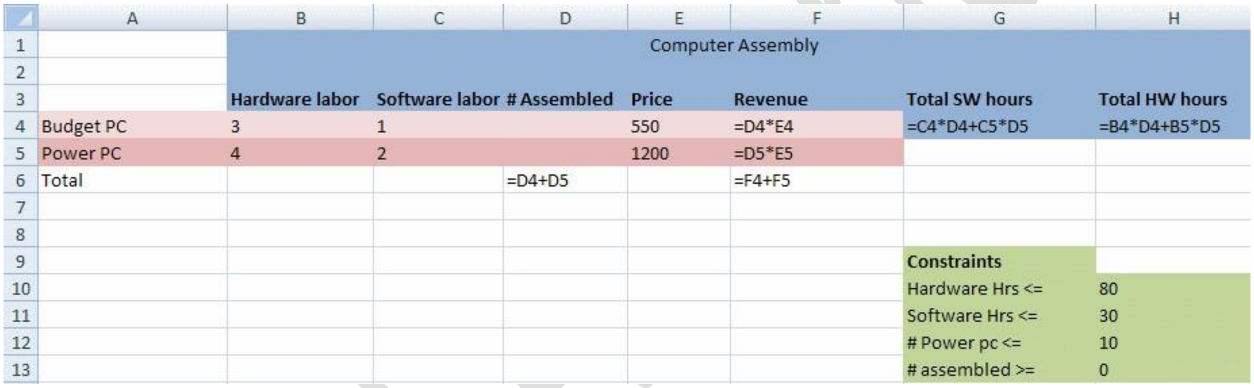
By Changing Cells:

$D$4:$D$5

Subject to the Constraints:

```
$D$4 <= $H$13
$D$5 <= $H$12
$D$5 >= $H$13
$G$4 <= $H$11
$H$4 <= $H$10
```

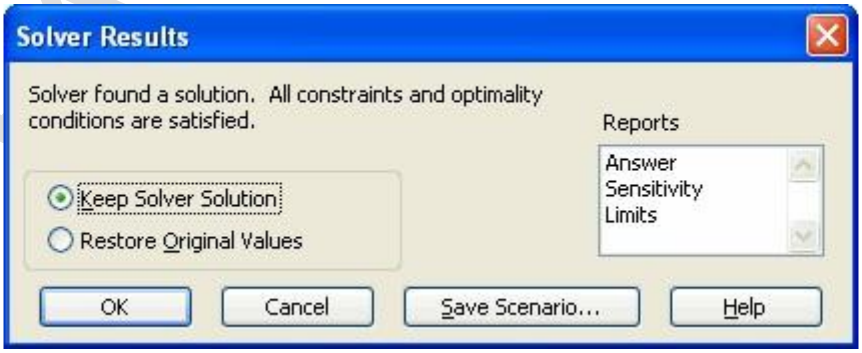
Here is the corresponding worksheet with formulas shown.



	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.



Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

● Keep Solver Solution
○ Restore Original Values

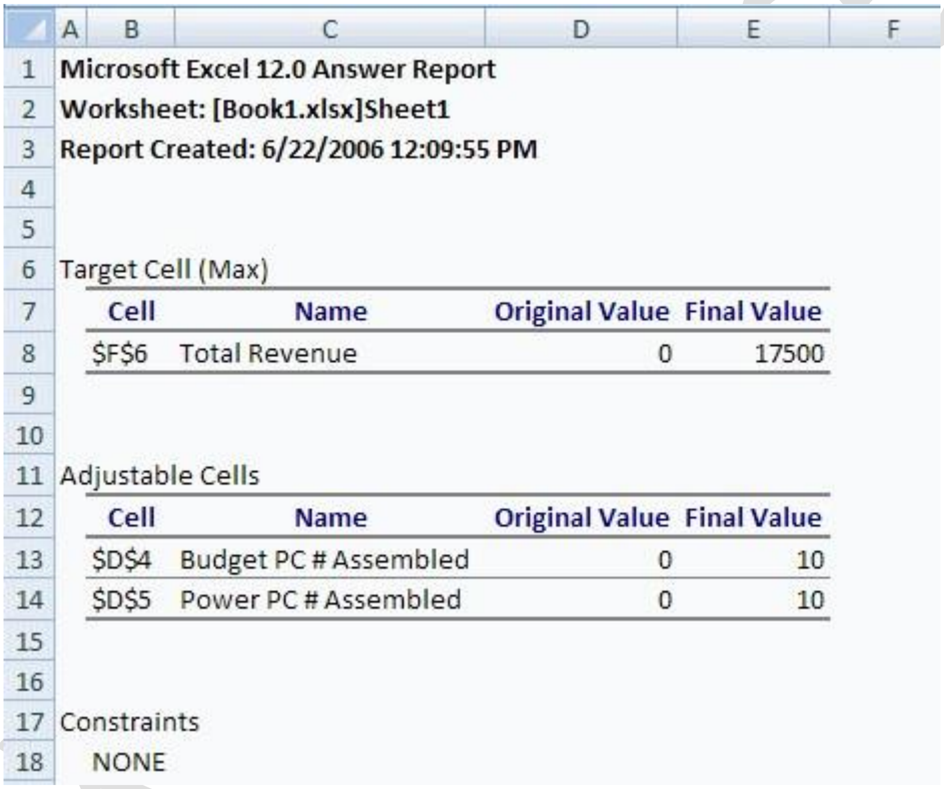
Reports

Answer
Sensitivity
Limits

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
$F$6	Total Revenue	0	17500

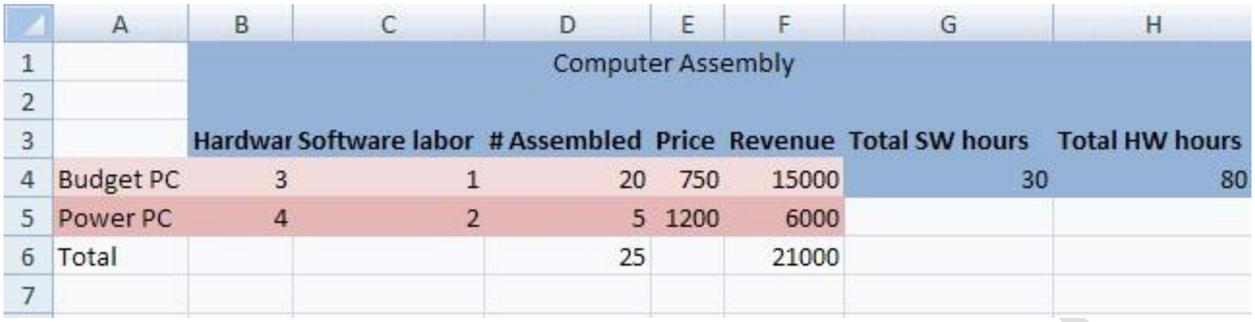
Adjustable Cells

Cell	Name	Original Value	Final Value
$D$4	Budget PC # Assembled	0	10
$D$5	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								

Using One and Two Input Data Tables

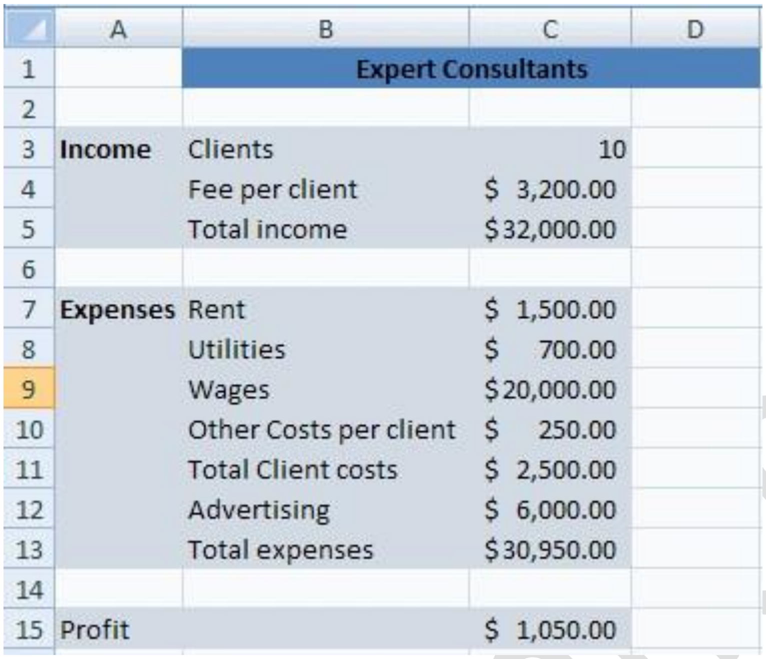
You can use Excel data tables to see how your formula results change when the data that the formula is based on changes. You do this by specifying a series of hypothetical values for Excel to evaluate the formulas with and then view the results of the evaluations. For example, you could examine how changes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

The following example involves a hypothetical consulting firm. Our first goal is to examine the effect of the number of clients for the firm, on the total profit, total expenses, and total income.



