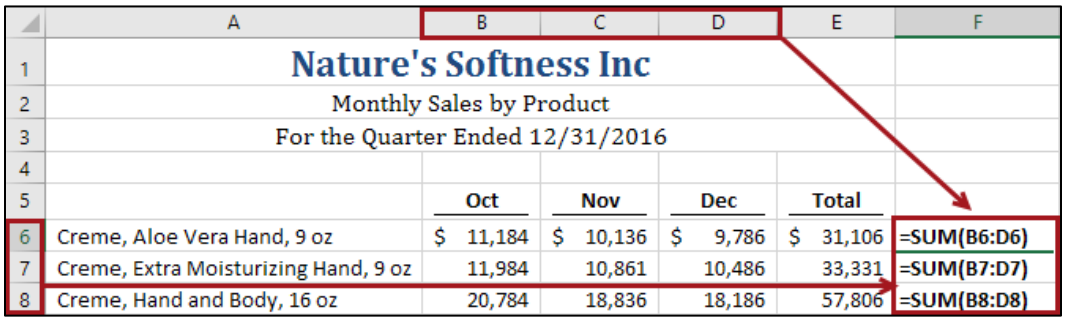
Formula Tips in Excel 2010/2016

Though it is advisable to keep the number of formulas in use to a minimum, formulas are a necessary component of most Excel workbooks. If you know which functions you should insert into formulas to achieve specific results, you are far more likely to be successful when working with Excel. Further, if you are familiar with some of the fundamentals of formulas such as absolute and relative cell referencing, you will help to minimize the likelihood of errors in your workbooks. In this webinar, we will explore these and other formula fundamentals.

Formula Fundamentals

Relative, Absolute, and Mixed Address Formulas

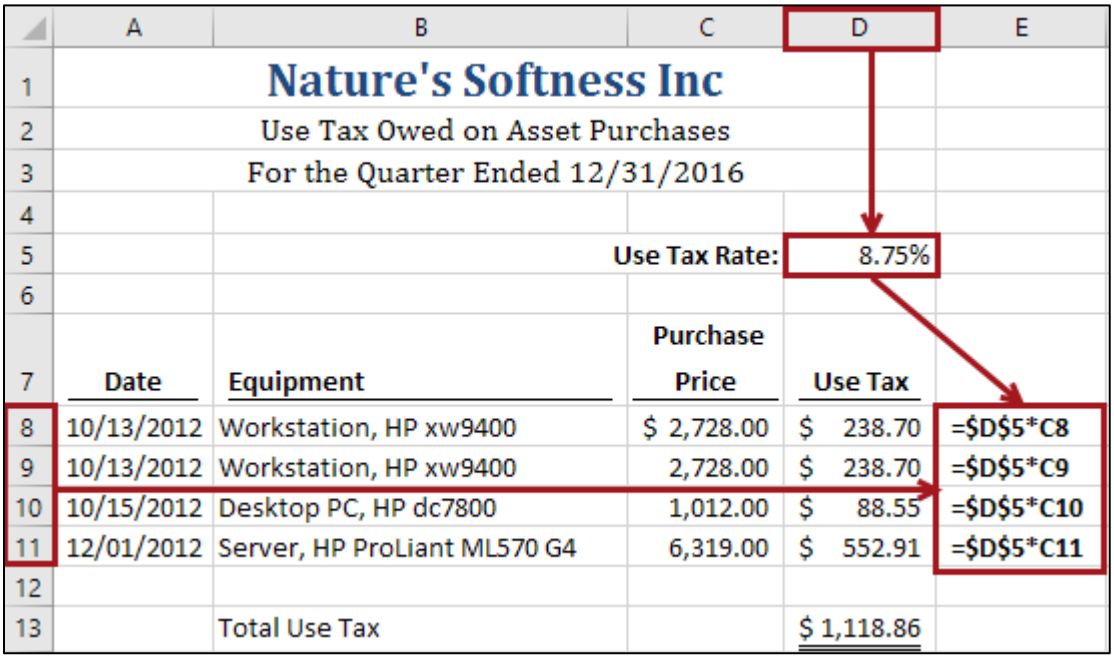
A full understanding of the rules of cell referencing is critical to building and troubleshooting formulas because the type of reference determines whether a formula can be copied successfully. Cell addresses can be relative, absolute, or mixed (partially relative and partially absolute). In the simplest case, cell addresses have two components – a column reference and a row reference. In default, cell addresses are *relative*. That means that the references to the column and row in the address are relative to a formula's current location. In summing monthly product sales, as shown in **Figure 85**, a simple formula using the **SUM** function has been entered in cell **E2**. The formula as entered was **=SUM(B2:D2)**, but it was automatically rewritten as it was copied down to reference rows 3, 4, and 5. The original and rewritten formulas are displayed in column F. Note how the row references in the cell addresses were rewritten to reference the row on which each formula was copied.



