

1. Cell References and Naming

1.1 Using Cell References in Formulas

A cell reference identifies a cell or group of cells in a workbook. When you include cell references in a formula, the formula is linked to the referenced cells. The resulting value of the formula is dependent on the values in the referenced cells and changes automatically when the values in the referenced cells change. When you enter or edit a formula, press **[F4]** to change reference types quickly. *For example, from B2 to \$B\$2 to B\$2 to \$B2.*

1.1.1 Relative References

Relative References refer to cells by their position in relation to the cell that contains the formula. *For example, a relative reference to cell A1 looks like this: =A1.*

1.1.2 Absolute References

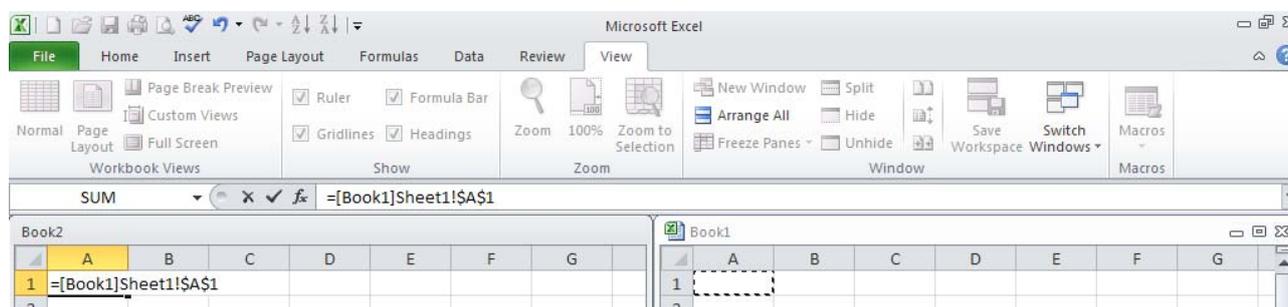
Absolute References refer to cells by their fixed position in the worksheet. *For example, an absolute reference to cell A1 looks like this: =\$A\$1.*

1.1.3 Mixed Reference

Mixed Reference contains a relative reference and an absolute reference. *For example, you can combine relative and absolute references to cell A1 to create these mixed references: =\$A1 or =A\$1.*

1.1.4 External References

You can refer to cells in worksheets in separate workbooks in the same way that you refer to cells in other worksheets within the same workbook.



1.2 How Copying Affects Cell References

One of the handiest things about using references is the capability to copy and paste formulas. But you need to understand what happens to your references after you paste so that you can create formulas with references that operate the way you want them to.

1.2.1 Copying Relative References

When you copy a cell containing a formula with relative cell references, the references change automatically, relative to the position of the cell where you paste the formula. *For example, suppose you type the formula =AVERAGE(B4:E4) in cell F4. You want to repeat this calculation for the remaining rows as well, you select cell F4 and choose Copy. Then you select cells F5:F8, choose Paste Special.* Because the formula in cell F4 contains a relative reference, Excel adjusts the references in each copy of the formula. *For example, cell F7 contains the formula =AVERAGE(B7:E7).*

F4		=AVERAGE(B4:E4)									
	A	B	C	D	E	F	G	H	I		
1	First Quarter Exam Scores										
2											
3	<i>Student</i>	<i>Exam 1</i>	<i>Exam 2</i>	<i>Exam 3</i>	<i>Exam 4</i>	<i>Average</i>					
4	Carothers, Andy	87	90	79	96	88.00					
5	Groncki, Douglas	92	94	94	97						
6	MacDonald, Scott	95	95	95	80						
7	Nusbaum, Tawana	85	87	87	88						
8	Rothenberg, Eric	81	88	88	85						
9											
10											

F7		=AVERAGE(B7:E7)									
	A	B	C	D	E	F	G	H	I		
1	First Quarter Exam Scores										
2											
3	<i>Student</i>	<i>Exam 1</i>	<i>Exam 2</i>	<i>Exam 3</i>	<i>Exam 4</i>	<i>Average</i>					
4	Carothers, Andy	87	90	79	96	88.00					
5	Groncki, Douglas	92	94	94	97	94.25					
6	MacDonald, Scott	95	95	95	80	91.50					
7	Nusbaum, Tawana	85	87	87	88	86.75					
8	Rothenberg, Eric	81	88	88	85	85.50					
9											
10											

1.2.2 Copying Absolute References

If you want cell references to remain the same when you copy them, you must change the reference to an absolute reference by pressing [F4]. *For example, cell C5 contains the relative reference formula =B2*B5. After you change cell B2 to an absolute reference, the result in cell C5 is =\$B\$2*B5. When you copy this modified formula to cells C6:C8, cell C8 now contains the formula: =\$B\$2*B8*

C5		=B2*B5		
	A	B	C	D
2	Hourly Rate	22.25		
3				
4	Name	Hours Worked	Wages Due	
5	DeVoe, Michael	27	600.75	
6	Fakhouri, Fadi	32		
7	Ito, Shu	40		
8	Ortiz, David J.	29		
9				
10				

C8		=\$B\$2*B8		
	A	B	C	D
2	Hourly Rate	22.25		
3				
4	Name	Hours Worked	Wages Due	
5	DeVoe, Michael	27	600.75	
6	Fakhouri, Fadi	32	712.00	
7	Ito, Shu	40	890.00	
8	Ortiz, David J.	29	645.25	
9				
10				

1.2.3 Copying Mixed References

When you copy a mixed reference, Excel anchors the absolute portion and adjusts the relative portion to reflect the location of the cell to which you copied the formula. *For example, cell C6 contains the formula =PMT(\$B6,\$C\$3,C\$5), the first cell reference, \$B6, indicates that we always want to refer to the values in column B but the row reference can change. Similarly, the mixed reference, C\$5, indicates that we always want to refer to the values in row 5 but the column reference can change. Therefore, cell E8 contains the formula =PMT(\$B8,\$C\$3,E\$5).*

C6		fx =PMT(\$B6,\$C\$3,C\$5)					
	A	B	C	D	E	F	G
1							
2		Loan Payment Calculator					
3		Years:	15				
4			Loan Amount				
5		Rate:	\$ 10,000	\$ 20,000	\$ 30,000	\$ 40,000	
6		6.00%	1,030	2,059	3,089	4,119	
7		6.50%	1,064	2,127	3,191	4,254	
8		7.00%	1,098	2,196	3,294	4,392	
9		7.50%	1,133	2,266	3,399	4,531	
10		8.00%	1,168	2,337	3,505	4,673	
11							

1.3 Naming Cells and Cell Ranges

1.3.1 Using Names in Formulas

When you use the name of a cell or a range in a formula by press [Ctrl] + [F3] to display the **Define Name** dialog box instantly, the result is the same as if you entered the cell or range address. *For example, suppose you entered the formula =A1+A2 in cell A3. If you defined the name "Test" as cell A1 and the name "Exam" as cell A2, the formula =Test+Exam has the same result.*

Keep the following basics in mind when using names in formulas:

- The Name box normally displays the address of the selected cell. If the selected cell or range is named, the name takes precedence over the address and is displayed in the Name box.
- When you define a name for a range of cells, the range name does not appear in the Name box unless the entire range is selected.
- When you click the Name box and select a name, the cell selection switches to the named cells.
- If you type a name that has already been defined into the Name box, Excel switches the selection instead of redefining the name.
- When you define a name, the address includes the worksheet name and the cell reference is absolute. For example, when you define the name TestName for cell C5 in Sheet1, the actual name definition is recorded as Sheet1!\$C\$5.

1.3.2 Rules for Naming

The following rules apply when you name cells and ranges in Excel:

- All names must begin with a letter, a backslash (\), or an underscore (_).
- Numbers can be used.
- Spaces cannot be used.
- Excel translates blank spaces in labels to underscores in defined names.
- Symbols other than backslash and underscore can't be used.
- Names that resemble cell references (*for example, AB\$5 or RIC7*) can't be used.
- Single letters, with the exception of the letters R and C, can be used as names.

A name can contain 255 characters, but if it contains more than 253 characters, you can't select it from the Name box. Excel does not distinguish between uppercase and lowercase characters in names. *For example, if you create the name Test and then create the name TEST in the same workbook, the first name is overwritten by the second.*

1.3.3 Creating Names

You can choose **Create from Selection** in **Formulas** tab to name several adjacent cells or ranges at once, using row labels, column labels, or both. Select the appropriate **Create names from values in the** options for the selected cells in the **Create Names from Selection** dialog box, and Excel uses the labels included in the selection to name each range. Please note that when using the Create Names command you actually need to select the labels as well as the data.

The left screenshot shows the 'Create Names from Selection' dialog box with the following options selected:

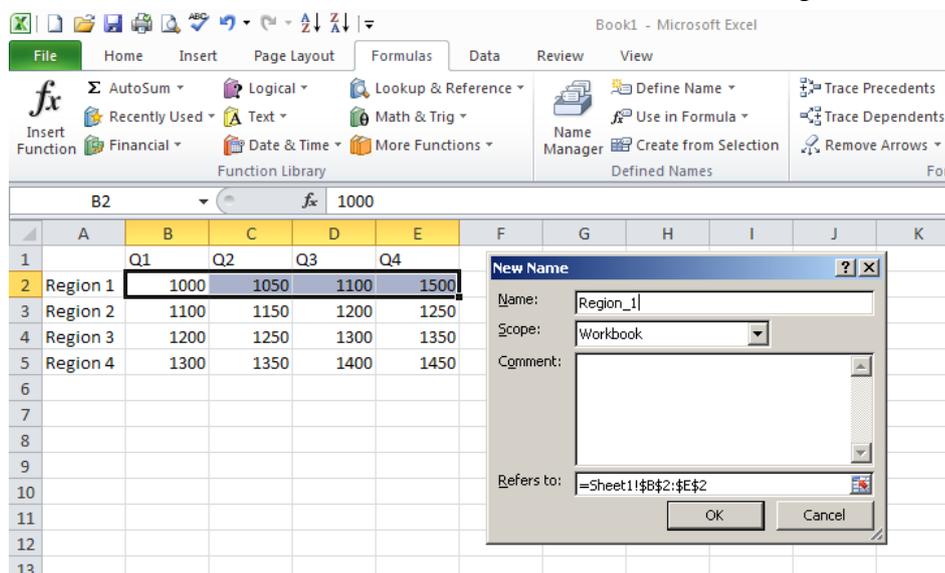
- Top row
- Left column
- Bottom row
- Right column

The right screenshot shows the resulting named ranges in the Name box:

Q1_	Q2_	Q3_	Q4_
Region_1	1000	1050	1100
Region_2	1100	1150	1200
Region_3	1200	1250	1300
Region_4	1300	1350	1400

1.3.4 Defining and Managing Names

Instead of coming up with new names for cells and ranges, you can simply use existing text labels to create names. Choosing **Define Name**, you can use text in adjacent cells to define cell and range names. You can choose this command also to redefine existing names.

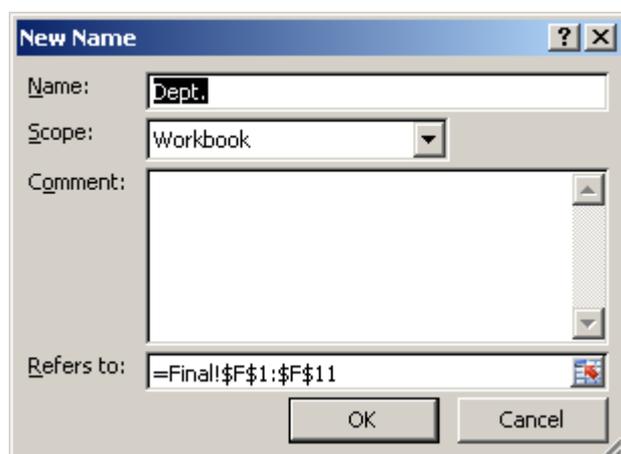


1.3.5 Creating Three-Dimensional Names

You can create three-dimensional names, which use 3-D references as their definitions. For example, suppose you have a 13-sheet workbook containing one identical sheet for each month plus one summary sheet. You can define a 3-D name that can be used to summarize totals from each monthly sheet by following these steps:

1. Select the target cell (*for example, cell B5*) in worksheet.
2. Choose **Formulas** tab, **Define Name** group, **Define Name**.
3. Type any name you choose in the **Names In Workbook** box and type
4. Type the formula (*for example, =Sheet1:Sheet12!B5*) in the **Refers To** box.
5. Press **[Enter]** to confirm

Now you can use this three-dimensional name in formulas that contain any of the following functions: SUM, MIN, VAR, COUNTA, PRODUCT, VARP, AVERAGE, STDEV, COUNT, MAX, and STDEVP.