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

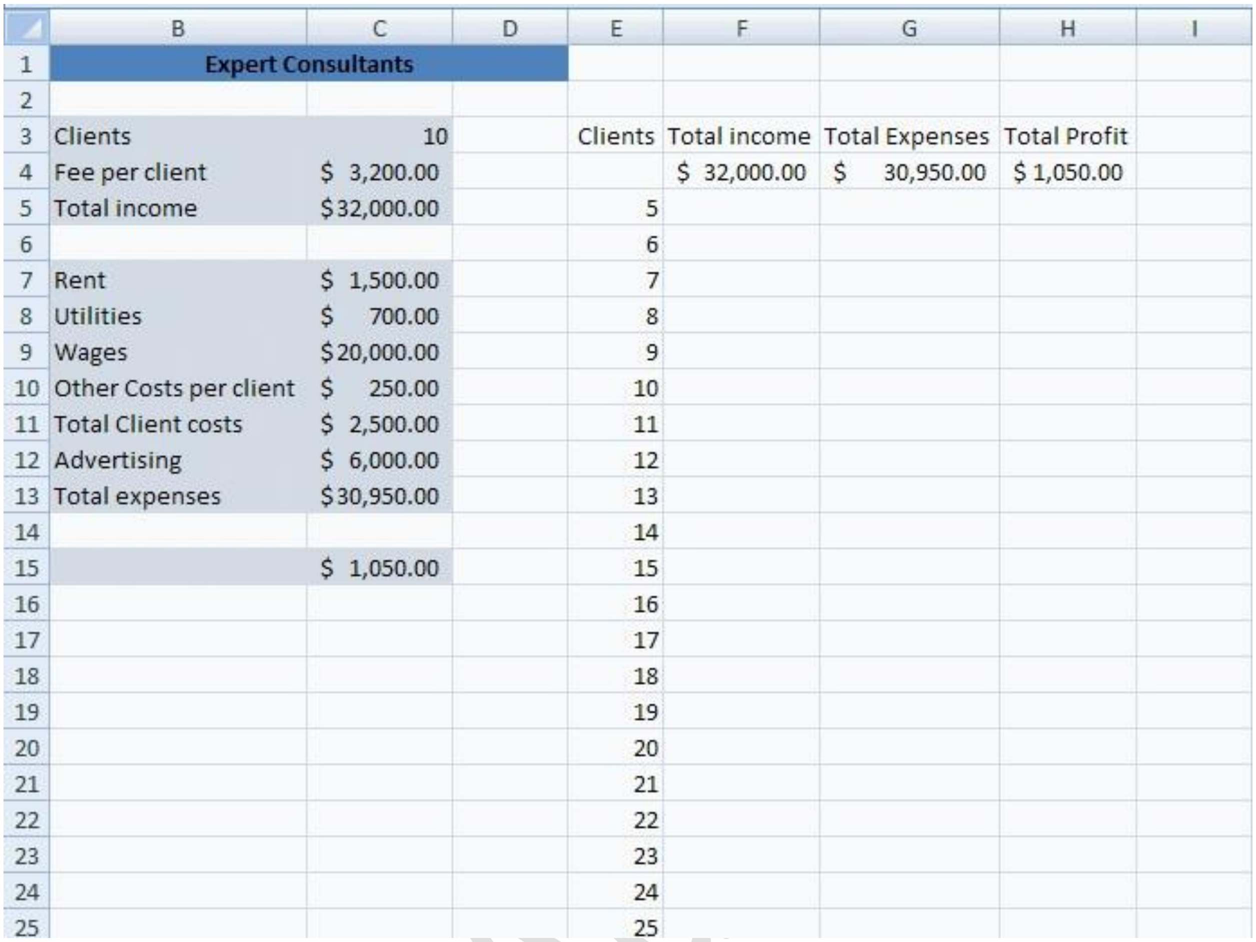
Currently, the firm has 10 clients. The values for wages, total client costs, total expenses, and profit are all calculated by formulas dependent on the number of clients the firm can retain. To see what results the profit formula and other formulas would produce for a range of hypothetical client numbers, 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.

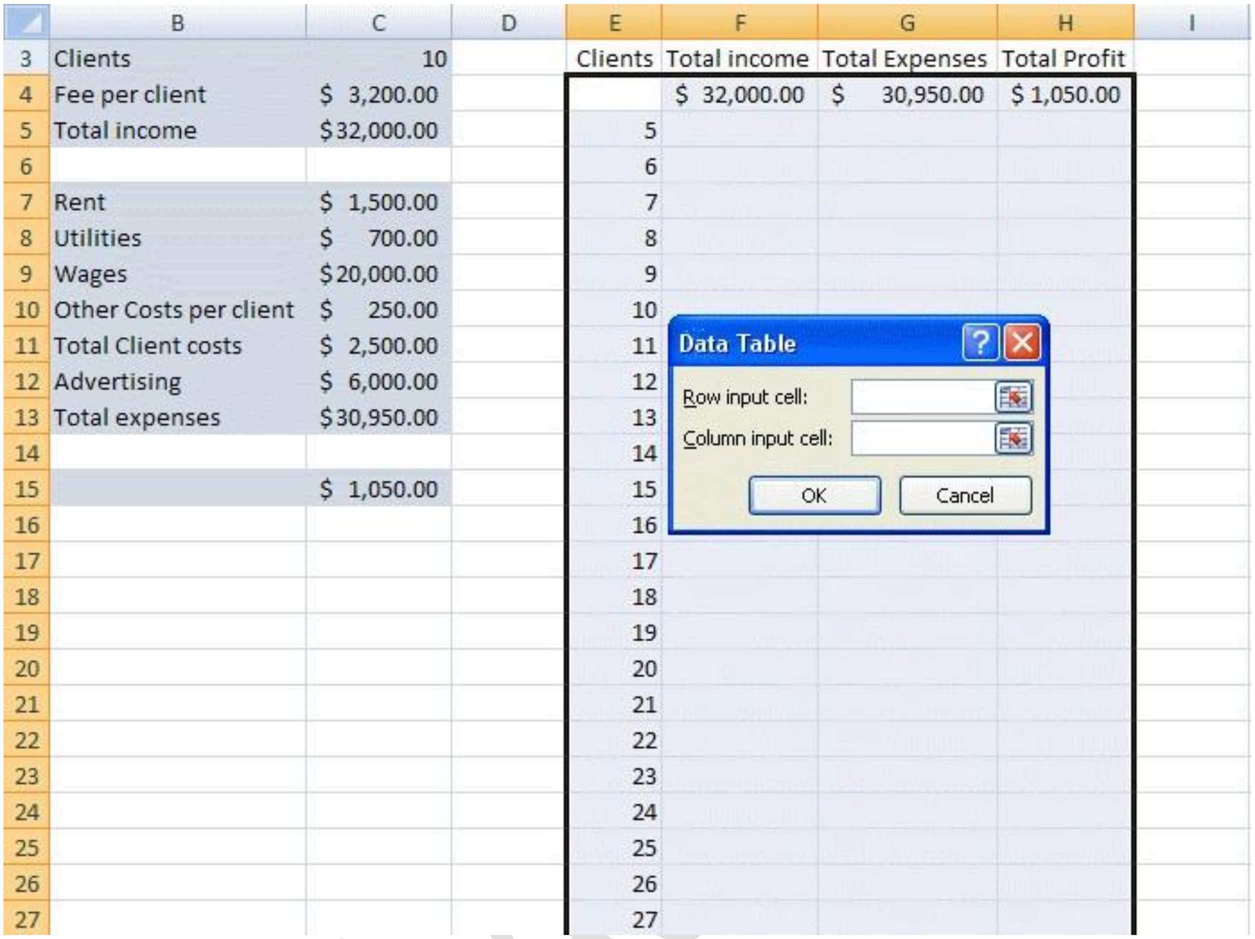


	B	C	D	E	F	G	H	I
1	Expert Consultants							
2								
3	Clients	10		Clients	Total income	Total Expenses	Total Profit	
4	Fee per client	$ 3,200.00			$ 32,000.00	$ 30,950.00	$ 1,050.00	
5	Total income	$32,000.00		5				
6				6				
7	Rent	$ 1,500.00		7				
8	Utilities	$ 700.00		8				
9	Wages	$20,000.00		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.



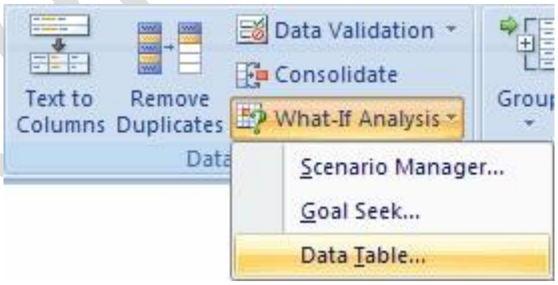
	B	C	D	E	F	G	H	I
3	Clients	10		Clients	Total income	Total Expenses	Total Profit	
4	Fee per client	$ 3,200.00			$ 32,000.00	$ 30,950.00	$ 1,050.00	
5	Total income	$32,000.00		5				
6				6				
7	Rent	$ 1,500.00		7				
8	Utilities	$ 700.00		8				
9	Wages	$20,000.00		9				
10	Other Costs per client	$ 250.00		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				
26				26				
27				27				

Data Table

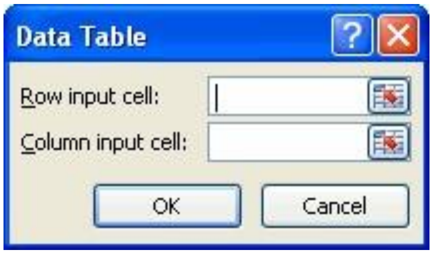
Row input cell:

Column input cell:

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

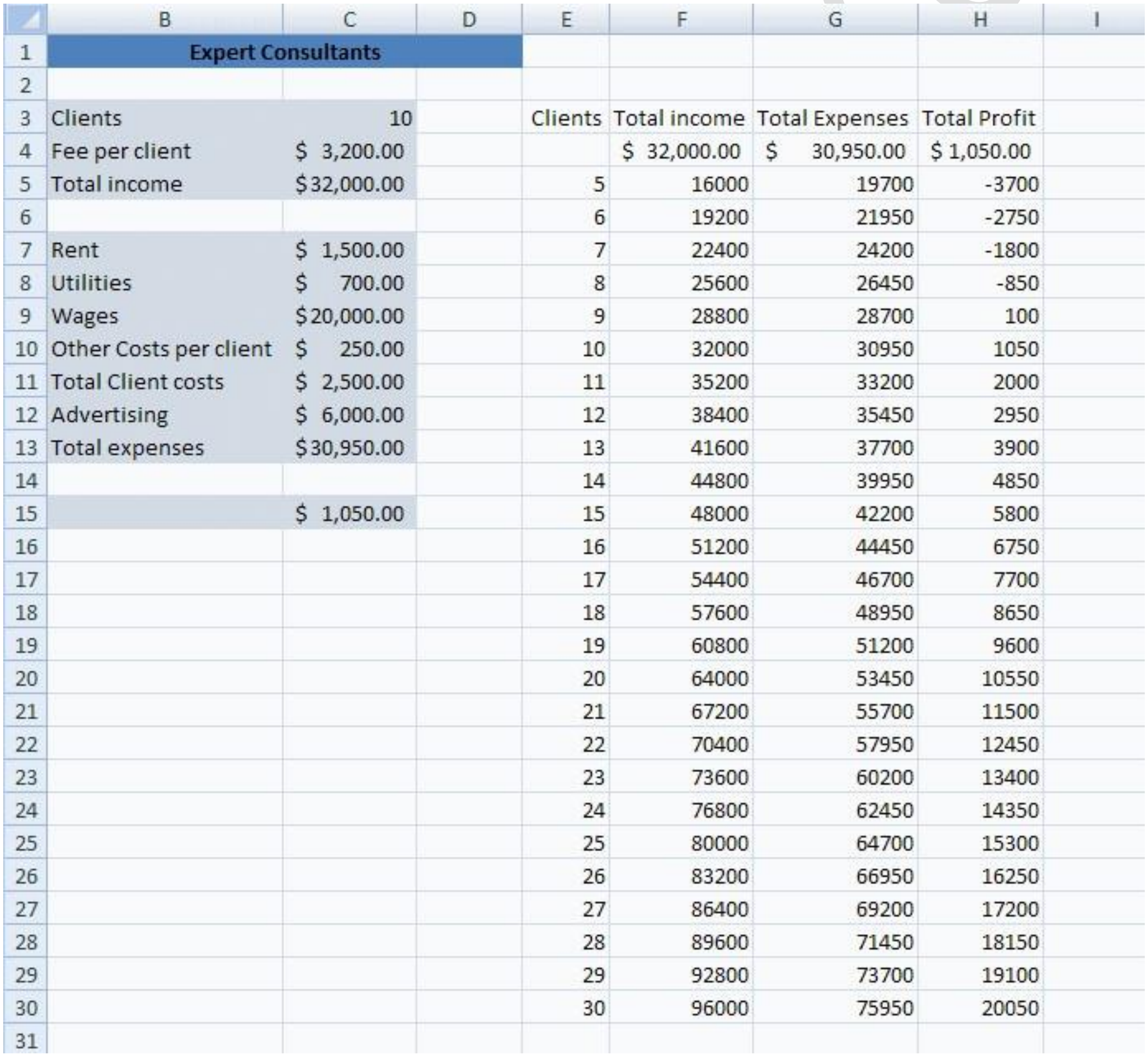


The range of input variables and formula references has been selected, and the Data Table dialogue box is ready for input.



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.)

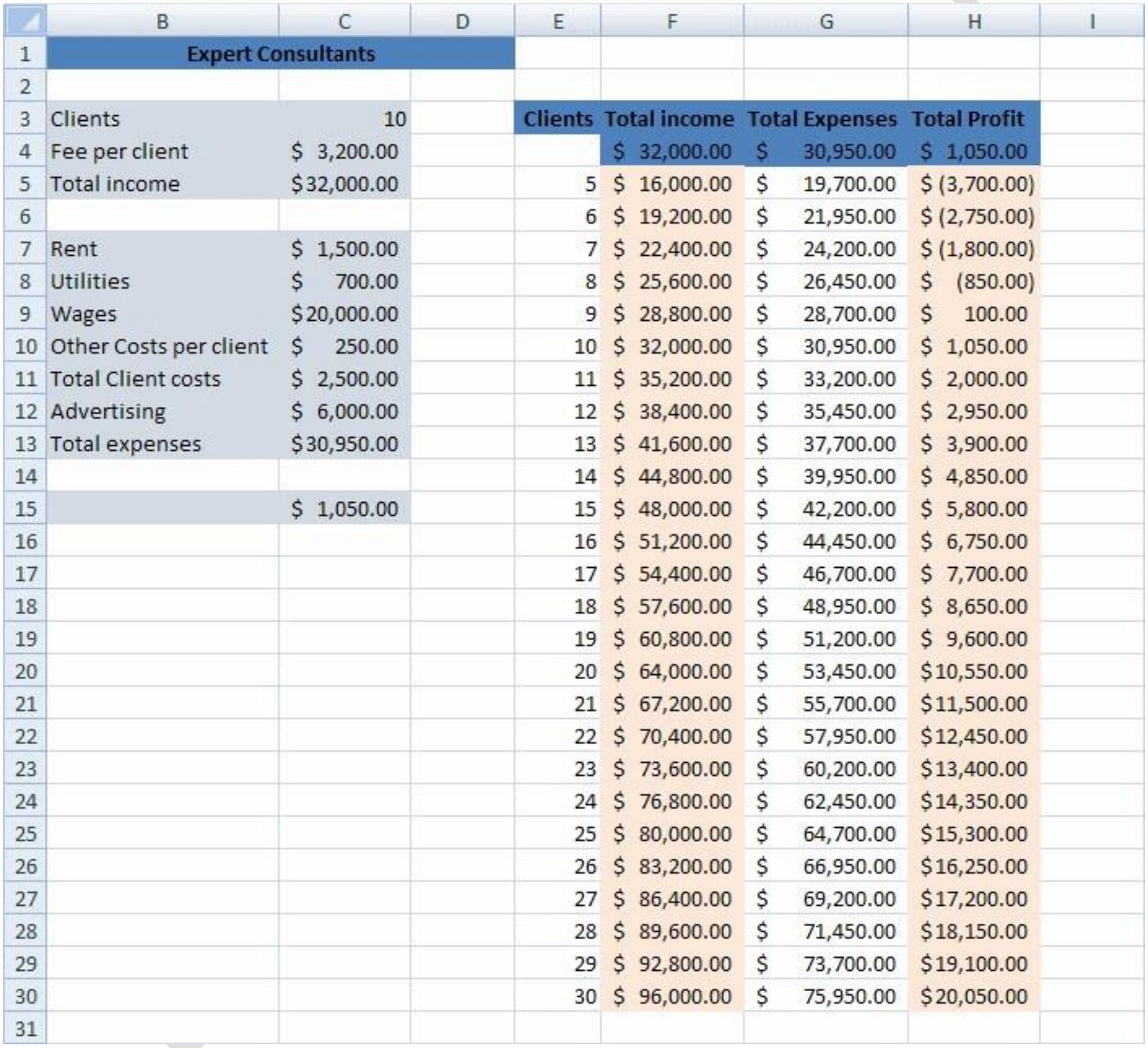
Clicking the OK button in the Table dialogue box will complete the data table.



	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.

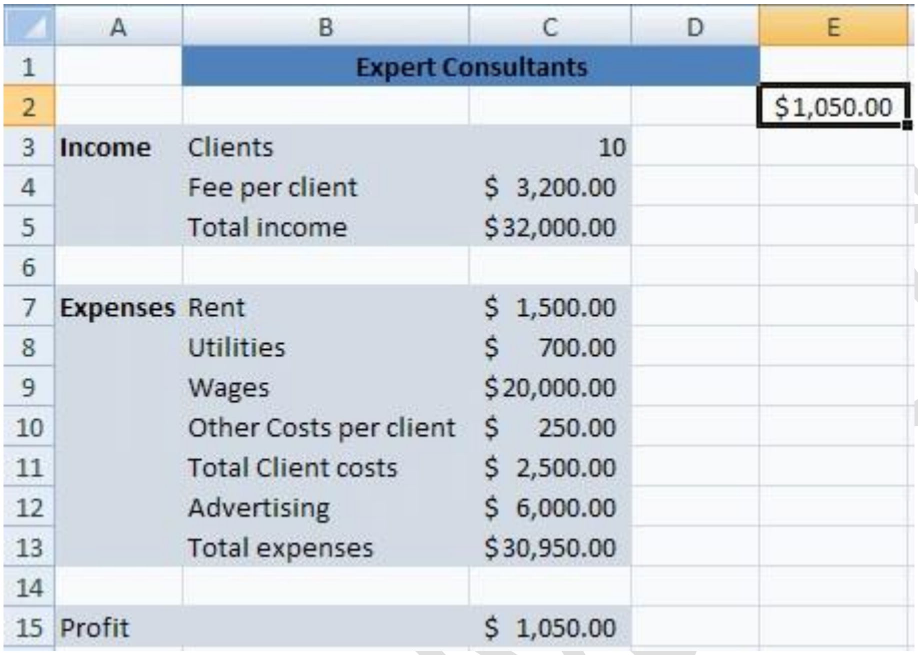


	B	C	D	E	F	G	H	I
1	Expert Consultants							
2								
3	Clients	10		Clients	Total income	Total Expenses	Total Profit	
4	Fee per client	$ 3,200.00			$ 32,000.00	$ 30,950.00	$ 1,050.00	
5	Total income	$32,000.00		5	$ 16,000.00	$ 19,700.00	$ (3,700.00)	
6				6	$ 19,200.00	$ 21,950.00	$ (2,750.00)	
7	Rent	$ 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.