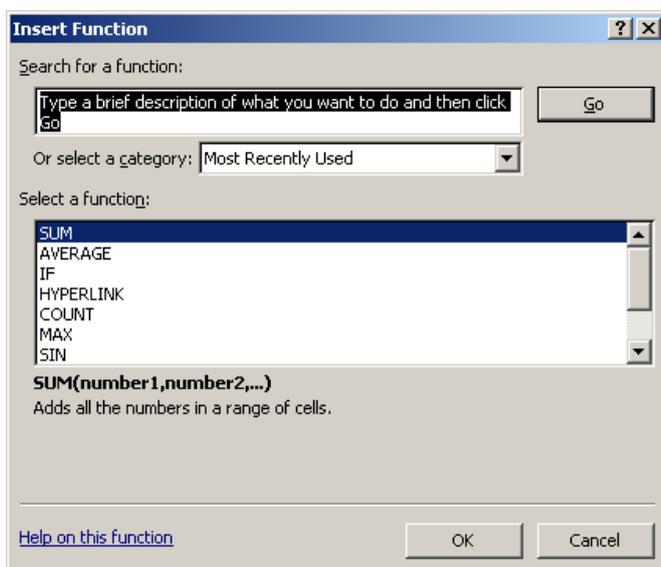


2. Using Function

In simplest terms, a function is a predefined formula. Many Excel functions are shorthand versions of frequently used formulas. For example, the SUM function adds a series of cell values by selecting a range by consolidate the formula =A1+A2+A3+A4+A5 to the formula =SUM(A1:A5).

2.1 Inserting Functions

When you want to use a built-in function, select **Formulas** tab, **Insert Function** to display the **Insert Function** dialog box. This dialog box gives you access to all built-in functions in Excel.



2.2 Error Values

An error value is the result of a formula that Excel can't resolve. The seven error values are

Error	Reason
#DIV/0!	You attempted to divide a number by zero. This error usually occurs when you create a formula with a divisor that refers to a blank cell.
#NAME?	You entered a name in a formula that isn't in the Define Name dialog box list. You might have mistyped the name or typed a deleted name. Excel also displays this error value if you do not enclose a text string in quotation marks.
#VALUE	You entered a mathematical formula that refers to a text entry.
#REF!	You deleted a range of cells whose references are included in a formula.
#N/A	No information is available for the calculation you want to perform.
#NUM!	You provided an invalid argument to a worksheet function. It also indicates that the result of a formula is too large or too small to be represented.
#NULL!	You included a space between two ranges in a formula to indicate an intersection, but the ranges have no common cells.

2.3 Mathematical Functions

2.3.1 The SUM Function

The SUM function totals a series of numbers. It takes the form =SUM(number1, number2, ...). The number arguments are a series of as many as 30 entries that can be numbers, formulas, ranges, or cell references that result in numbers. SUM ignores arguments that refer to text values, logical values, or blank cells.

2.3.2 The MOD Function

The MOD function returns the remainder of a division operation (modulus). It takes the arguments (number, divisor). The result of the MOD function is the remainder produced when number is divided by divisor. *For example, the function =MOD(9, 4) returns 1 (the remainder)..*

2.3.3 The COMBIN Function

The COMBIN function determines the number of possible combinations that can be taken from a pool of items. It takes the arguments (number, number_chosen), where number is the total number of items in the pool and number_chosen is the number of items you want to group in each combination. *For example, to determine how many different combination for Mark Six, type the formula =COMBIN(49, 6). The result indicates that 13983816 combinations could be created.*

2.3.4 The RAND Functions

The RAND function generates a random number between 0 and 1. It is one of the few Excel functions that do not take an argument. Note that you must still type the parentheses after the function name. The result of a RAND function changes each time you recalculate your worksheet. If you use automatic recalculation, the value of the RAND function changes each time you make a worksheet entry.

2.3.5 The ROUND, ROUNDDOWN, and ROUNDUP Functions

The ROUND function rounds a number to a specified number of decimal places, rounding digits less than 5 down and digits greater than or equal to 5 up. It takes the arguments (number, num_digits). If num_digits is a positive number, then number is rounded to the specified number of decimal points; if num_digits is negative, the function rounds to the left of the decimal point; if num_digits is 0, the function rounds to the nearest integer. *For example, the formula =ROUND(123.4567, -2) returns 100, and the formula =ROUND(123.4567, 3) returns 123.457. The ROUNDDOWN and ROUNDUP functions take the same form as ROUND. As their names imply, they always round down or up, respectively.*

2.3.6 The EVEN and ODD Functions

The EVEN function rounds a number up to the nearest even integer. The ODD function rounds a number up to the nearest odd integer. Negative numbers are correspondingly rounded down. *For example, the formula =EVEN(23.4) returns 24, and the formula =ODD(-4) returns -5.*

2.3.7 The FLOOR and CEILING Functions

The FLOOR function rounds a number down to the nearest given multiple, and the CEILING function rounds a number up to the nearest given multiple. These functions take the arguments (number, multiple). *For example, the formula =FLOOR(23.4, 0.5) returns 23, and the formula =CEILING(5, 1.5) returns 6.*

2.3.8 The INT Function

The INT function rounds numbers down to the nearest integer. When a number is negative, INT also rounds that number down to the next integer. *For example, the formulas =INT(100.01) and =INT(100.99999999) both return the value 100, even though the number 100.99999999 is essentially equal to 101. If each of the numbers in the examples were negative, the resulting value would be -101.*

2.3.9 The TRUNC Function

The TRUNC function truncates everything to the right of the decimal point in a number, regardless of its sign. It takes the arguments (number, num_digits). If num_digits isn't specified, it's set to 0. Otherwise, TRUNC truncates everything after the specified number of digits to the right of the decimal point. *For example, the formula =TRUNC(13.978) returns the value 13; the formula =TRUNC(13.978, 1) returns the value 13.9.*

2.4 Text Functions

Text functions in Excel are some of the most useful word-processing and data-management tools you'll find anywhere because they do things that word-processing programs can't do.

2.4.1 The TEXT Function

The TEXT function converts a number into a text string with a specified format. Its arguments are (value, format_text), where value represents any number, formula, or cell reference; and format_text is the format for displaying the resulting string. *For example, the formula =TEXT(98/4, "0.00") returns the text string 24.50. You can use any Excel formatting symbol (\$, #, 0, and so on) except the asterisk (*) to specify the format you want, but you can't use the General format.*

2.4.2 The DOLLAR Function

Like the TEXT function, the DOLLAR function converts a number into a string. DOLLAR, however, formats the resulting string as currency with the number of decimal places you specify. The arguments (number, decimals) specify a number or reference and the number of decimal places you want. *For example, the formula =DOLLAR(45.899, 2) returns the text string \$45.90. Notice that Excel rounds the number when necessary.*

If you omit decimals, Excel uses two decimal places. If you add a comma after the first argument but omit the second argument, Excel uses zero decimal places. If you use a negative number for decimals, Excel rounds to the left of the decimal point.

2.4.3 The LEN Function

The LEN function returns the number of characters in an entry. The single argument can be a number, a string enclosed in double quotation marks, or a reference to a cell. Trailing zeros are ignored. *For example, the formula =LEN("Test") returns 4.*

2.4.4 The ASCII Functions: CHAR and CODE

Every computer uses numeric codes to represent characters. The most prevalent system of numeric codes is ASCII (American Standard Code for Information Interchange). ASCII uses a number from 0 to 127 (or in some systems, to 255) to represent each number, letter, and symbol.

The CHAR and CODE functions deal with these ASCII codes. The CHAR function returns the character that corresponds to an ASCII code number; the CODE function returns the ASCII code number for the first character of its argument. For example, the formula =CHAR(83) returns the text "S". The formula =CODE("S") returns the ASCII code 83. If you type a literal character as the text argument, be sure to enclose the character in quotation marks; otherwise, Excel returns the #NAME? error value.

2.4.5 The Clean-Up Functions: TRIM and CLEAN

Leading and trailing blank characters often prevent you from correctly sorting entries in a worksheet or a database. If you use string functions to manipulate text in your worksheet, extra spaces can prevent your formulas from working correctly.

The TRIM function eliminates leading, trailing, and extra blank characters from a string, leaving only single spaces between words.

The CLEAN function is similar to TRIM, except it operates on only nonprintable characters, such as tabs and program-specific codes. CLEAN is especially useful if you import data from another program or operating system, because the translation process often introduces nonprintable characters that appear as symbols or boxes. You can use CLEAN to remove these characters from the data.

2.4.6 The EXACT Function

The EXACT function is a conditional function that determines whether two strings match exactly. The function ignores formatting, but it is case-sensitive, so uppercase letters are considered different than lowercase letters. If both strings are identical, the function returns TRUE. Both arguments must be literal strings enclosed in quotation marks, references to cells that contain text, numeric values, or formulas that evaluate to numeric values. *For example, if cell A5 and cell A6 of your worksheet both contain the text “Totals”, the formula =EXACT(A5, A6) returns TRUE.*

2.4.7 The Case Functions: UPPER, LOWER, and PROPER

Three functions manipulate the case of characters in text strings. The UPPER and LOWER functions convert text strings to all uppercase or all lowercase letters. The PROPER function capitalizes the first letter in each word, capitalizes any other letters in the text string that do not follow another letter, and converts all other letters to lowercase. For example, if cell A1 contains the text “hello World”, you can type the formula =UPPER(A1) to return “HELLO WORLD”. Similarly, the formula =LOWER(A1) returns “hello world”, and =PROPER(A1) returns “Hello World”.

2.5 Substring Text Functions

The following functions locate and return portions of a text string or assemble larger strings from smaller ones: FIND, SEARCH, RIGHT, LEFT, MID, SUBSTITUTE, REPLACE, and CONCATENATE.

2.5.1 The FIND and SEARCH Functions

You use the FIND and SEARCH functions to locate the position of a substring within a string. Both functions return the position in the string of the character you specify (Excel counts blank spaces and punctuation marks as characters). These two functions work the same way, except FIND is case sensitive and SEARCH allows wildcards. Both functions take the same arguments: (find_text, within_text, start_num). The optional start_num argument is helpful when within_text contains more than one occurrence of find_text. If you omit start_num, Excel reports the first match it locates. *For example, to locate the “x” in the string “Advanced Microsoft Excel”, you would type the formula =FIND("x", "Advanced Microsoft Excel"). The formula returns 21, because “x” is the 21st character in the string.*

If you’re not sure of the character sequence you’re searching for, you can use the SEARCH function and include wildcards in your find_text string. *Suppose you’ve used the names “Smith” and “Smyth” in your worksheet. To determine whether either name is in cell A1, type the formula =SEARCH("Sm?th", A1). If cell A1 contains the text “John Smith” or “John Smyth”, the SEARCH function returns the value 6 where is the starting point of the string Sm?th.*

2.5.2 The RIGHT and LEFT Functions

The RIGHT function returns the rightmost series of characters from a specified string; the LEFT function returns the leftmost series of characters. These functions take the same arguments: (text, num_chars). The num_chars argument indicates the number of characters to extract from the text argument.

These functions count blank spaces in the text argument as characters; if text contains leading or trailing blank characters, you might want to use a TRIM function within the RIGHT or LEFT function to ensure the expected result. *For example, suppose you type “Advanced Microsoft Excel” in cell A1 of your worksheet. The formula =RIGHT(A1,5) returns the word “Excel”.*

2.5.3 The MID Function

You can use the MID function to extract a series of characters from a text string. This function takes the arguments (text, start_num, num_chars). *For example, if cell A1 contains the text “Advanced Microsoft Excel”, you can type the formula =MID(A1, 10, 9) to extract the text “Microsoft” from the entry in cell A1.*

2.5.4 The REPLACE and SUBSTITUTE Functions

The REPLACE and SUBSTITUTE functions substitute new text for old text. The REPLACE function replaces one string of characters with another string of characters and takes the arguments (old_text, start_num, num_chars, new_text). *Suppose cell A1 contains the text “Advanced Microsoft Excel”. To replace the first four characters with the string “Beginning”, type the formula =REPLACE(A1, 1, 8, "Beginning"). The result is “Beginning Microsoft Excel”.*

With the SUBSTITUTE function, you specify the text to replace. The function takes the arguments (text, old_text, new_text, instance_num). The instance_num argument optionally replaces only the specified occurrence of old_text. If you don't include instance_num, Excel changes all occurrences of old_text to new_text. *Suppose cell A1 contains the text “Advanced Microsoft Excel” and you want to place it in cell A2 but change it to “Advanced Microsoft Word”. Type this formula in cell A2 =SUBSTITUTE(A1,"Excel","Word").*

2.5.5 The CONCATENATE Function

To assemble strings from up to 30 smaller strings or references, the CONCATENATE function is the function equivalent of the & character. *For example, if cell A1 contains the text “Hello” with a trailing space character, the formula =CONCATENATE(A1 "World") returns “Hello World”.*

2.6 Logical Functions

You use logical functions to test for specific conditions. These functions are often called logical operators in discussions of Boolean logic. You use logical operators to arrive at one of two conclusions: TRUE or FALSE. We'll discuss the most useful logical functions in this section.

2.6.1 The SUMIF and COUNTIF Function

With SUMIF, you can add specific values in a range, based on a criterion you supply. This performs all the calculations you need in one cell, and eliminates having to create a column of IF formula.

For example, you can type the formula =SUMIF(C12:C27, "Pass", A12:A27) to find the total of all numbers in A12:A27 in which the cell in the same row in column C contains the word "Pass".

Similarly, COUNTIF counts the cells that match specified criteria and takes the arguments (range, criteria). *For example, you can find the number of months in which sales fell below \$600 using a conditional test, as in the formula =COUNTIF(Sales, "<600").*

2.6.2 The IF Function

The IF function returns values based on supplied conditional tests. It takes the arguments (logical_test, value_if_true, value_if_false). *For example, the formula =IF(A1<22, 5, 10) returns 5 if the value in cell A1 is less than 22; otherwise, it returns 10.*

You can nest other functions and use text arguments to return nothing instead of zero if the result is false. *For example, the formula =IF(SUM(A1:A10)>0, SUM(A1:A10), " ") returns a null string (" ") if the conditional test is false.*

2.6.3 The AND, OR, and NOT Functions

The functions work with the logical operators =, >, <, >=, <=, and <>. The AND and OR functions can each have as many as 30 logical arguments. The NOT function takes only one argument which can be conditional tests, arrays, or references to cells that contain logical values).

The OR function returns the logical value TRUE if any one of the conditional tests is true; the AND function returns the logical value TRUE only if all the conditional tests are true. NOT instructs Excel to return the logical value TRUE if the argument is false or the logical value FALSE if the argument is true. The truth table for AND, OR and NOT functions are listed below:

x	y	x AND y
T	T	T
T	F	F
F	T	F
F	F	F

x	y	x OR y
T	T	T
T	F	T
F	T	T
F	F	F

x	NOT x
T	F
F	T

Suppose you want to return the text "Pass" only if a student has an average score above 75 and fewer than five unexcused absences. If we typed the formula =IF(AND(G4<5, F4>75), "Pass", "Fail"). This fails the student in row 5 because of the five absences. If you use OR instead of AND in the formula, all students would pass.

2.7 Information Functions

Information functions allow you to gather information about the contents of cells, their formatting, and the computing environment as well as perform conditional tests for the presence of specific types of values.

2.7.1 The TYPE and ERROR.TYPE Functions

The TYPE function determines whether a cell contains text, a number, a logical value, an array, or an error value. The result is a code for the type of entry in the referenced cell: 1 for a number (or a blank cell), 2 for text, 4 for a logical value (TRUE or FALSE), 16 for an error value, and 64 for an array. *For example, if cell A1 contains the number 100, the formula =TYPE(A1) returns 1. If A1 contains the text "Hello", the formula returns 2.*

Like the TYPE function, the ERROR.TYPE function detects the contents of a cell, except it detects different types of error values. The result is a code for the type of error value in the referenced cell: 1 for #NULL!, 2 for #DIV/0!, 3 for #VALUE!, 4 for #REF!, 5 for #NAME!, 6 for #NUM!, and 7 for #N/A. Any other value in the referenced cell returns the error value #N/A. *For example, if cell A1 contains a formula that displays the error value #NAME!, the formula =ERROR.TYPE(A1) returns 5. If A1 contains the text Microsoft Excel, the formula returns #N/A.*

2.7.2 The COUNTBLANK Function

The COUNTBLANK function counts the number of empty cells in the specified range, which is its only argument. However, this function is tricky because formulas that evaluate to null text strings, such as "=", or to zero might seem empty, but they aren't and therefore won't be counted.

2.7.3 Using the IS Information Functions

You can use the ISBLANK, ISERR, ISERROR, ISLOGICAL, ISNA, ISNONTEXT, ISNUMBER, ISREF, and ISTEXT functions to determine whether a referenced cell or range contains the corresponding type of value. All IS Information functions take a single argument. *For example, the ISBLANK function takes the form =ISBLANK(value). If value refers to a blank cell, the function returns the logical value TRUE; otherwise, it returns FALSE.*

2.7.4 An ISERR Example

You can use ISERR to avoid getting error values as formula results. *For example, the FIND function returns the position at which a substring is found within a larger string. If the substring isn't there, FIND returns #VALUE!. Adding an ISERR function, such as =IF(ISERR(FIND("12A", A1)), " ", "Yes"). Because you're not interested in the error, which is simply a by-product of the calculation, this traps the error, leaving only the results that you are interested in.*

2.8 Lookup Functions

Lookup functions help you use your own worksheet tables as sources of information to be used elsewhere in formulas. You can use three primary functions to look up information stored in a list or a table or to manipulate references: LOOKUP, VLOOKUP, and HLOOKUP.

VLOOKUP and HLOOKUP are nearly identical functions that look up information stored in tables you have constructed. VLOOKUP and HLOOKUP operate in either vertical or horizontal orientation (respectively), but LOOKUP works either way.

When you look up information in a table, you normally use a row index and a column index to locate a particular cell. Excel derives the first index by finding the largest value in the first column or row that is less than or equal to a lookup value you supply and then uses a row number or column number argument as the other index. Make sure the table is sorted by the row or column containing the lookup values. These functions take the following forms:

=VLOOKUP(lookup_value, table_array, col_index_num, range_lookup)

=HLOOKUP(lookup_value, table_array, row_index_num, range_lookup)

The LOOKUP function takes two forms, the first is called the vector form, and the second is called the array form:

=LOOKUP(lookup_value, lookup_vector, result_vector)

=LOOKUP(lookup_value, array)

LOOKUP Function Argument	Description
lookup_value	The value, cell reference, or text (enclosed in quotation marks) that you want to find in a table or a range.
table_array	A cell range or name that defines the table to look in.
row_index_num, col_index_num	The row or column number of the table from which to select the result, counted relative to the table (not according to the actual row and column numbers).
range_lookup	A logical value that determines whether the function matches the lookup_value exactly or approximately. Type FALSE to match the lookup_value exactly. The default is TRUE, which finds the closest match.
lookup_vector	A one-row or one-column range that contains numbers, text, or logical values.
result_vector	A one-row or one-column range that must be the same size as lookup_vector.
array	A range containing numbers, text, or logical values to compare with lookup_value.

2.8.1 The VLOOKUP and HLOOKUP Functions

For the VLOOKUP and HLOOKUP functions, whether a lookup table should be considered vertical or horizontal depends on where the comparison values (the first index) are located. If the values are in the leftmost column of the table, the table is vertical; if they are in the first row of the table, the table is horizontal. The comparison values can be numbers or text, but they must be arranged in ascending order. No comparison value should be used more than once in a table.

B12 fx =VLOOKUP(B11,A2:B8,2)

	A	B	C	D	E
1	Mark above	Grade			
2	0	U			
3	40	F			
4	50	E			
5	60	D			
6	70	C			
7	80	B			
8	90	A			
9					
10					
11	Input Mark:	75			
12	Grade:	C			
13					

B6 fx =HLOOKUP(B5,B1:H2,2)

	A	B	C	D	E	F	G	H
1	Mark above	0	40	50	60	70	80	90
2	Grade	U	F	E	D	C	B	A
3								
4								
5	Input Mark:	75						
6	Grade:	C						
7								

Remember that these lookup functions normally search for the greatest comparison value that is less than or equal to the lookup value, not for an exact match between the comparison values and the lookup value. If all the comparison values in the first row or column of the table range are greater than the lookup value, the function returns the #N/A error value. If all the comparison values are less than the lookup value, however, the function returns the value that corresponds to the last (largest) comparison value in the table, which might not be what you want. If you require an exact match, type FALSE as the range_lookup argument.

B6 fx =HLOOKUP(B5,B1:H2,2)

	A	B	C	D	E	F	G	H
1	Mark above	0	40	50	60	70	80	90
2	Grade	U	F	E	D	C	B	A
3								
4								
5	Input Mark:	-10						
6	Grade:	#N/A						
7								

2.8.2 The LOOKUP Function

The array form of LOOKUP is similar to VLOOKUP and HLOOKUP but works with either a horizontal or a vertical table, using the dimensions of the table to figure out the location of the comparison values. If the table is taller than it is wide or the table is square, the function treats it as a vertical table and assumes that the comparison values are in the leftmost column. If the table is wider than it is tall, the function views the table as horizontal and assumes that the comparison values are in the first row of the table. The result is always in the last row or column of the specified table; you can't specify column or row numbers. Because HLOOKUP and VLOOKUP are more predictable and controllable, you'll generally find using them preferable to using LOOKUP.

The lookup_vector and result_vector arguments are often adjacent ranges, but they don't have to be when you use LOOKUP. They can be located in separate areas of the worksheet, and one range can be horizontal and the other vertical. The only requirement is that they must have the same number of elements.

B14		fx =LOOKUP(B13, B2:B8,B10:H10)						
	A	B	C	D	E	F	G	H
1	Comment	Mark Above						
2	Unclassified	0						
3	Fail	40						
4	Poor	50						
5	Fair	60						
6	Average	70						
7	Good	80						
8	Excellent	90						
9								
10	Grade	U	F	E	D	C	B	A
11	No. of Student	10	20	30	40	30	20	10
12								
13	Input Mark:	75						
14	Grade:	C						
15								

2.9 Reference Functions

2.9.1 The CHOOSE Function

You use the CHOOSE function to retrieve an item from a list of values. The function takes the arguments (index_num, value 1, value 2, ...) and can include up to 29 values. The index_num argument is the position in the list you want to return; it must be positive and can't exceed the number of elements in the list. The function returns the value of the element in the list that occupies the position indicated by index_num. *For example, the function =CHOOSE(2, "A", "B", "C", "D", "E") returns the value "B".*

You can use individual cell references for the list, but you can't specify ranges. You might be tempted to create a function, *such as =CHOOSE(A10, C1:C5), to take the place of the longer function in the preceding example. If you do, however, the result is a #VALUE! error value.*

2.9.2 The MATCH Function

The MATCH function returns the position of the item in the list that most closely matches a lookup value. This function takes the arguments (lookup_value, lookup_array, match_type), where lookup_value and the items in the lookup_array can be numeric values or text strings, and match_type defines the rules for the search.

match_type	Description
1 (default)	Finds the largest value in the specified range (which must be sorted in ascending order) that is less than or equal to lookup_value. If no items in the range meet these criteria, the function returns #N/A.
0	Finds the first value in the specified range (no sorting necessary) that is equal to lookup_value. If no items in the range match, the function returns #N/A.
-1	Finds the smallest value in the specified range (which must be sorted in descending order) that is greater than or equal to lookup_value. If no items in the range meet these criteria, the function returns #N/A.

When you use MATCH to locate text strings, you should specify a match_type argument of 0 (an exact match). You can then use the wildcards * and ? in the lookup_value argument.

E2 fx =MATCH(D2,A2:A5,0)

	A	B	C	D	E
1	Company	Turnover		Input	Return Value
2	IBM	1234		I?M	1
3	Microsoft	3456		M*t	1
4	HP	6789		M?t	#N/A
5	Nvidia	7890			
6					

2.9.3 The INDEX Function

The INDEX function has two forms: an array form, which returns a value, and a reference form, which returns a cell reference. The forms of these functions are

=INDEX(array, row_num, column_num)

=INDEX(reference, row_num, column_num, area_num)

The array form works only with an array argument; it returns the value of the result, not the cell reference. The result is the value at the position in array indicated by row_num and column_num. The reference form returns a cell address instead of a value and is useful when you want to perform operations on a cell, rather than on its value.

A5 fx =INDEX(A1:D3, 3, 2)

	A	B	C	D	E
1	A1	B1	C1	D1	
2	A2	B2	C2	D2	
3	A3	B3	C3	D3	
4					
5	B3				

2.10 Date and Time Functions

2.10.1 The TODAY and NOW Functions

You can enter =TODAY() into a cell or a formula to insert the serial value of the current date. Similarly, you can enter =NOW() into a cell or formula to insert the current date and time. The result of the function is a serial date and time value that includes an integer (the date) and a decimal value (the time).

2.10.2 The YEAR, MONTH, and DAY Functions

The YEAR, MONTH, and DAY functions return the value of the year, month, and day portions of a serial date value. All three take a single argument, which can be a serial date value; a reference to a cell that contains either a date function or a serial date value; or a text date enclosed in quotation marks. *For example, if cell A1 contains the date 31/12/1999, the formula =YEAR(A1) returns the value 1999, the formula =MONTH(A1) returns the value 12, and the formula =DAY(A1) returns the value 31*

2.10.3 The HOUR, MINUTE, and SECOND Functions

Just as the YEAR, MONTH, and DAY functions extract the value of the year, month, and day portions of a serial date value, the HOUR, MINUTE, and SECOND functions extract the value of the hour, minute, and second portions of a serial time value. *For example, if cell A1 contains the time 12:15:35 PM, the formula =HOUR(A1) returns the value 12, the formula =MINUTE(A1) returns the value 15, and the formula =SECOND(A1) returns the value 25.*

2.10.4 The WEEKDAY Function

The WEEKDAY function returns the day of the week for a specific date and takes the arguments (serial_number, return_type). The serial_number argument can be a serial date value; a reference to a cell that contains either a date function or a serial date value; or text, such as 31/12/1999. The function returns a number that represents the day of the week that the specified date falls on. The optional return_type argument determines the way the result is displayed.

If return_type is	WEEKDAY returns
1 (default)	A number from 1 through 7 where 1 is Sunday and 7 is Saturday
2	A number from 1 through 7 where 1 is Monday and 7 is Sunday
3	A number from 0 through 6 where 0 is Monday and 6 is Sunday

3. Auditing Worksheet

Excel has a number of powerful and flexible features that help you audit and debug your worksheets and document your work.