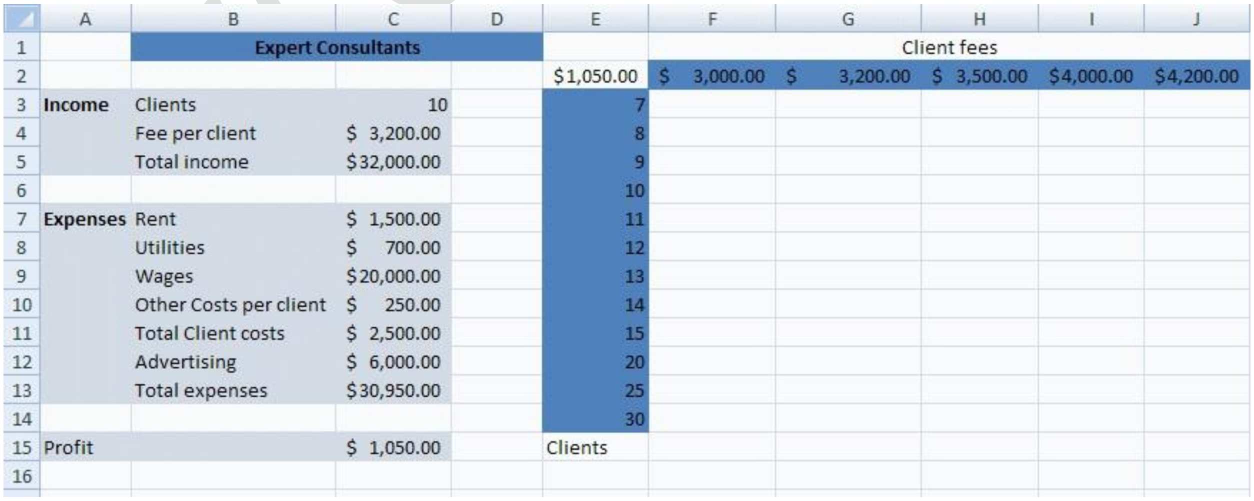
To build a two input data table, first choose an empty cell and enter a reference to the formula you want to examine. Using the same consulting firm example as before, we will choose cell E2 and enter the reference =C15 for the Profit formula.



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

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:

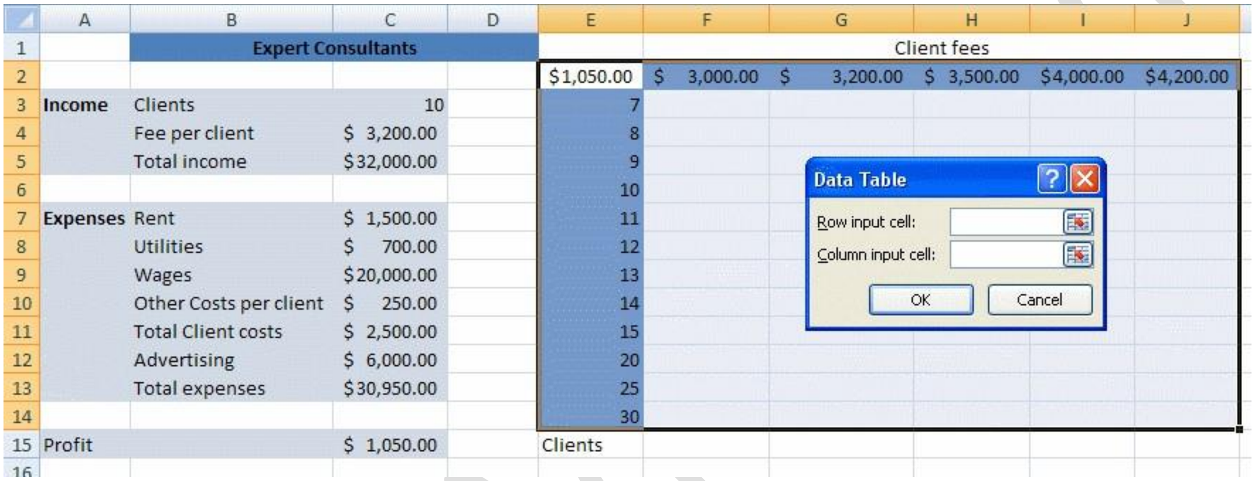


	A	B	C	D	E	F	G	H	I	J
1		Expert Consultants				Client fees				
2					$1,050.00	$ 3,000.00	$ 3,200.00	$ 3,500.00	$4,000.00	$4,200.00
3	Income	Clients	10		7					
4		Fee per client	$ 3,200.00		8					
5		Total income	$32,000.00		9					
6					10					
7	Expenses	Rent	$ 1,500.00		11					
8		Utilities	$ 700.00		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.

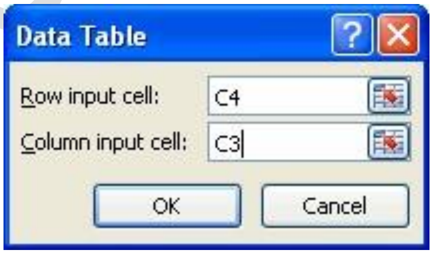
First we will select the range of data cells E2:J14 and then we will choose the Data Table option from the What If Analysis button.



	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					

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.

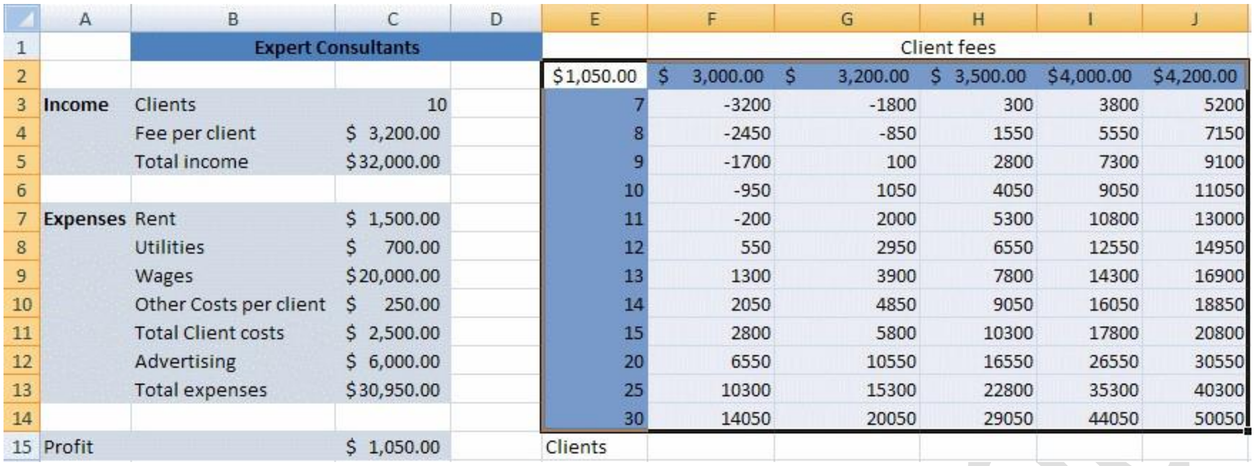


Data Table

Row input cell: C4

Column input cell: C3

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.

Unit 6: Macros

In this unit, you will learn how to:

Record and run a macro
Record a relative reference macro
Assign shortcut keys to run a macro
Copy a macro from a workbook or template

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.

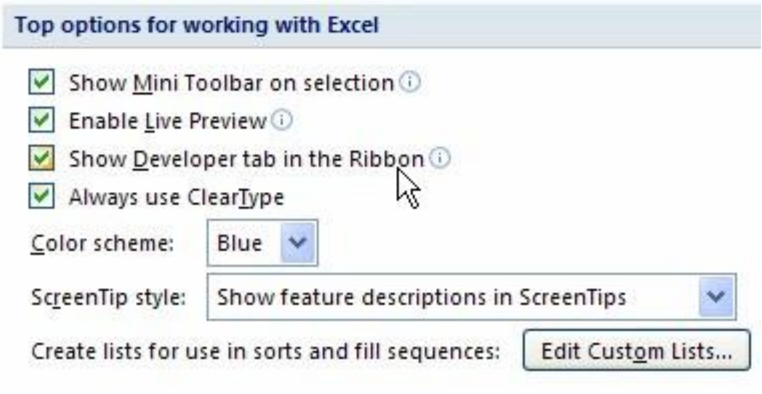
Simple macros are great for any long sequence of keystrokes that you find yourself repeating often, or for combinations of basic Excel actions that would be convenient to automate for a particular workbook. Macros can be saved with the workbook in which they were created, or they can be saved in a separate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

Creating a macro

As mentioned preciously, Excel will record the actions you take when creating a macro and convert them into Visual Basic code. This code contains all of the instructions that Excel needs to execute your macro.

The Developer Ribbon

It is not difficult to create a macro in Excel 2013, but first, you should make sure that the Developer tab is accessible. To do this, click the Excel Options button on the Office menu and then click the Popular option on the left of the Excel Options window. Under the Top Options for working with Excel option, you will see a checkbox next to the words Show Developer Tab in the Ribbon.



Top options for working with Excel

☑ Show Mini Toolbar on selection

☑ Enable Live Preview

☑ Show Developer tab in the Ribbon

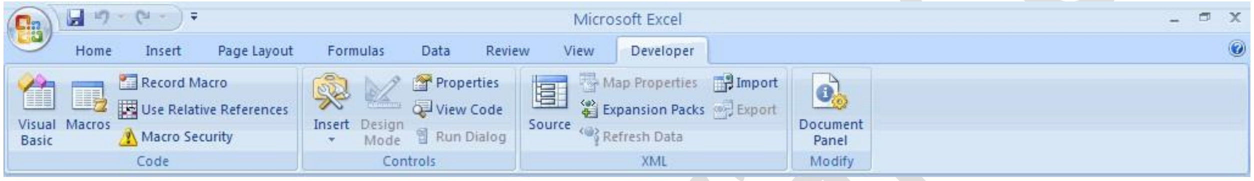
☑ Always use ClearType

Color scheme: Blue

ScreenTip style: Show feature descriptions in ScreenTips

Create lists for use in sorts and fill sequences: Edit Custom Lists...

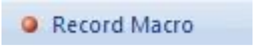
If you check this box, and click OK in the options window, the Developer tab will appear at the top of the Excel screen. If you click the Developer tab, you will see the Developer Ribbon.



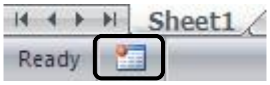
Recording a macro

It is a good idea to plan out or even practice the sequence of actions for your macro before you record it. This will not only ensure that the actions lead to the expected outcome, but also help you to avoid mistakes and unwanted actions in your macros. This is especially true for long macros that require several actions.

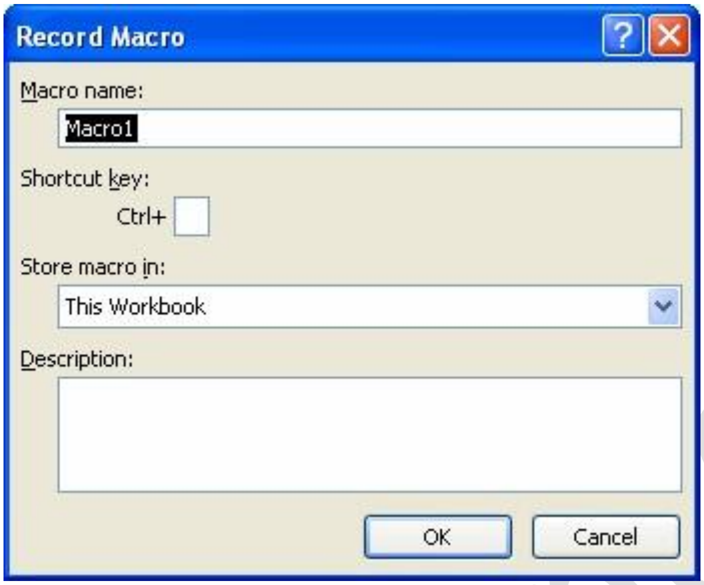
To record a macro, select the Developer ribbon, and click the Record Macro button in the Code group.



You can also click the Record Macro button on the Status bar, at the bottom left of the Excel screen.



This will display the Record Macro dialogue box.



Macro name: Macro1

Shortcut key: Ctrl+

Store macro in: This Workbook

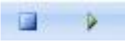
Description:

Here you can give your new macro a name (do not use spaces) and choose the location where it should be stored. Your options are to store it with this workbook, another workbook, or a special macro workbook.

This special macro workbook is called the personal macro workbook. A macro that is stored here can be run in any Excel workbook on the same computer. If you are creating a macro that you think you would like to use in multiple workbooks you should consider saving it in the personal macro workbook.

You can also enter a brief description of the macro you are about to create in the bottom text area (labeled Description). By default, the macro will be called Macro1 (if this is your first macro) and it will be saved in the current workbook.

If you click OK, you will see two buttons appear in the lower left corner of the Excel screen in the status bar.



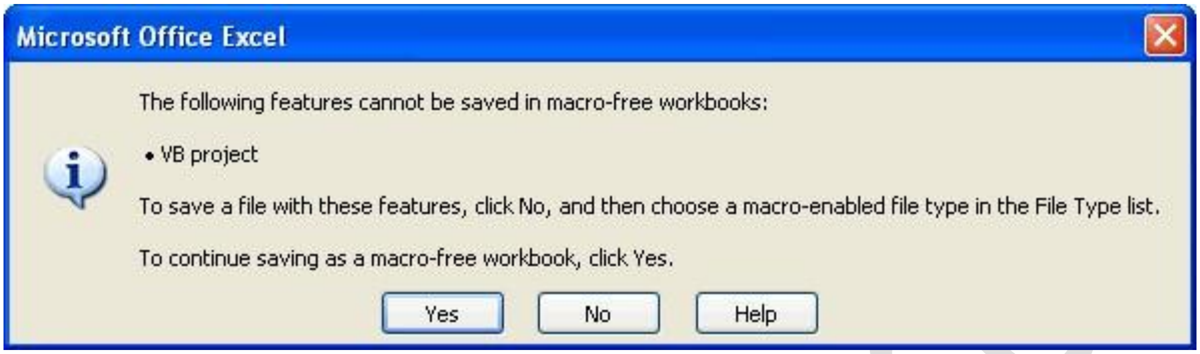
The small blue square is the Stop button, which will stop the macro from recording. The small green triangle is a Play button, which is used to play a recorded macro.

When you see these buttons, every keystroke or Excel action that you take with the mouse will be translated into VBA (Visual Basic for Applications) code by Excel. This code is what makes up your macro.

When you are finished performing the actions or keystrokes you wanted to record, you can stop recording the macro by clicking the Stop button in the status bar in the lower left.

The sequence of Excel operations or procedures that you recorded will be replayed (meaning the actions will be performed) every time you run the macro.

If you record a macro and then try to save your workbook, you may see an alert like the following.



Microsoft Office Excel

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

- VB project

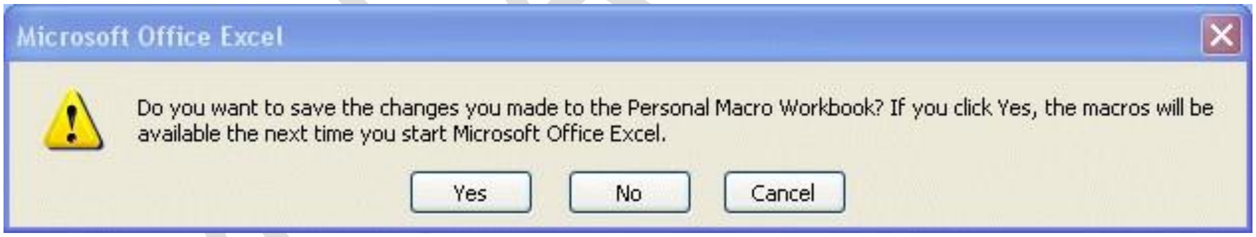
To save a file with these features, click No, and then choose a macro-enabled file type in the File Type list.

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

If you click the Yes button, the macros created and stored in this workbook will not be saved. To keep these macros, you must specify a macro enabled workbook in the Save As type field in the Save As dialogue box.

If the macro is stored in the personal macro workbook, you can run the macro in any workbook you open (given the appropriate security settings), even if the open workbook was not saved as a macro enabled workbook.

If you choose to store a macro in the personal macro workbook, you will get the following message when you close Excel.



Microsoft Office Excel

Do you want to save the changes you made to the Personal Macro Workbook? If you click Yes, the macros will be available the next time you start Microsoft Office Excel.

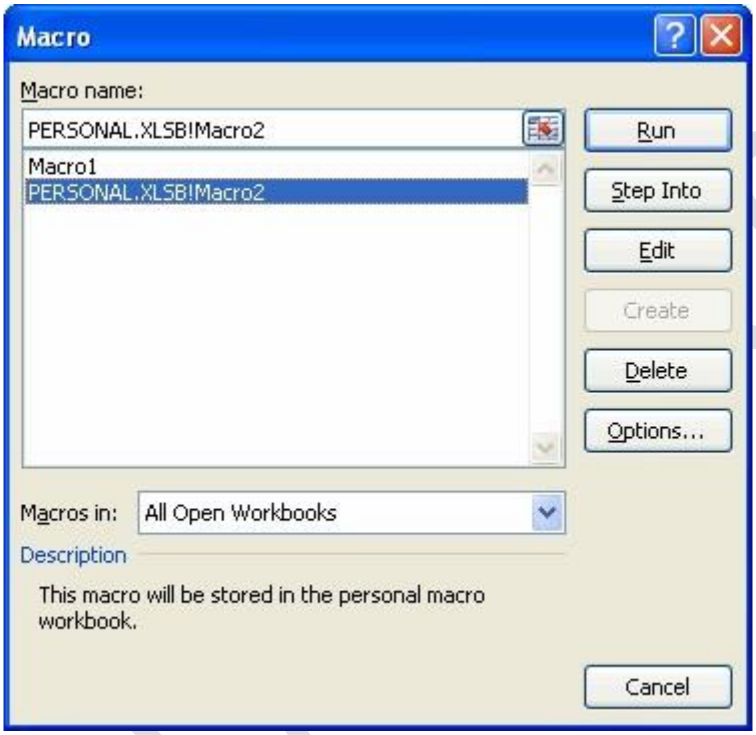
Clicking the Yes button will commit the macro you created to the personal macro workbook. Macros in this personal macro workbook will be available for use when you open Excel.

Playing a Macro

When you play a macro, the actions or keystrokes you chose for your macro will be performed in the sequence in which they were recorded.

To play a macro, click the Macros button on the Developer Ribbon.

When you click this button you will see the Macro dialogue box.



Macro name:
PERSONAL.XLSB!Macro2

Macro1
PERSONAL.XLSB!Macro2

Macros in: All Open Workbooks

Description

This macro will be stored in the personal macro workbook.

Here you can see the macros available for use. Note that macros saved in the personal macro workbook will have Personal.XLSB! in front of their name (such as the macro selected in the list).