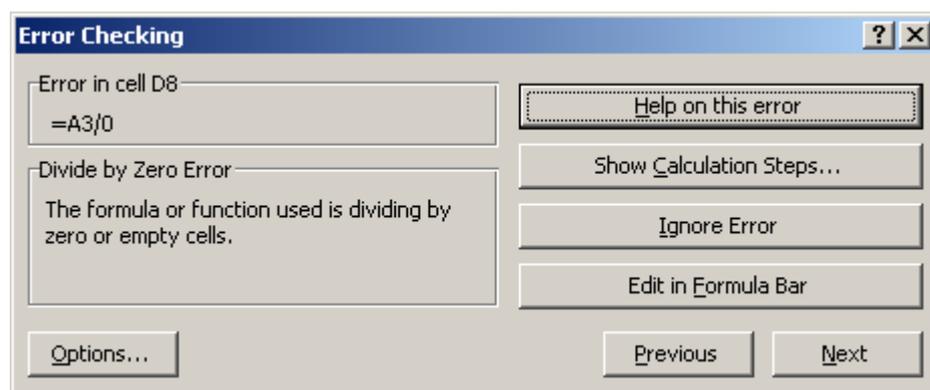
3.1 Redisplay the Active Cell

If you scroll through your worksheet and the active cell is no longer visible, you can redisplay it by pressing **[Ctrl] + [Backspace]** button.

3.2 Error Checking

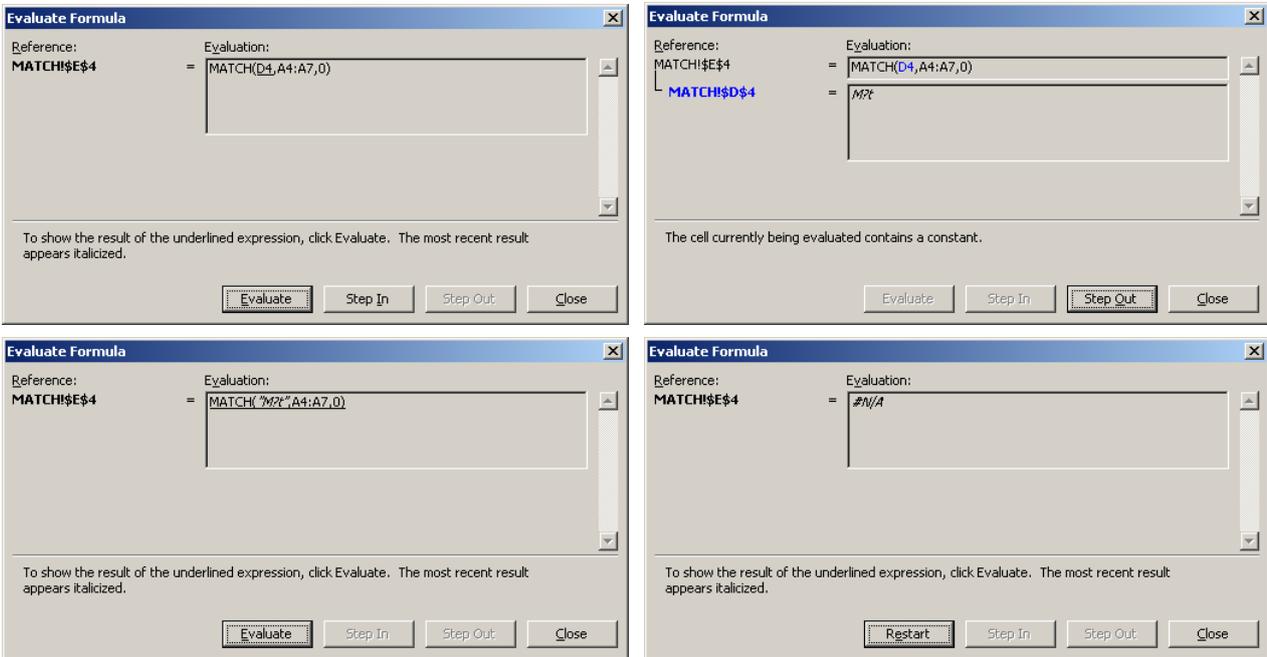
Choose **Formulas** tab, **Error Checking** group, and select **Error Checking** to quickly find any error values in the current worksheet and display the **Error Checking** dialog box. The first erroneous cell in the worksheet is selected and its contents are displayed in the dialog box, along with a suggestion about the nature of the problem. When your problem appears in the dialog box, the following selections are available:

- **Help on this Error** displays a Help topic relating to the problem cell.
- **Show Calculation Steps** displays the Evaluate Formula dialog box.
- **Ignore Error** skips over the selected cell. To reset the ignored errors, click **[Options]** button and then click **[Reset Ignored Errors]** button.
- **Edit in Formula Bar** opens the selected cell in the formula bar for editing. When you're finished, click **[Resume]** button.



3.3 Evaluating and Auditing Formulas

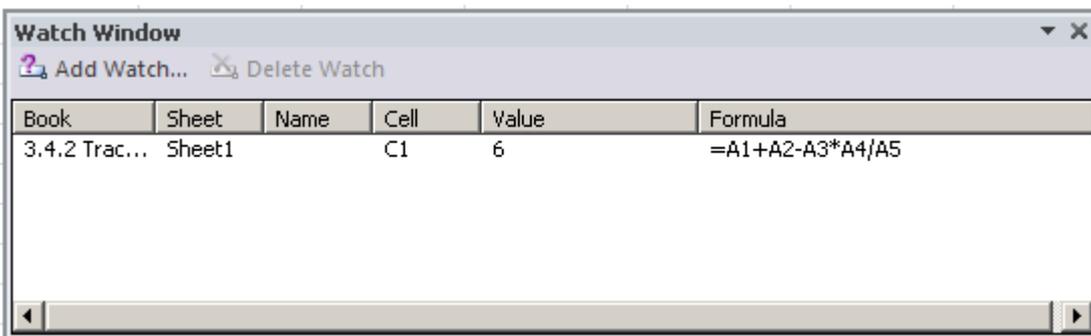
Sometimes it's difficult to tell what's going on in a complex nested formula. When you choose **Formulas** tab, select **Evaluate Formula**, you can check complex formulas easily. Click **[Evaluate]** button to replace calculable arguments with their resulting values. You can click Evaluate as many times as necessary if your formula contains many nested levels.



3.4 Watching Formulas

Sometimes you might want to keep an eye on a formula as you make changes to other parts of a worksheet, or even when working on other workbooks that supply information to a worksheet. Instead of constantly having to return to the formula's location to see the results of your ministrations, you can use the Watch Window, which provides remote viewing for any cell on any open worksheet.

Select a cell you want to keep an eye on and choose **Formulas** tab, **Watch Window**. Then select the cell and click **[Add Watch]** button in the **Add Watch** window.



3.5 Tracing Cell References

You can also use cell tracers to help find the source of those pesky errors that occasionally appear in your worksheets.

3.5.1 Understanding Precedents and Dependents

The terms precedent and dependent crop up quite often in this section. They refer to the relationships that cells containing formulas create with other cells. A lot of what a spreadsheet is all about is wrapped up in these concepts, so here's a brief description of each term:

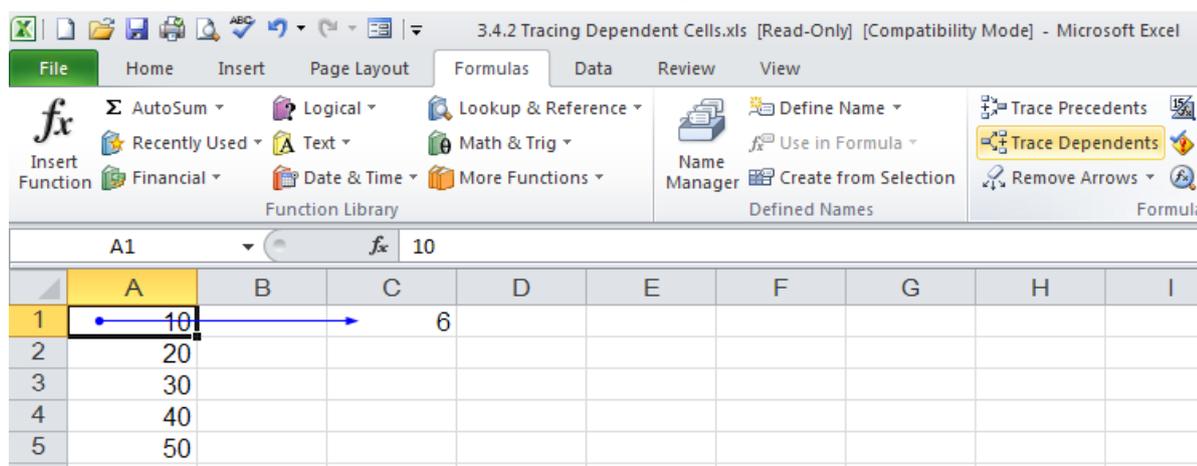
- Precedents are cells whose values are used by the formula in the selected cell. A cell that has precedents always contains a formula.
- Dependents are cells that use the value in the selected cell. A cell that has dependents can contain either a formula or a constant value.

For example, if the formula =SUM(A1:A5) is in cell A6, cell A6 has precedents (A1:A5) but no apparent dependents. Cell A1 has a dependent (A6), but no apparent precedents. A cell can be both a precedent and a dependent if the cell contains a formula and is also referenced by another formula.

3.5.2 Tracing Dependent Cells

To find out which cells contain formulas that use this value, you can click the [**Trace Dependents**] button on the **Formula Auditing** toolbar. Although this worksheet is elementary, to make it easier to illustrate the cell tracers, consider the ramifications of using the cell tracers in a large and complex worksheet.

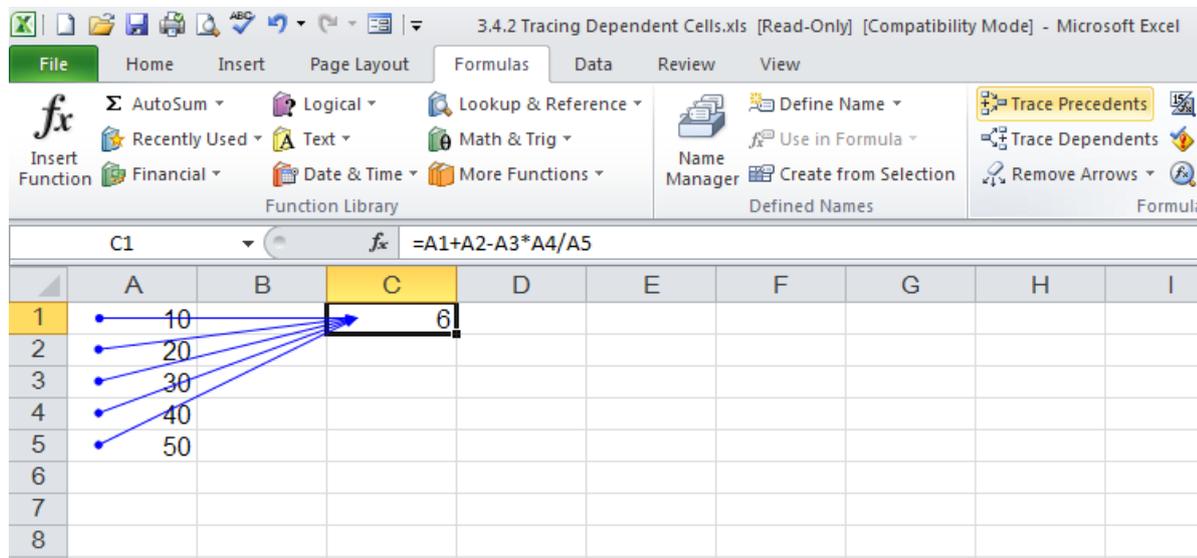
The tracer arrows indicate that cell is directly referred to by the formulas in another cells. The dot appears in the cell indicating that it has dependents. If you click Trace Dependents again, another set of arrows appears, indicating the next level of dependencies or indirect dependents.



3.5.3 Tracing Precedent Cells

You can also trace in the opposite direction by starting from a cell that contains a formula and tracing the cells that are referred to in the formula. To find out which cells this formula refers to, we clicked [**Trace Precedents**] button.

The dots identify these cells as precedents in the data flow. Notice that the arrow still points in the same direction (toward the formula and in the direction of the data flow) even though we started from the opposite end of the path. To continue the trace, click the [**Trace Precedents**] button again.



3.5.4 Clearing Tracer Arrows

Each time you trace another cell's precedents or dependents, additional tracer arrows appear. It's a good idea to start fresh each time you want to trace cells. To remove all the tracer arrows from the screen, click the [**Remove All Arrows**] button on the **Formula Auditing** toolbar.

3.6 Adding Comments to Cells

You can attach comments to cells to document your work, explain calculations and assumptions, or provide reminders. Select the cell you want to annotate and click the [**New Comment**] button on the **Formula Auditing** toolbar. Then type your message in the comment box that appears.

	A	B	C	D	E
1					
2	Hourly Rate	42.55			
3					
4	<i>Name</i>	<i>Hours Worked</i>	<i>Wages Due</i>	<i>Taxes Withheld</i>	<i>Net Wages</i>
5	Carothers, Andy	27	\$1,149	\$230	\$919
6	Hay, Jeff	32			\$1,089
7	Ito, Shu	40			\$1,362
8	Johnson, Willis	29			\$987
9					
10					
11					

Mark Dodge:
We can approve up to 50 hours per week for Shu.