To play the selected macro, click the Run button. To view or edit the Visual Basic code for the selected macro, click the Edit button. To delete the selected macro, click the Delete button.

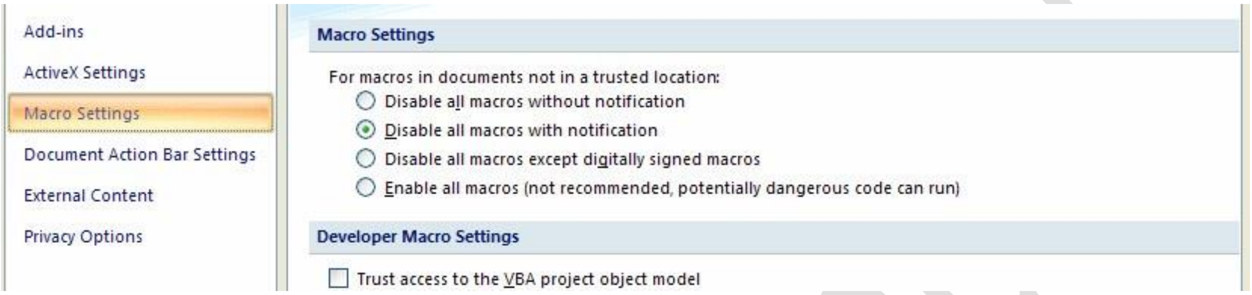
When you click the Run button, the Macro dialogue box will disappear, and the sequence of instructions that was recorded in the macro will be performed on the current workbook.

Note: You can see the macros from this workbook and from the personal macro workbook because the “All open workbooks” option is selected in the bottom drop list of the macro dialogue box. If you select the “This workbook” option from the list, or the “Personal.xlsb” option, only the macros corresponding to the specific option you select will be shown.

You can also run a macro by assigning and using shortcut keys (see ‘Assigning a Keystroke to a Macro’).

Macro Security

It is well known that VBA macros can be a source or carrier of computer viruses. Because of this, you should never use a macro if you do not know where it came from, and you should never open a document that contains a macro unless you trust the source of the document. If you click the Macro Security button on the Developer tab you will see an Excel Trust Center window open to display a list of options for macro security.



Macro Settings

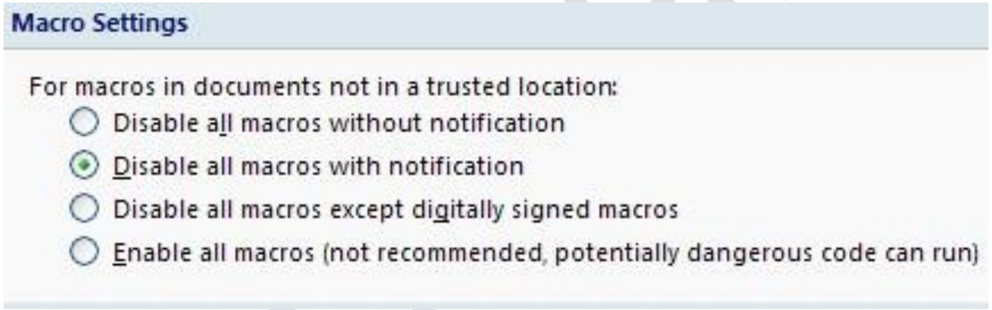
For macros in documents not in a trusted location:

○ Disable all macros without notification
● Disable all macros with notification
○ Disable all macros except digitally signed macros
○ Enable all macros (not recommended, potentially dangerous code can run)

Developer Macro Settings

□ Trust access to the VBA project object model

You can also display these options by first clicking the Excel Options button on the Office menu, and then choosing the Trust Center option followed by the Trust Center Settings button.



Macro Settings

For macros in documents not in a trusted location:

○ Disable all macros without notification
● Disable all macros with notification
○ Disable all macros except digitally signed macros
○ Enable all macros (not recommended, potentially dangerous code can run)

Under the Macro Settings heading, you have four options to choose from:

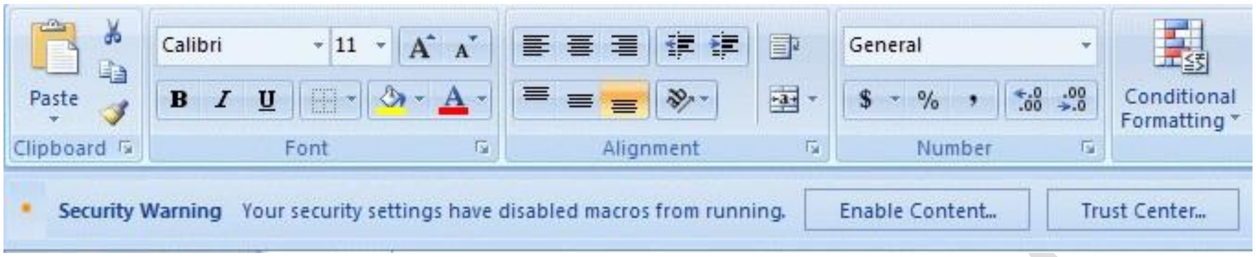
Disable all macros without notification

This setting will prevent macros in files that are not in trusted locations from being run. (Trusted locations are normally on your own hard drive or shared folders on your network.) If you select this option, macros that are not in a trusted location will be disabled. Any alert boxes or messages associated with macro security will also be disabled.

Disable all macros with notification

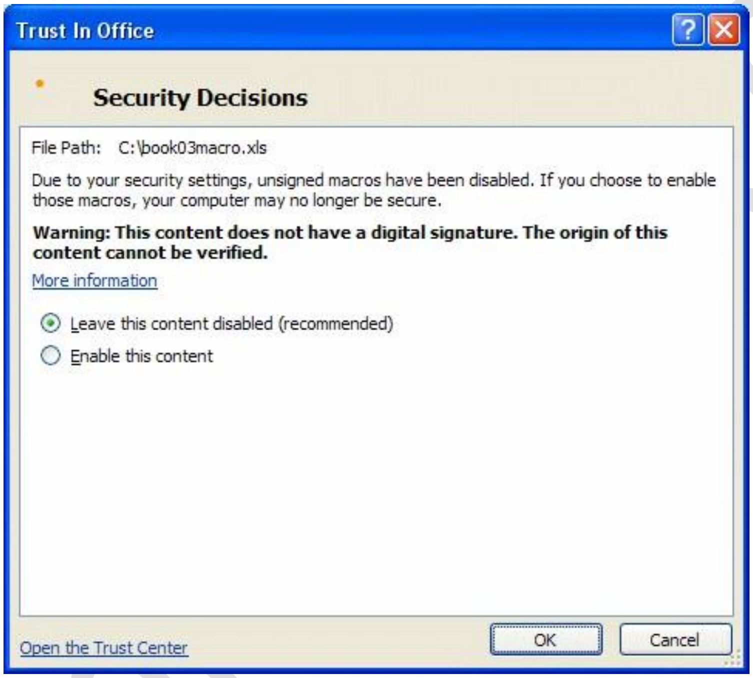
This is Excel 2013’s default macro security setting. With this setting, macros in files that are not in trusted locations will be disabled, but you will still see security alerts if a workbook from a non trusted location contains a macro.

This means you will have the option of choosing to allow or not allow the macro to run.



Security Warning Your security settings have disabled macros from running.

Notice how the security alert appears below the ribbon. If you click the Enable Content button, the following dialogue will appear. To enable the macro, click the “Enable this content” radio button and then click OK.



Trust In Office

Security Decisions

File Path: C:\book03macro.xls

Due to your security settings, unsigned macros have been disabled. If you choose to enable those macros, your computer may no longer be secure.

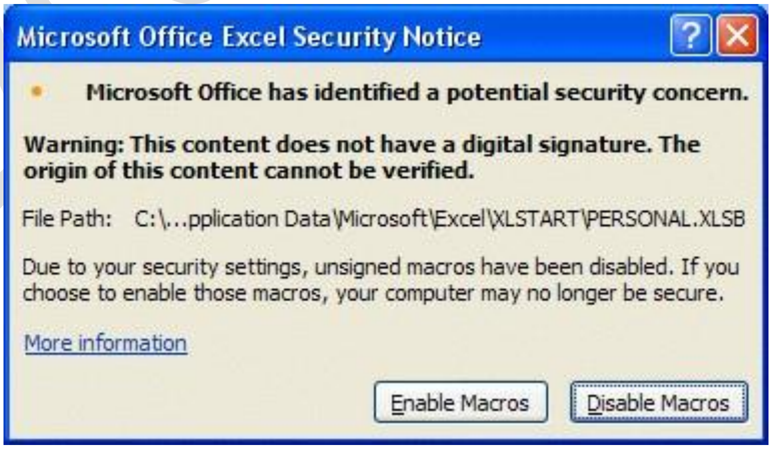
Warning: This content does not have a digital signature. The origin of this content cannot be verified.

More information

● Leave this content disabled (recommended)

○ Enable this content

Depending on your Trust Center settings, you may see a warning dialogue when you open a workbook containing a macro.



Microsoft Office Excel Security Notice

Microsoft Office has identified a potential security concern.

Warning: This content does not have a digital signature. The origin of this content cannot be verified.

File Path: C:\...pplication Data\Microsoft\Excel\XLSTART\PERSONAL.XLSB

Due to your security settings, unsigned macros have been disabled. If you choose to enable those macros, your computer may no longer be secure.

More information

Once again, clicking the Enable Macros button will allow you to run the macro.