4. Working with External Data

You can compile your data into Excel by retyping or by copying and pasting. But importing is most effective when you're working with large amounts of data that would be time-consuming to enter manually or too large to copy and paste. In addition, if you import your data, Excel can automatically update your reports and summaries whenever the original source database is updated.

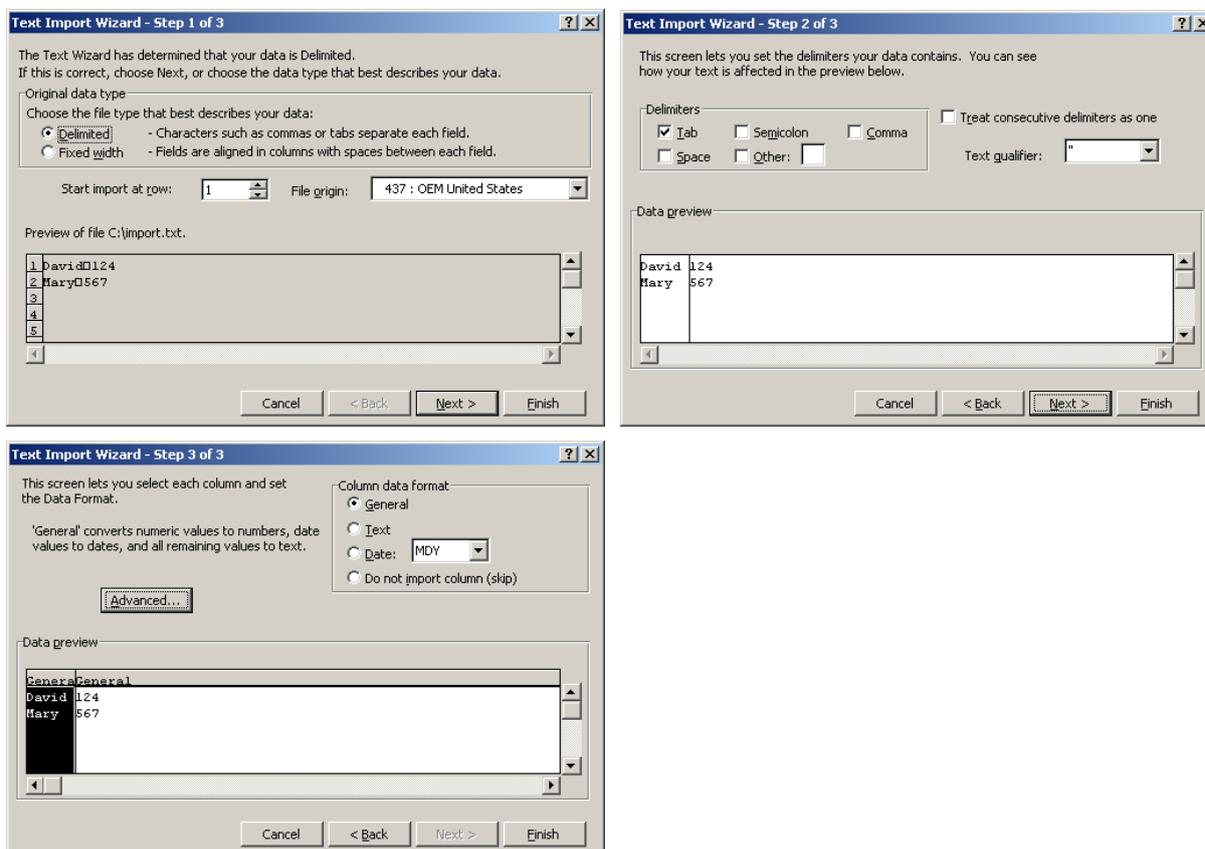
4.1 Import External Data Files

The main benefit of connecting to external data is that you can periodically analyze this data in Excel without repeatedly copying the data, which is an operation that can be time-consuming and error-prone. After connecting to external data, you can also automatically refresh your Excel workbooks from the original data source whenever the data source is updated with new information.

4.1.1 Import a Text File

In order to import a text file in fixed-length or tab-delimiter format, follow these steps:

1. Click the cell you want to put the data and then select **Data** tab, **Get External Data** group, **From Text**.
2. Locate the file you want to imported
3. To specify how you want to divide the text into columns, follow the instructions in the **Text Import Wizard**, and then click [**Finish**].



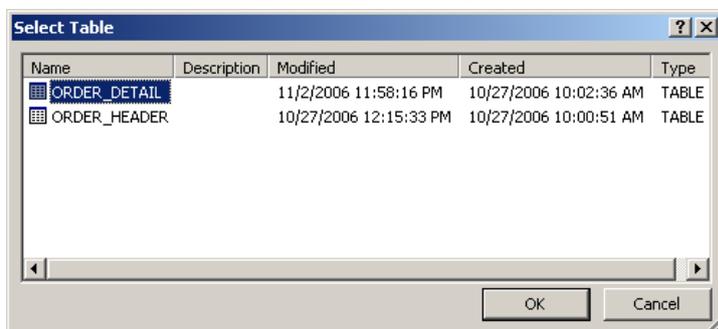
4. In the **Import Data** dialog box, click **Properties** to set formatting and layout options for the imported data.
5. In the **Import Data** dialog box, do one of the following:
 - To return the data to the location you selected, select **Existing worksheet**, and then click **[OK]**.
 - To return the data to a new worksheet, select **New worksheet**, and then click **[OK]**. Excel adds a new worksheet to your workbook and automatically puts the external data range in the upper-left corner of the new worksheet.



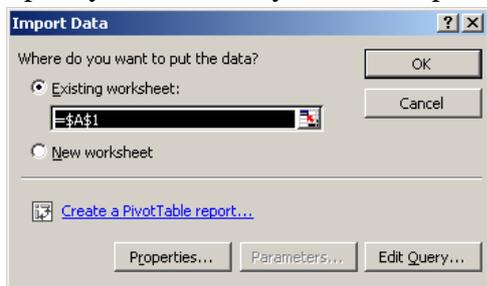
4.1.2 Import Microsoft Access Tables

In order to import a Microsoft Access table, follow these steps:

1. Click the cell you want to put the data and then select **Data** tab, **Get External Data** group, **From Access**.
2. Locate the file you want to imported
3. Specify which table you wish to import in the **Select Table** dialog, and then click **[OK]**.



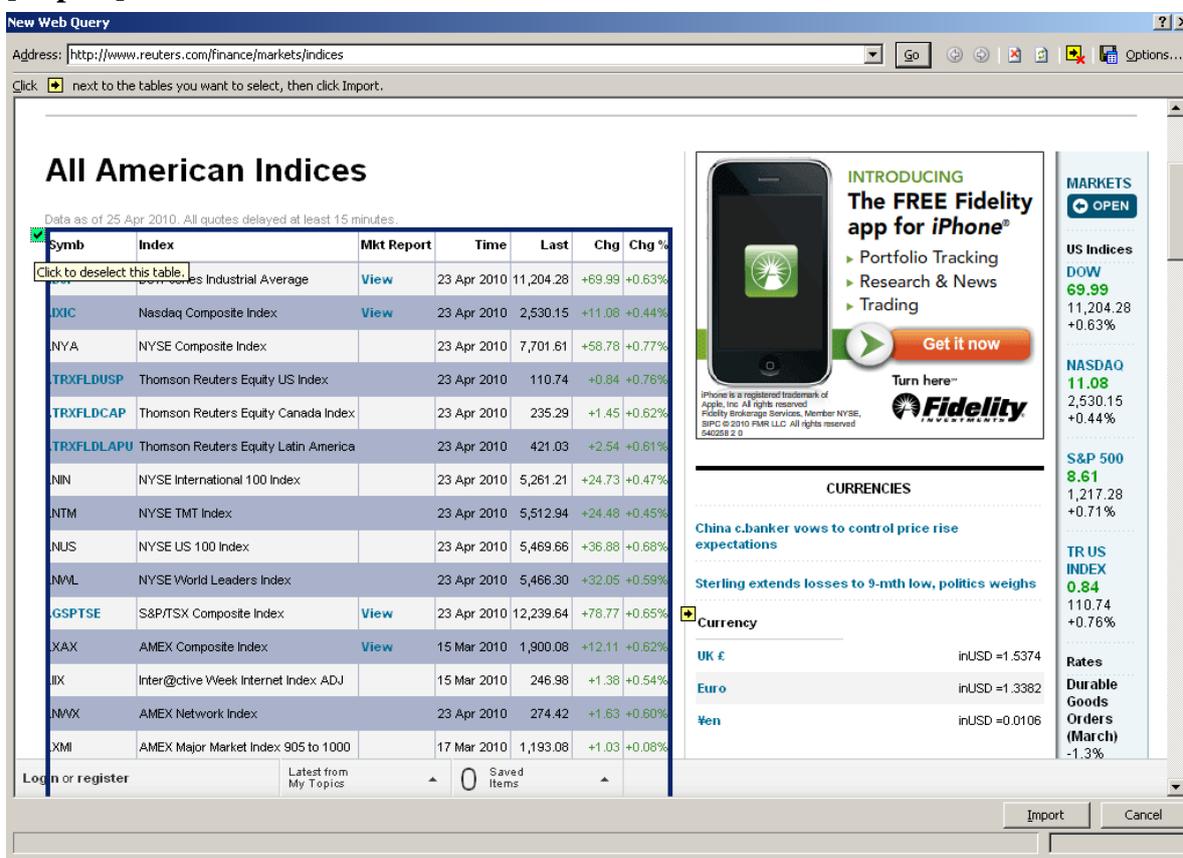
4. Specify the location you want to place the import data and click **[OK]** to finish.



4.2 Using a Web Query to Return Internet Data

Web queries can assist you in inserting refreshable data from the Web into your worksheets. Web queries were available in earlier versions of Excel, but have been greatly improved. You can now navigate to any Web page while setting up a Web query, the same way as you would navigate to that page in your browser. Once you are at the page, you can select one or more tables by clicking the icon images automatically added to the page by Excel

1. Click the cell you want to put the data and then select **Data** tab, Get **External Data** group, and choose **From Web**.
2. Enter the URL in the **Address** box of the **New Web Query** windows and press the **[Go]** button to retrieve the target web page.
3. Select the data area you wish to import from the **New Web Query** windows, and then press **[Import]** button to continue..



4. Specify the location you want to place the import data and click **[OK]** to finish.



4.3 Refresh Imported Data

Excel provides many options for refreshing imported data, including refreshing the data whenever you open the workbook and automatically refreshing data at timed intervals. You can continue to work in Excel while data is being refreshed, and you can also check the status of the refresh while it's being refreshed.

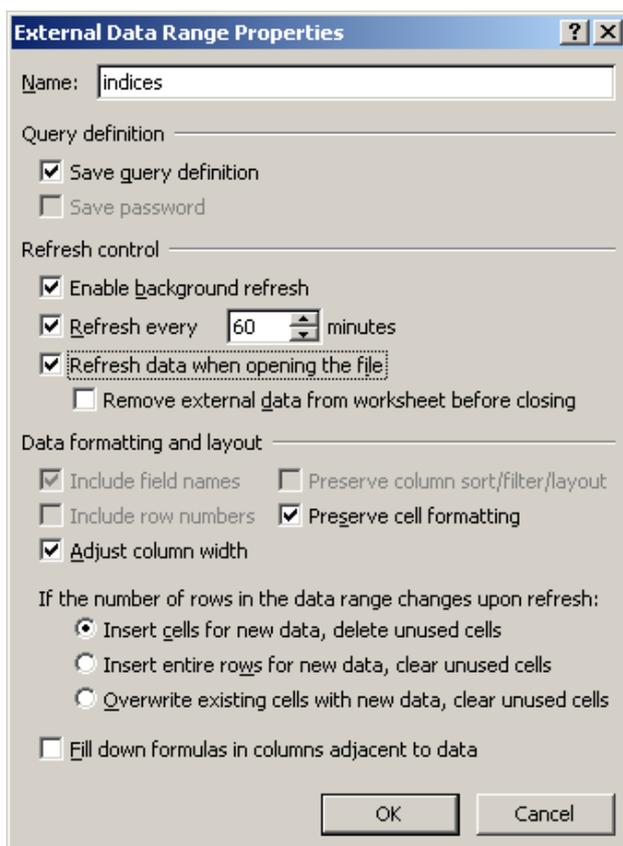
If your external data source requires a password to gain access to the data, you can require that the password be entered each time the external data range is refreshed.

When an external data range expands and additional records are returned, Excel can fill formulas in adjacent columns or within the data range so that they remain next to the appropriate data.

4.3.1 Automatically Refresh Data when a Workbook is Opened

You can refresh an external data range automatically when you open the workbook, and optionally save the workbook without saving the external data, so that the workbook file size is reduced.