Disable all macros except digitally signed macros

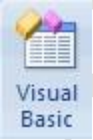
If you choose this setting, Excel will basically behave in the same way as it does when you choose "Disable all macros with notification." The only difference is that if the macro is digitally signed by a publisher that is trusted, you will be able to run the macro. If a macro is digitally signed but the publisher is not trusted, you will then have the option to either enable the macros or to trust the publisher. Any macro that is not digitally signed will not be allowed to run.

Enable All Macros

This is the weakest macro security setting available. If you use this setting, a macro from any workbook can be run, even if it is not in a trusted location and it is not digitally signed. Because of the risks associated with running unknown macros, use this setting with caution.

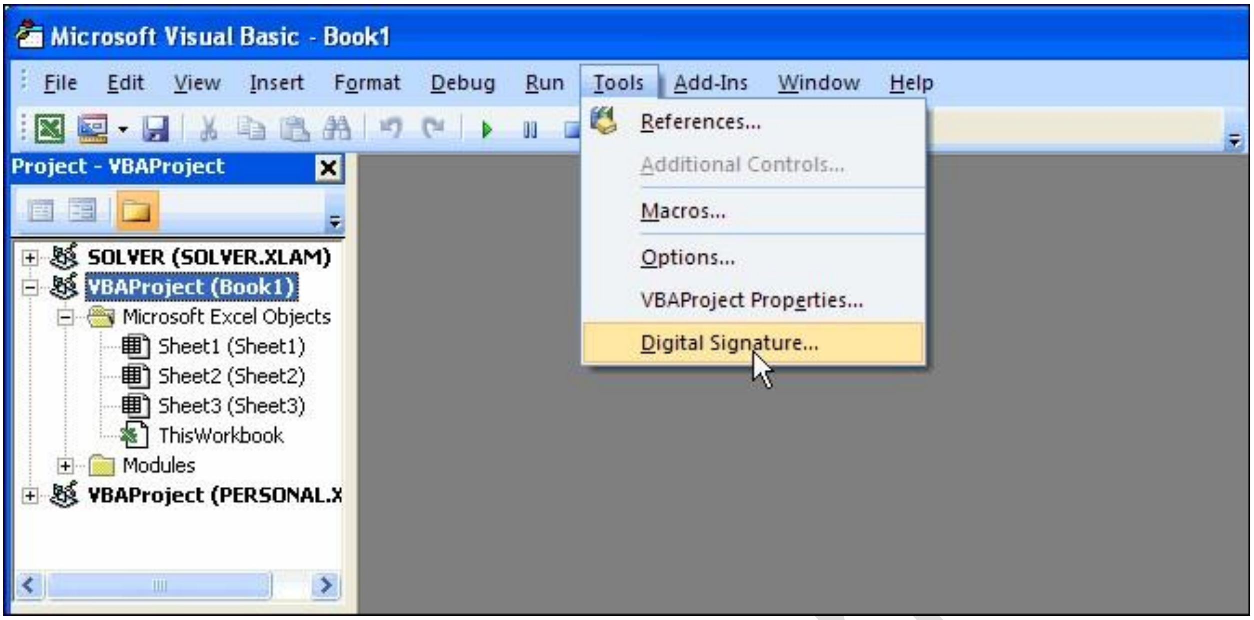
To set the macro security level, select the radio button corresponding to the level of security that you want and click the OK button in the lower right of the Trust Center window.

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.

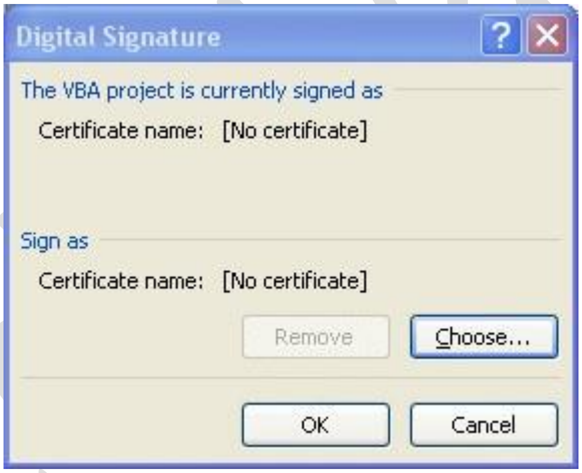


In the tool bar at the top of the Visual Basic Editor's window, click the Project Explorer button.

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.


Digital Signature

The VBA project is currently signed as

Certificate name: [No certificate]

Sign as

Certificate name: [No certificate]

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

More Macro Tasks

Once you get comfortable recording some basic macros, you can begin to explore some additional macro related features and tasks.

In this lesson, you will learn how to record a relative reference macro, how to assign a keystroke to a macro, and how to copy a macro from another workbook or template. Finally, you will learn how to delete workbook macros.

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.

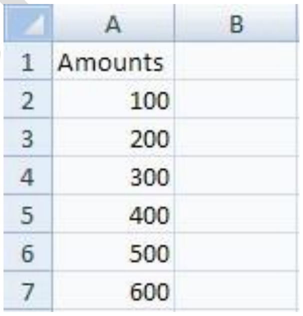
Once you click this button, simply click the Record Macro button as before to create the actual macro. When you stop the macro recording process, all of the cells that were involved in the macro will be treated as having relative references.

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

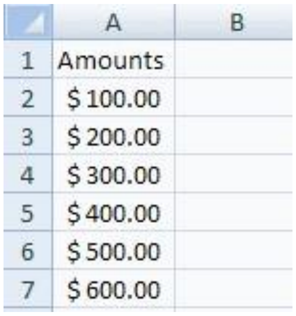
To run a relative reference macro, first make sure that the active cell is chosen such that the cells that you want the macro to run on are in the correct relative position.

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.



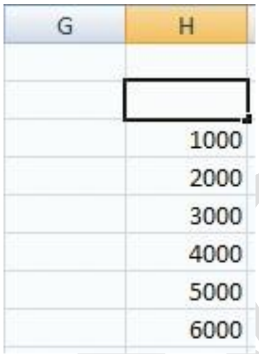
	A	B
1	Amounts	
2	100	
3	200	
4	300	
5	400	
6	500	
7	600	

Here is the data after recording the macro.



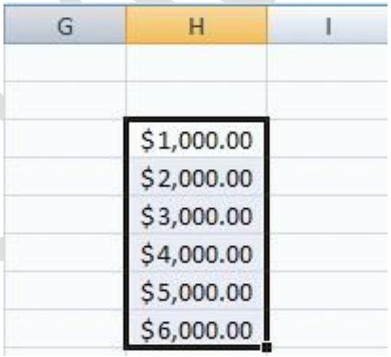
	A	B
1	Amounts	
2	$ 100.00	
3	$ 200.00	
4	$ 300.00	
5	$ 400.00	
6	$ 500.00	
7	$ 600.00	

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.



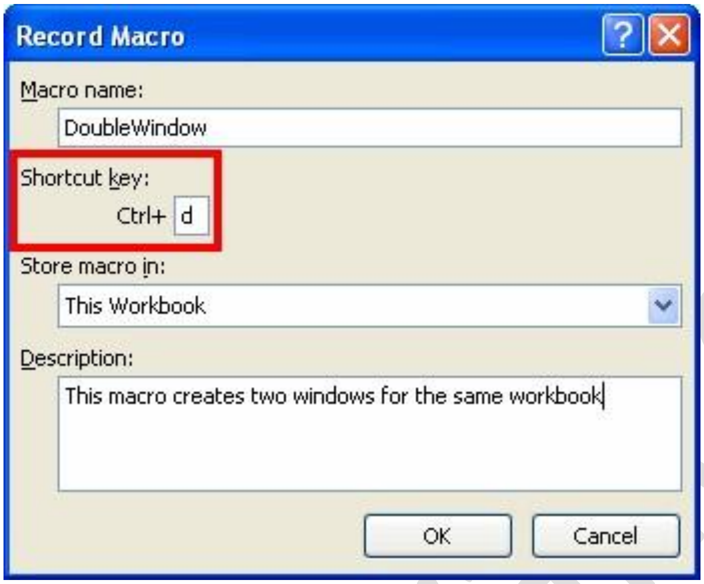
G	H	I
	$1,000.00	
	$2,000.00	
	$3,000.00	
	$4,000.00	
	$5,000.00	
	$6,000.00	

Assigning a Keystroke to a Macro

When assigning macro shortcut keys, keep in mind that if the shortcut already has a task associated with it, the macro will now override the original purpose of the shortcut. For example, assigning Ctrl + N as a shortcut for a macro will disable Ctrl + N as a shortcut for opening a new workbook.

There are two ways to assign a shortcut key to a macro. You can assign a shortcut key when you are first recording your macro or you can assign a short cut key to an existing macro.

To assign a shortcut to a macro when you record it, click the Record Macro button on the Developer Ribbon. When you see the Record Macro box, just enter a letter in the Shortcut Key data field.