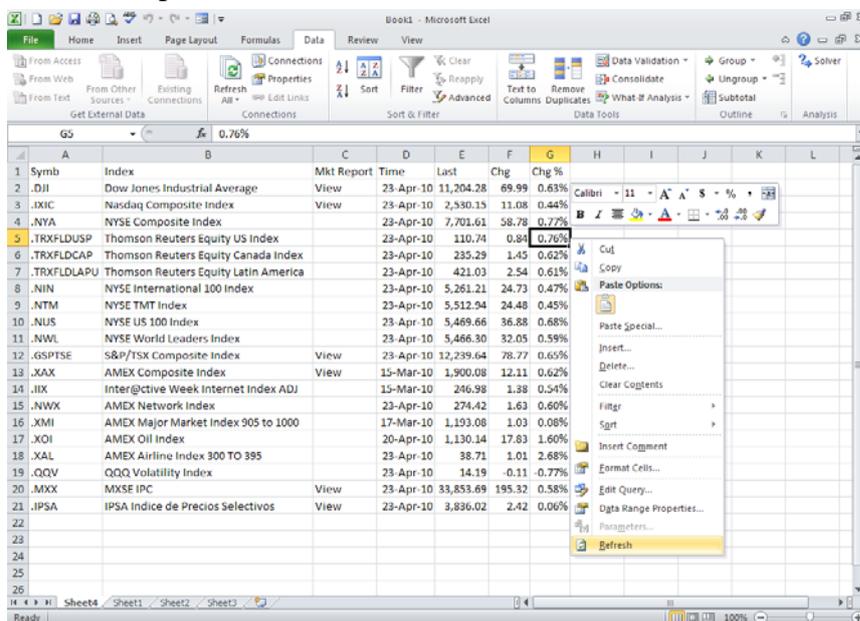
1. Right click the imported external data, and select **Data Range Properties**.
2. Select the **Refresh data when opening the file** check box in the **External Data Range Properties** dialog box.



3. If you want to save the workbook with the query definition but without the external data, select the **Remove external data from worksheet before closing** check box.

4.3.2 Refresh Data in an Imported File

1. If your worksheet contains more than one external data range that you want to refresh, right click the imported data and click **Refresh**.



4.3.3 Require a Password to Refresh an External Data Range

If your data source requires a password to connect to it, you can require that the password is entered before the external data range can be refreshed. This procedure does not apply to data retrieved from a text file (*.txt) or a Web query (*.iqy). Please note that the stored passwords are not encrypted. Moreover, Excel prompts for the password only the first time that the external data range is refreshed in each Excel session. The next time you start Excel, you will be prompted for the password again if you open the workbook that contains the query, and then click **Refresh Data**

1. Right click the imported external data, and select **Data Range Properties**.
2. Clear the **Save password** check box under **Query definition** section.

4.3.4 Preserve Cell Formatting when you Refresh an External Data Range

1. Right click the imported external data, and select **Data Range Properties**.
2. Do one or both of the following under **Data formatting and layout** section:
 - If you want to preserve cell formatting that you apply, select the **Preserve cell formatting** check box.
 - If you want to preserve column widths that you set, clear the **Adjust column width** check box.
3. Click [**OK**].
4. To refresh the external data range, right click the imported data and click **Refresh**.

4.3.5 Run a Query in the Background or While you Wait

Running a query in the background allows you to use Microsoft Excel while the query runs.

1. Right click the imported external data, and select **Data Range Properties**.
2. Select the **Enable background refresh** check box to run the query in the background or clear it to run the query while you wait.

4.3.5.1 Stop a Query

To stop a query from running when **Enable background refresh** is turned off, press [ESC] button.

4.3.5.2 Stop a Background Refresh

To stop a query that is running in the background, double-click the **Background Refresh**  icon on the status bar to display the **External Data Refresh Status** dialog box, and then click [Stop Refresh].

4.3.5.3 Recording a Macro

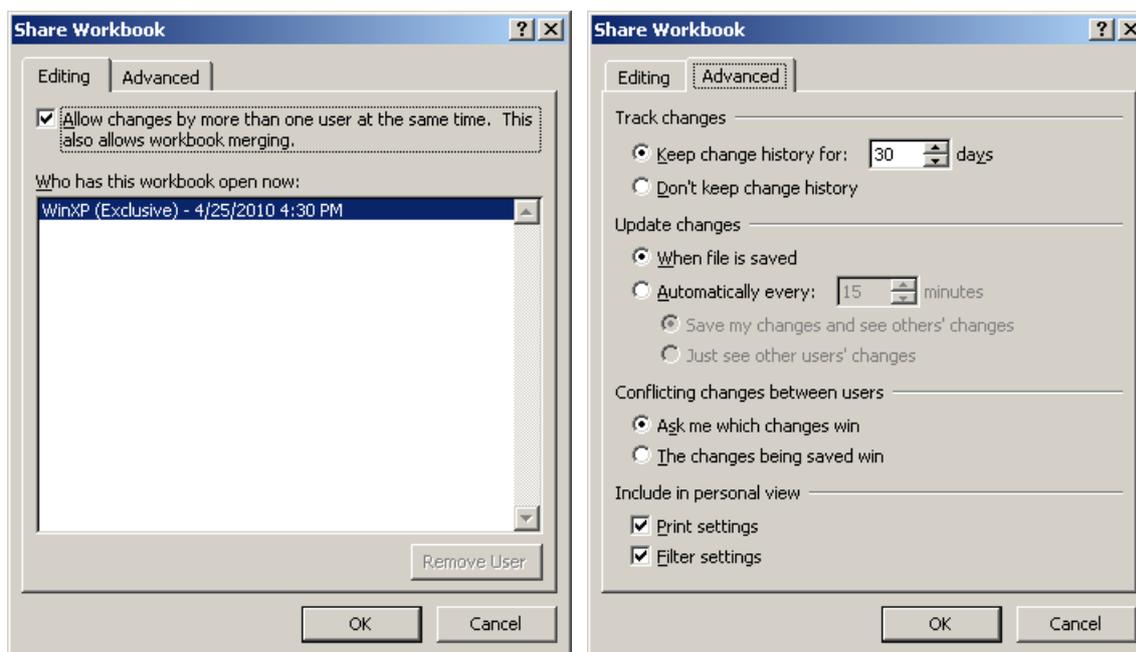
While you are recording a macro that includes a query, Excel will not run the query in the background. To change the recorded macro so that it runs in the background, edit the macro in the Visual Basic Editor and change the refresh method for the **QueryTable** object from "**BackgroundQuery := False**" to "**BackgroundQuery := True**".

5. Collaboration and Security

5.1 Sharing Workbook

It has always been possible to share Excel files on a network. You just had to make sure that you coordinated your efforts to avoid having more than one person open a file at the same time. Recent versions of Excel, however, allow two or more people to work on the same workbook simultaneously,

1. Select **Review** tab, **Change** group, **Share Workbook** to popup the **Share Workbook** dialog,
2. Only one user is allowed to change the workbook by default. You can change some aspects of the default behavior of shared workbooks. Each shared workbook user can set these options individually. Use the first section on the **Advanced** tab to specify the length of time you want to keep track of changes, or whether you want to track them at all.
3. Save your workbook.



5.2 Tracking

Excel can maintain and display information about how a worksheet was changed. Change tracking logs details about workbook changes each time you save a workbook. You can use this history to understand what changes were made, and to accept or reject revisions.

This capability is particularly useful when several users edit a workbook. It's also useful when you submit a workbook to reviewers for comments, and then want to merge input into one copy, selecting which changes and comments to keep

5.2.1 Tracking Changes

Change tracking in Excel is closely linked with shared workbooks.

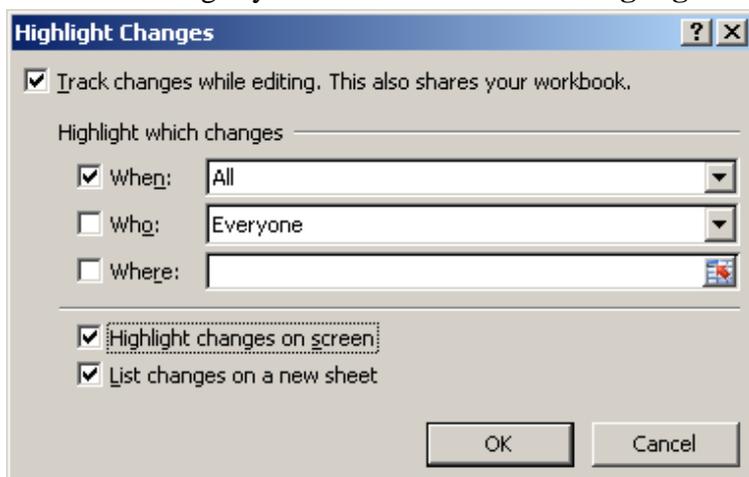
1. Select **Review** tab, **Changes** group, **Protect and Share Workbook**.
2. Select **Sharing with track changes** check box in the **Protect Share Workbook** dialog.
3. If you click **Sharing with track changes** and then click **[OK]**, change tracking for the shared workbook is protected so no one in your workgroup can turn it off directly. However, anyone can turn off the protection. To eliminate this possibility, you can enter a password in the **Protect Shared Workbook** dialog box. But you must do this when the workbook is not in shared mode. Then anyone who tries to turn off protection must enter the identical, case-sensitive password.



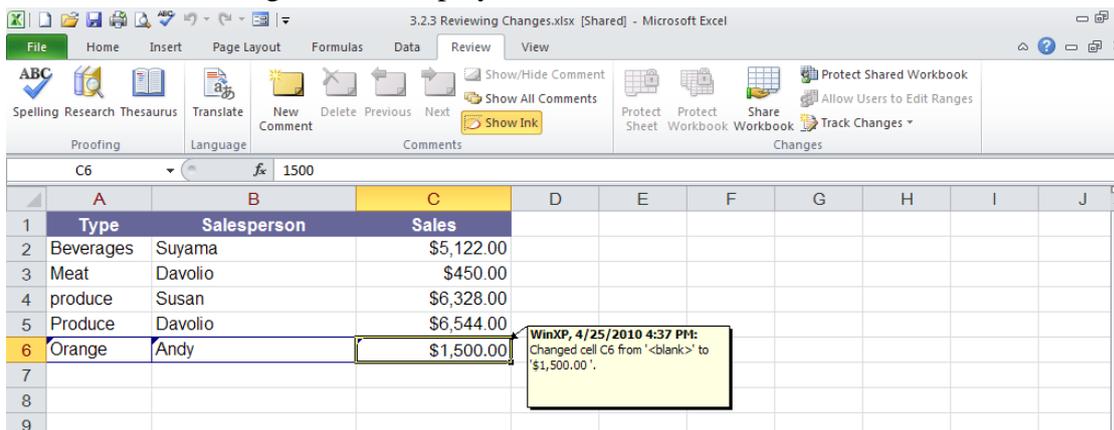
5.2.2 Highlight Changes

You can decide at any time to go through each change that has been made to the shared workbook, provided the Track Changes While Editing check box was selected in the Highlight Changes dialog box when the worksheet was first saved for sharing.

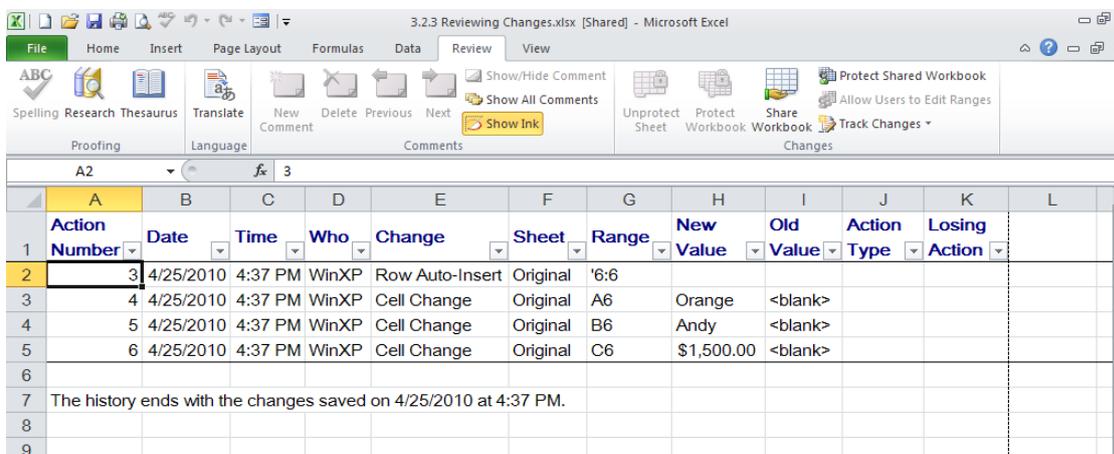
1. Select the cell you wish to review, and then choose **Review** tab, **Changes** group, **Track Changes** and its sub item **Highlight Changes**
2. Select the changes you wish to review in the **Highlight Change** dialog box.