Record Macro

Macro name: DoubleWindow

Shortcut key: Ctrl+ d

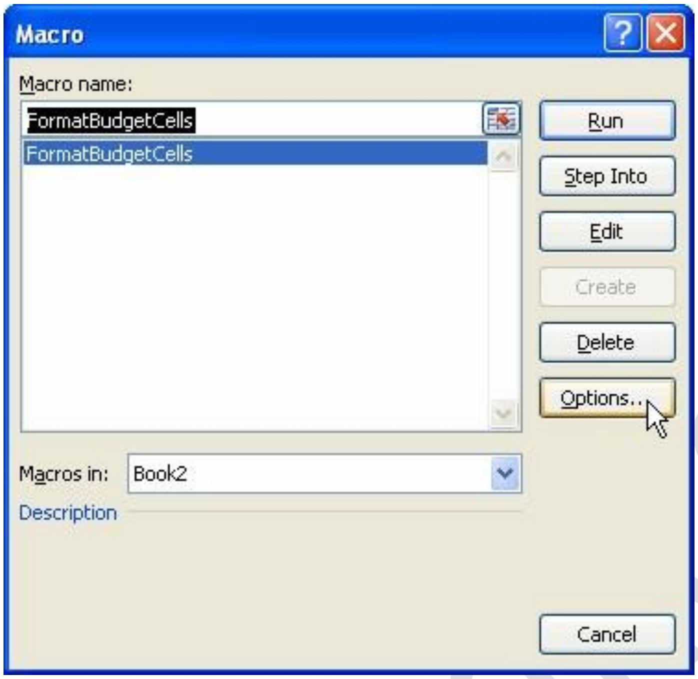
Store macro in: This Workbook

Description: This macro creates two windows for the same workbook

Here you can see that the letter d has been assigned as a shortcut key for this macro. Once the macro has been recorded, pressing Ctrl + d will run it. This avoids the process of picking a macro from the Macro dialogue box every time you want to run it. If you press Shift + D when you enter the shortcut key, the shortcut will then become Ctrl + Shift + D for the macro.

If you store the macro in the personal macro workbook, the shortcut will be carried with it wherever you use the macro.

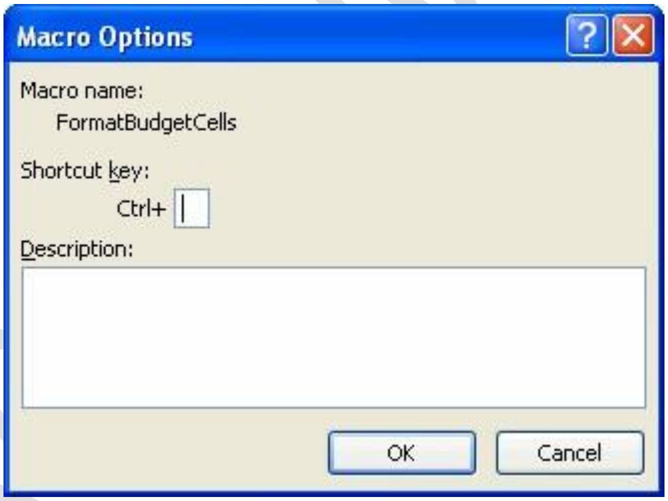
To assign a shortcut to an existing macro, click the Macro button to display the Macro dialogue box. In the Macro dialogue box, select the name of the macro you want to assign a shortcut key to and then click the Options button.



Macro name:
FormatBudgetCells

Macros in: Book2

This will open the Macro Options dialogue box for the macro you selected.



Macro name:
FormatBudgetCells

Shortcut key:
Ctrl+

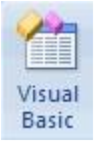
Description:

There is a small data field that can be used to enter a shortcut key for this macro. You can enter a letter or use a Shift key/letter combination as before. If you click OK, the shortcut key will be assigned to the macro.

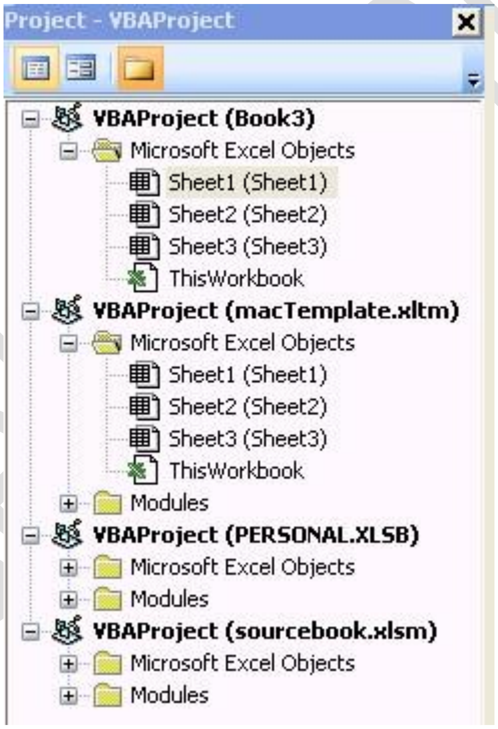
Copying a Macro from a Workbook or Template

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.



```
Project - VBAProject
VBAProject (Book3)
    Microsoft Excel Objects
        Sheet1 (Sheet1)
        Sheet2 (Sheet2)
        Sheet3 (Sheet3)
        ThisWorkbook
VBAProject (macTemplate.xltm)
    Microsoft Excel Objects
        Sheet1 (Sheet1)
        Sheet2 (Sheet2)
        Sheet3 (Sheet3)
        ThisWorkbook
    Modules
VBAProject (PERSONAL.XLSB)
    Microsoft Excel Objects
    Modules
VBAProject (sourcebook.xlsm)
    Microsoft Excel Objects
    Modules
```

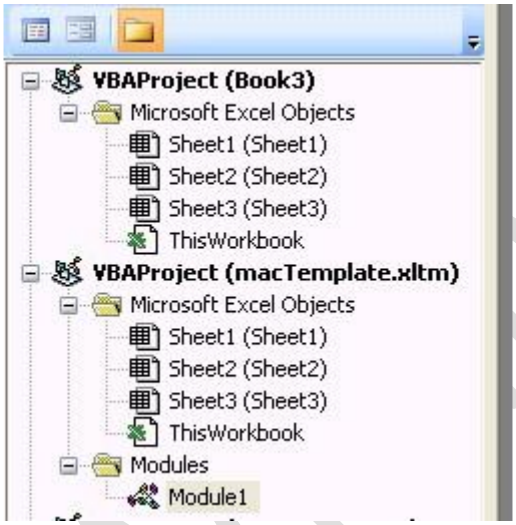
(If you cannot see the Project Explorer panel, click the Project Explorer button in the Visual Basic Editor toolbar.)

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.

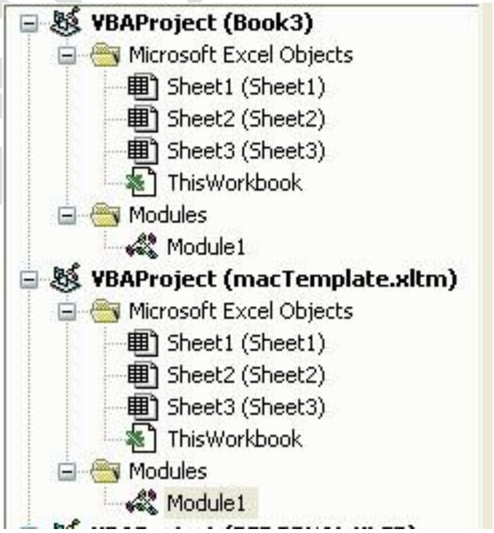
To copy a macro from a template or a workbook, open the module folder for the workbook or template that you want to copy from by clicking on the plus (+) sign corresponding to the folder.

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



VBAProject (Book3)
Microsoft Excel Objects
Sheet1 (Sheet1)
Sheet2 (Sheet2)
Sheet3 (Sheet3)
ThisWorkbook
VBAProject (macTemplate.xltm)
Microsoft Excel Objects
Sheet1 (Sheet1)
Sheet2 (Sheet2)
Sheet3 (Sheet3)
ThisWorkbook
Modules
Module1

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 (+).

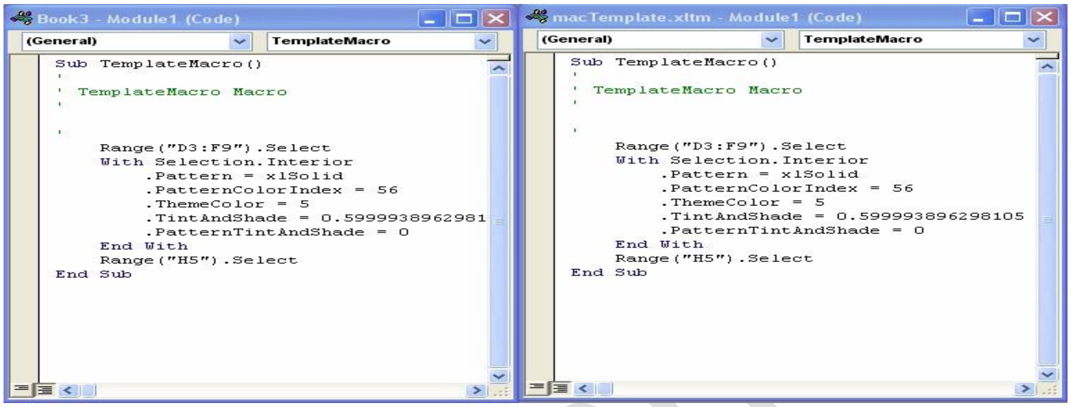


VBAProject (Book3)
Microsoft Excel Objects
Sheet1 (Sheet1)
Sheet2 (Sheet2)
Sheet3 (Sheet3)
ThisWorkbook
Modules
Module1
VBAProject (macTemplate.xltm)
Microsoft Excel Objects
Sheet1 (Sheet1)
Sheet2 (Sheet2)
Sheet3 (Sheet3)
ThisWorkbook
Modules
Module1

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.



```
Sub TemplateMacro()
'
' TemplateMacro Macro
'

'
    Range("D3:F9").Select
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = 56
        .ThemeColor = 5
        .TintAndShade = 0.5999938962981
        .PatternTintAndShade = 0
    End With
    Range("H5").Select
End Sub
```

```
Sub TemplateMacro()
'
' TemplateMacro Macro
'

'
    Range("D3:F9").Select
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = 56
        .ThemeColor = 5
        .TintAndShade = 0.599993896298105
        .PatternTintAndShade = 0
    End With
    Range("H5").Select
End Sub
```

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