3. The list for the change will be displayed in color for indication.

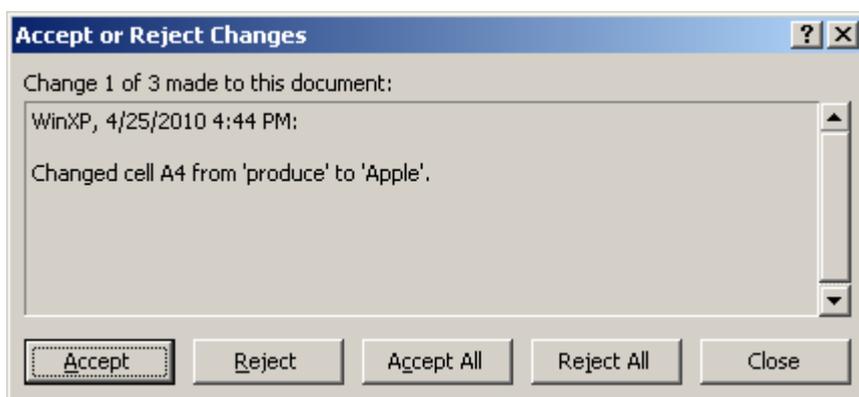


4. If the **List change on a new chance** checkbox is select, the selected changes will be displayed in a report in a new worksheet.



5.2.3 Accept or Reject Changes

1. Select the cell you wish to review, and then choose Select **Review** tab, **Changes** group, Select **Track Changes** and its sub item **Accept/Reject Changes**.
2. Then you can accept or reject each change one by one.



5.3 Protecting Files

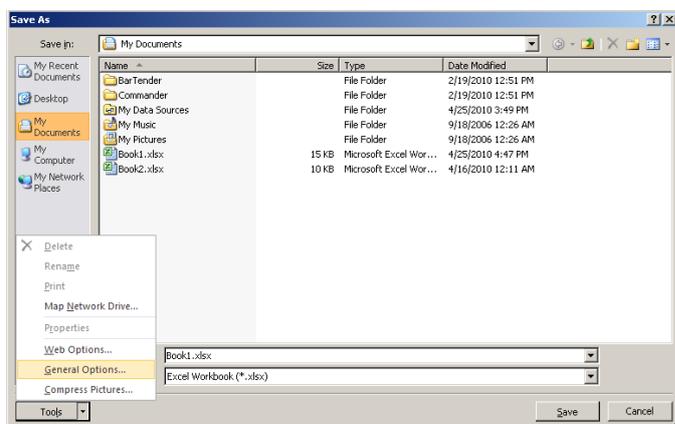
Excel features related to hiding data or locking data with passwords are not intended to secure or protect confidential information in Excel. These features are merely meant to obscure data or formulas that might confuse some users or to prevent others from viewing or making changes to that data. Excel does not encrypt data that is hidden or locked in a workbook. To help prevent modification of confidential data and to help protect it from being viewed, you may want to limit access to any workbook files that contain such information by storing them in locations that are available only to authorized users. Excel provides several layers of security and protection to control who can access and change your data:

- For optimal security, you should protect your entire workbook file with a password, allowing only authorized users to view or modify your data.
- For additional protection of specific data, you can protect certain worksheet or workbook elements, with or without a password. Use element protection to help prevent anyone from accidentally or deliberately changing, moving, or deleting important data.

5.3.1 Securing a Workbook File

Password security at the workbook file level uses advanced encryption to help protect your workbook from unauthorized access.

1. Select **File** → **Save As**, and then select **Tools** → **General Options** in the **Save As** dialog.



2. You can specify two separate passwords that users must type to and then save the workbook.
 - **Open and view the file** – This password is encrypted to help protect your data from unauthorized access.
 - **Modify the file** – This password is not encrypted and is only meant to give specific users permission to edit workbook data and save changes to the file.

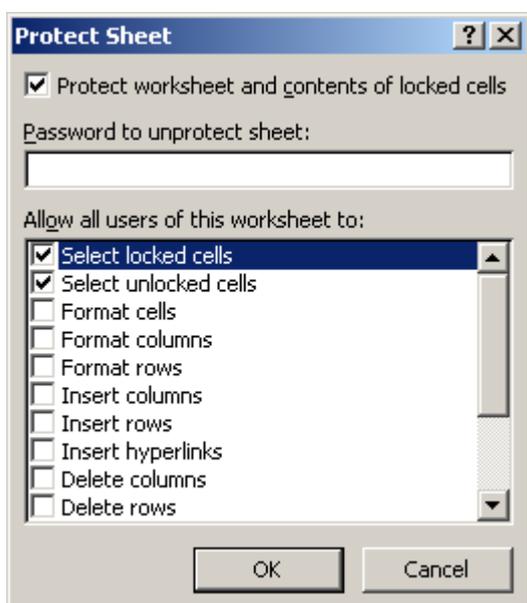


5.3.2 Protecting Specific Worksheet or Workbook Elements

When you share an Excel file so that others can collaborate on the data, you can prevent any user from making changes to specific worksheet or workbook elements by protecting (or locking down) certain parts of the file. You can also specify a password to allow individual users to modify specific elements.

When you protect a worksheet, all cells on the worksheet are locked by default, and users cannot make any changes to a locked cell. For example, they cannot insert, modify, delete, or format data in a locked cell. You can, however, specify which elements users will be allowed to change when you protect the worksheet.

1. To protect a worksheet, select **Review** group, **Change Tab**, **Protect Sheet**. You can control access to individual worksheet or chart sheet elements by selecting or clearing the following check boxes



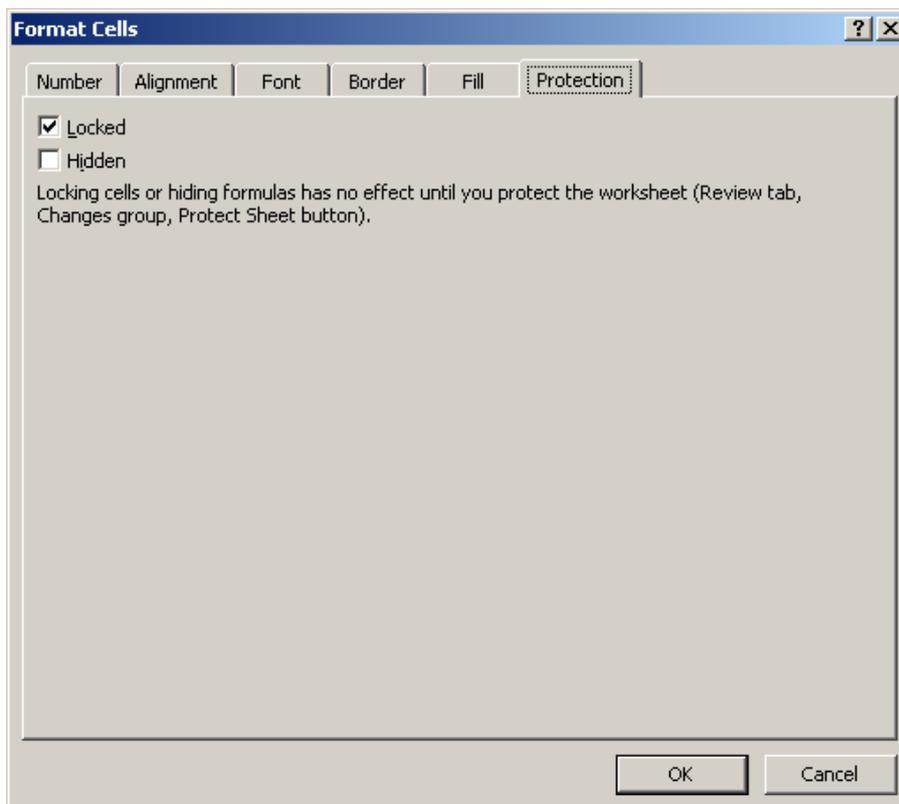
2. To protect a workbook, select **Review** group, **Change Tab**, **Protect Workbook**. You can control access to entity workbook by selecting or clearing the following check boxes.



5.3.3 Permission to Access Specific Areas of a Protected Worksheet

Before you protect a worksheet, you can unlock the ranges that you want users to be able to change or enter data in. You can:

- Unlock cells for all users on the Protection tab of the Format Cells dialog box by choosing **Format → Cells**.



- Unlock cells for specific users in the **Allow Users to Edit Ranges** dialog box by choosing select **Review** group, **Change Tab, Allow Users to Edit Ranges**.



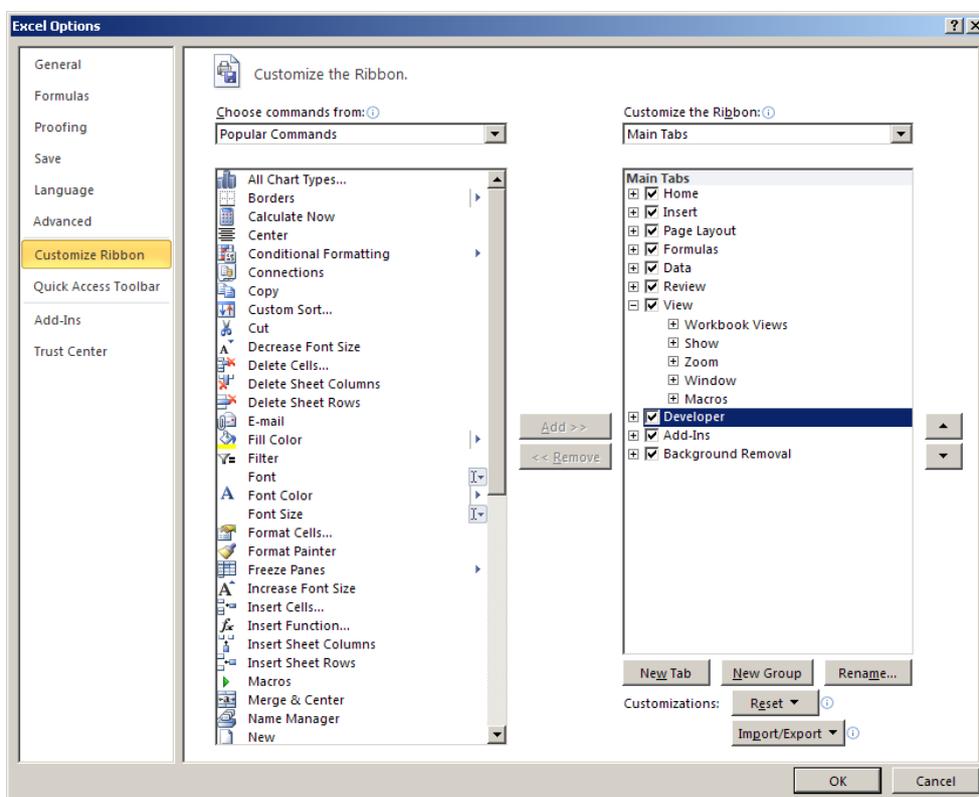
Users whom you specify in the Permissions for range dialog box (Permissions button) can automatically edit the range without entering the password. All other users are prompted for the password when they want to edit the range.

6. Macro and VBA

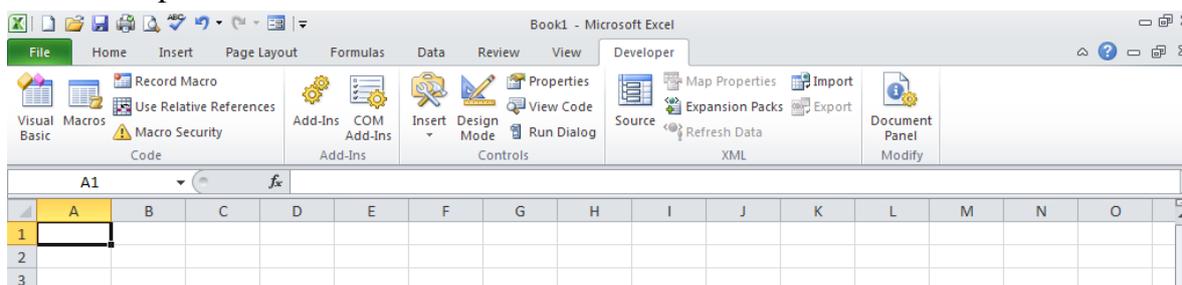
If you perform a task repeatedly in Microsoft Excel, you can automate the task with a macro. A macro is a series of commands and functions that are stored in a Microsoft Visual Basic module and can be run whenever you need to perform the task. For example, if you often enter long text strings in cells, you can create a macro to format those cells so that the text wraps.

6.1 Enable Developer Tab in Ribbon

1. Click the **Microsoft Office Button**, and then click **Excel Options**.
2. Click **Popular**, and then select the **Show Developer** tab in the **Ribbon** check box, and press **[OK]** to confirm.



3. The Developer tab will be enable to use.

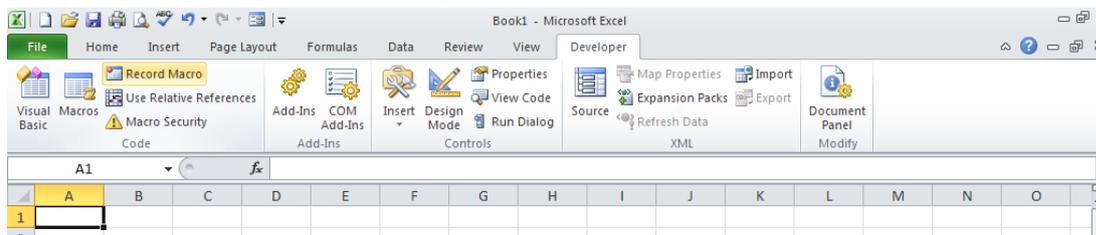


6.2 Record a Macro

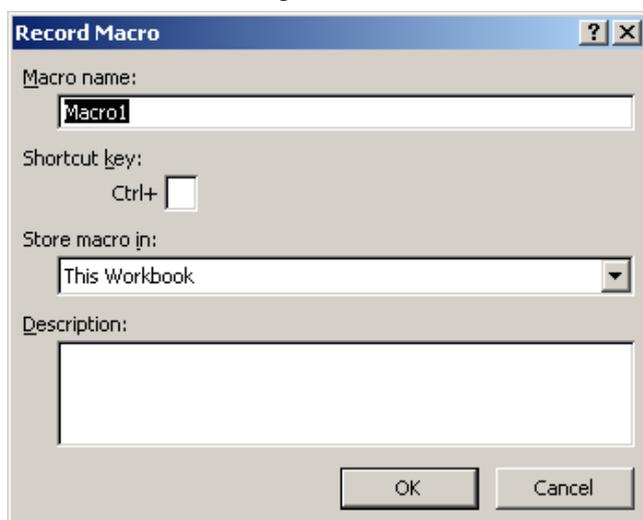
6.2.1 Record a Macro using Macro Recorder

When you record a macro, Excel stores information about each step you take as you perform a series of commands. You then run the macro to repeat, or play back, the commands. If you make a mistake when you record the macro, corrections you make are also recorded. Visual Basic stores each macro in a new module attached to a workbook.

1. Select **Developer** tab, **Code Group**, **Record Macros**



2. Perform the following action in the **Record Macro** dialog box.



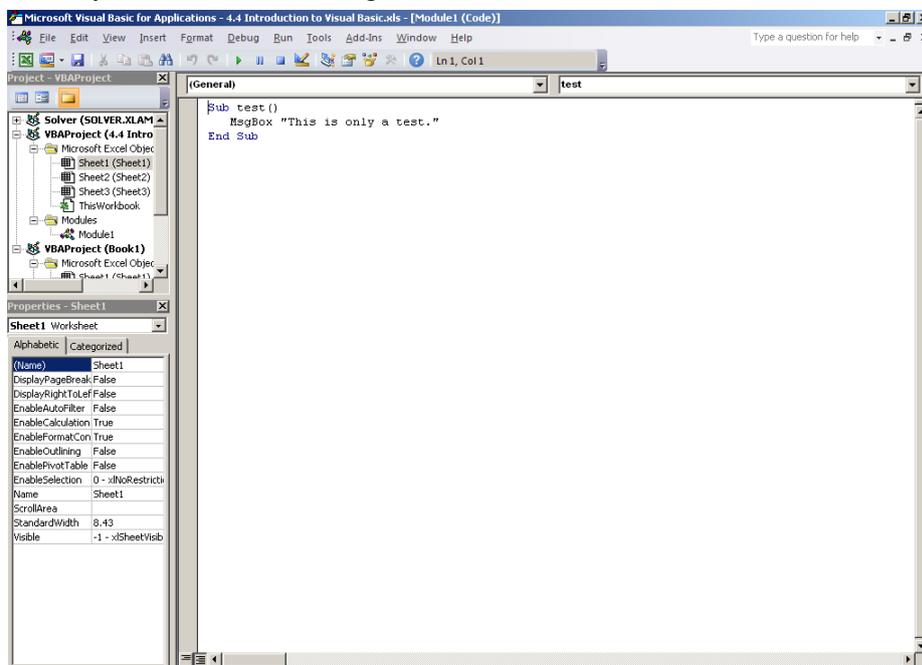
- In the **Record Macro** dialog box, enter a name in the **Macro Name** box
- If you want to run the macro by pressing a keyboard shortcut key, enter a letter in the **Shortcut** key box. The shortcut key will override any equivalent default Excel shortcut keys while the workbook that contains the macro is open
- In the **Store macro in** box, click the location where you want to store the macro. If you want a macro to be available whenever you use Excel, select **Personal Macro Workbook**.
- If you want to include a description of the macro, type it in the **Description** box.
- Click **[OK]**. If you want the macro to run relative to the position of the active cell, record it using relative cell references. On the **Stop Recording** toolbar, click **Relative Reference** so that it is selected. Excel will continue to record macros with relative references until you quit Excel or until you click **Relative Reference** again, so that it is not selected.

3. Carry out the actions you want to record.

4. Select **View** tab, **Code Group**, **Stop Macro** when finish the record.

6.2.2 Create a macro using Visual Basic Editor

1. Select **Developer** tab, **Code Group**, **Visual Basic**
2. Select the **Insert** → **Module** in the **Microsoft Visual Basic Editor**.
3. Type or copy your code into the code window of the module.
4. If you want to run the macro from the module window, press **[F5]**.
5. When you're finished editing, click **File** → **Close and Return to Microsoft Excel**.

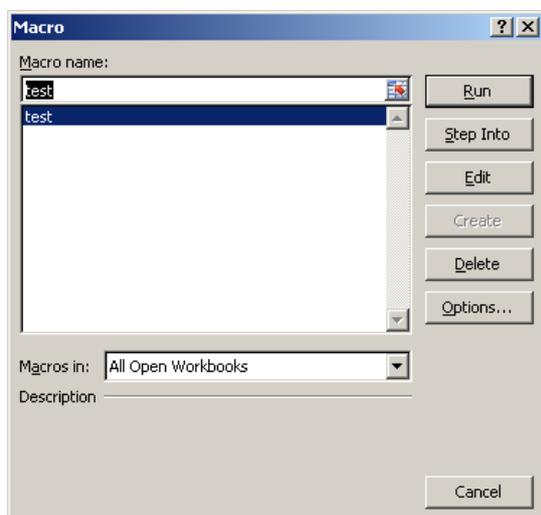


6.3 Execute the Macro

The next time you need to flag a cell, you can run the macro. If you're going to use the macro frequently, you can create a toolbar button for it, or assign a keystroke for it, or both.

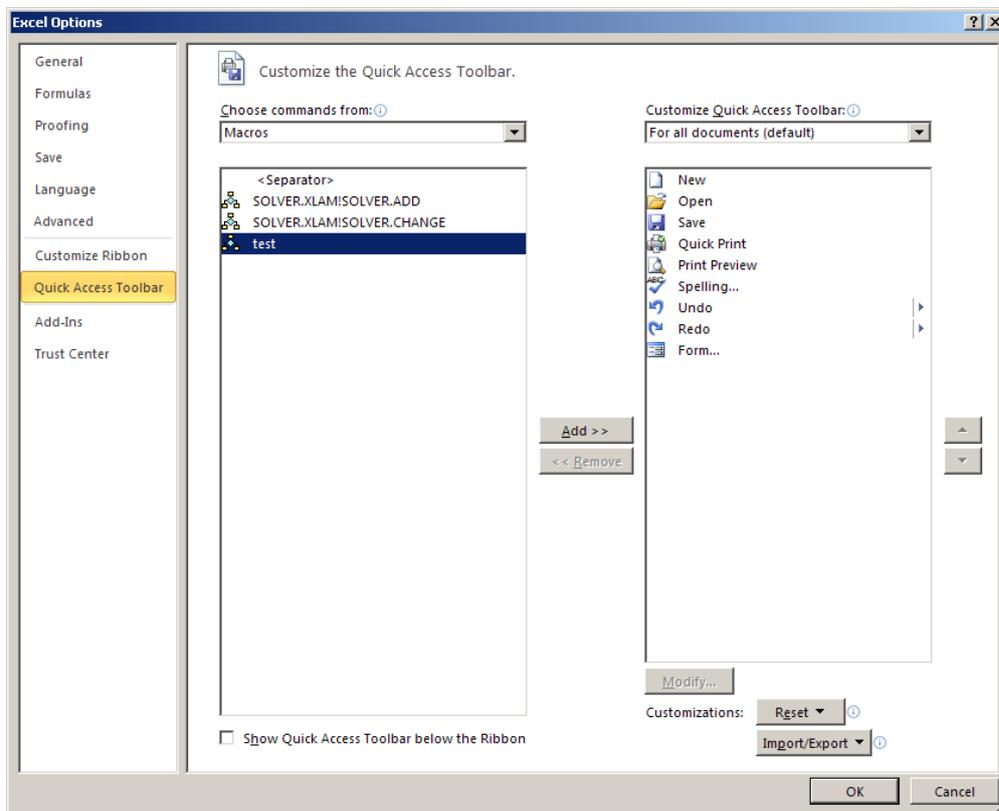
6.3.1 Run your Macro Using the Tools Menu

1. Select **View** tab, **Code Group**, **Macros** to call the **Macro** dialog box.
2. Click the name of your macro, and then click **Run**.



6.3.2 Create a Toolbar Button to Run your Macro

1. Click the Microsoft Office Button, and then click **Excel Options**, and in the **Quick Access Toolbar** tab.
2. Select **Macros** in the **Choose command from**
3. Select the macro you want to assign and press the **Add** button, and then click **[OK]**.



6.3.3 Assign a Keystroke to Run your Macro

1. Click the worksheet, and then select **Developer** tab, **Code Group**, **Macro**.
2. Select the name of your macro, and then click **[Options]**.
3. In the Shortcut key box, type the key to use along with **[Ctrl]** button to run your macro.

6.4 Managing your Macros

After you record a macro, you can view the macro code with the Visual Basic Editor to correct errors or change what the macro does. For example, if you wanted the text-wrapping macro to also make the text bold, you could record another macro to make a cell bold and then copy the instructions from that macro to the text-wrapping macro.

The Visual Basic Editor is a program designed to make writing and editing macro code easy for beginners, and provides plenty of online Help. You don't have to learn how to program or use the Visual Basic language to make simple changes to your macros. With the Visual Basic Editor, you can edit macros, copy macros from one module to another, copy macros between different workbooks, rename the modules that store the macros, or rename the macros.

6.5 Introduction to Visual Basic Editor

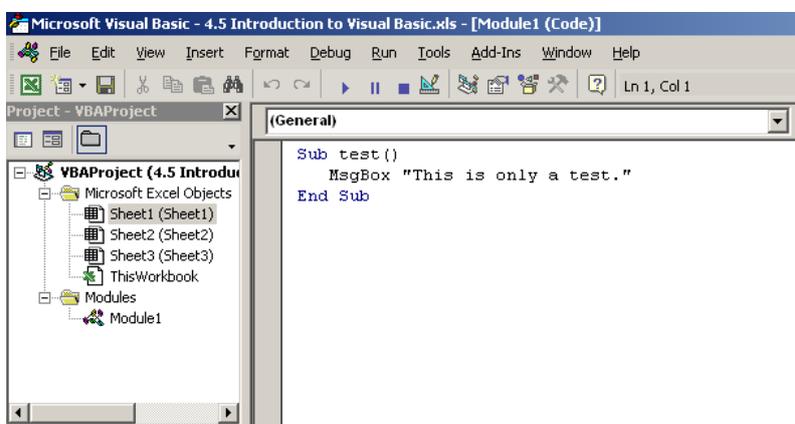
6.5.1 Using the Visual Basic Editor

The Visual Basic Editor is a powerful tool that lets you extend the power and versatility of macros beyond anything that can be done through recording alone.

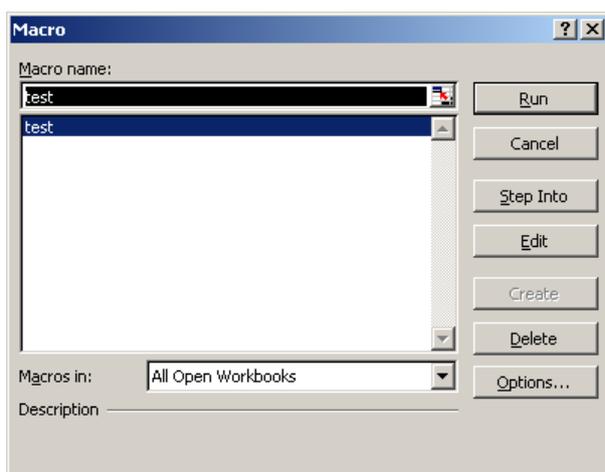
1. Start Excel and open a new, blank workbook.
2. Select **Developer** tab, **Code Group**, **Visual Basic**.
3. In the **Project** window, double-click **ThisWorkbook**.
4. Enter this code into the code window:

```
sub test()  
    MsgBox "This is only a test."  
end sub
```

5. Save the file and then select **File** → **Visual Basic Editor** and close the workbook.



6. Select **Developer** tab, **Code Group**, **Macros** in the Excel worksheet. Then select the previous macro "text" we created in the Visual Basic Editor.



7. Press **[Run]** to execute the macro and a message box will be displayed..



6.6 Macro Security

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

6.7 Macro Security Settings and their Effects

The following list summarizes the various macro security settings. Under all settings, if antivirus software that works with 2007 Microsoft Office system is installed and the workbook contains macros, the workbook is scanned for known viruses before it is opened.

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

6.8 Change Macro Security Settings

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

1. On the **Developer** tab, in the **Code** group, click **Macro Security**.
2. In the **Macro Settings** category, under **Macro Settings**, click the option that you want. Any changes that you make in the Macro Settings category in Excel apply only to Excel and do not affect any other Microsoft Office program.