	A	B	C	D	E	F
1	Nature's Softness Inc					
2	Monthly Sales by Product					
3	For the Quarter Ended 12/31/2016					
4						
5		Oct	Nov	Dec	Total	
6	Creme, Aloe Vera Hand, 9 oz	$ 11,184	$ 10,136	$ 9,786	$ 31,106	=SUM(B6:D6)
7	Creme, Extra Moisturizing Hand, 9 oz	11,984	10,861	10,486	33,331	=SUM(B7:D7)
8	Creme, Hand and Body, 16 oz	20,784	18,836	18,186	57,806	=SUM(B8:D8)

Figure 1 - Relative Address Formulas Rewritten Automatically as Copied

An *absolute* address means that row and column references are *not* relative to a formula's current location but remain the same as the formula is copied. Excel uses dollar signs (**$**) to indicate whether a column or row reference is absolute. The cell address **C13** is relative, while the address **$C$13** is absolute, in this case, as to column and row. Use absolute addresses whenever a component of a formula must always point to the same cell. In computing quarterly use tax, as shown in **Figure 1**, a simple formula in cell **D4** is used to multiply the tax rate times the asset purchase price. The formula as entered is **=$D$1*C4**. The reference to **$D$1**indicates that the column and row references are absolute and will not change as the formula is copied. (The dollar signs may be typed in as the formula is entered, or you can press **F4** to enter the dollar signs automatically. Just press **F4** immediately after typing in a cell reference or pointing to a cell as a formula is created.) However, the reference to the cell **C4** containing an asset's purchase price was entered as a relative reference. As the formula is copied down, the cell reference pointing to the tax rate remains constant, but the cell reference pointing to an asset's purchase price is automatically rewritten to reference rows 5, 6, and 7. The original and rewritten formulas are displayed in column E. Note how the reference to cell **$D$1** remains constant, but the row references to the asset purchase price were rewritten to reference the row on which each formula was copied.

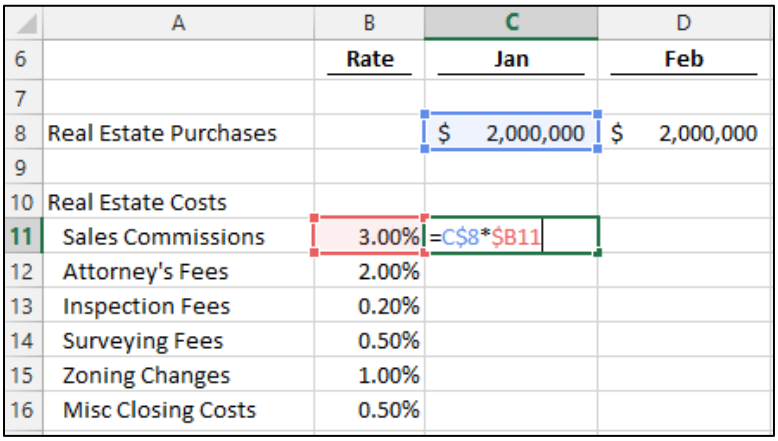


	A	B	C	D	E
1		**Nature's Softness Inc**			
2		Use Tax Owed on Asset Purchases			
3		For the Quarter Ended 12/31/2016			
4					
5			**Use Tax Rate:**	8.75%	
6					
7	**Date**	**Equipment**	**Purchase Price**	**Use Tax**	
8	10/13/2012	Workstation, HP xw9400	$ 2,728.00	$ 238.70	=$D$5*C8
9	10/13/2012	Workstation, HP xw9400	2,728.00	$ 238.70	=$D$5*C9
10	10/15/2012	Desktop PC, HP dc7800	1,012.00	$ 88.55	=$D$5*C10
11	12/01/2012	Server, HP ProLiant ML570 G4	6,319.00	$ 552.91	=$D$5*C11
12					
13		Total Use Tax		$ 1,118.86	

Figure 2 - Absolute Address Remaining Constant as a Formula is Copied

A *mixed* address means that the references in a cell address are *not* the same. One part, either row or column, is relative, and the other is absolute. The cell address **$C13** is absolute as to column and relative as to row, while the cell address **C$13** is relative as to column and absolute as to row. Use mixed addresses whenever a formula needs to be copied down *and* across.

In planning for real estate acquisitions, as shown in **Figure 2**, a single formula in cell **C11** computes ancillary real estate costs as a percentage of total planned real estate purchases. The formula as entered is **=C$8*$B11**. The reference to cell **C$8** indicates that the column is relative, and the row is absolute. As the formula is copied down and across, the formula will always point to row **8** relative to the column in which the formula is copied. The reference to cell **$B11** indicates that the column is absolute, and the row is relative. As the formula is copied down and across, the formula will always point to column **B** relative to the row in which the formula is copied. In this case, the use of a mixed address formula is more efficient and reduces the potential for error because a single formula is built, rather than six.



	A	B	C	D
6		**Rate**	**Jan**	**Feb**
7				
8	Real Estate Purchases		$ 2,000,000	$ 2,000,000
9				
10	Real Estate Costs			
11	Sales Commissions	3.00%	=C$8*$B11	
12	Attorney's Fees	2.00%		
13	Inspection Fees	0.20%		
14	Surveying Fees	0.50%		
15	Zoning Changes	1.00%		
16	Misc Closing Costs	0.50%		

Figure 3 - Building a Single Formula to Copy Down and Across

The Formula Bar

In default, the **Formula Bar** displays one line of the formula or text. However, note the scroll buttons and drop-down arrow at the right end of the Formula Bar. The scroll buttons allow you to scroll through the lines of a multiline formula or multiple lines of text, while the drop-down arrow displays up to three lines. When the drop-down arrow is used, the formula box expands so that the worksheet grid is not hidden by long, complex formulas

as shown in **Figure 4**.

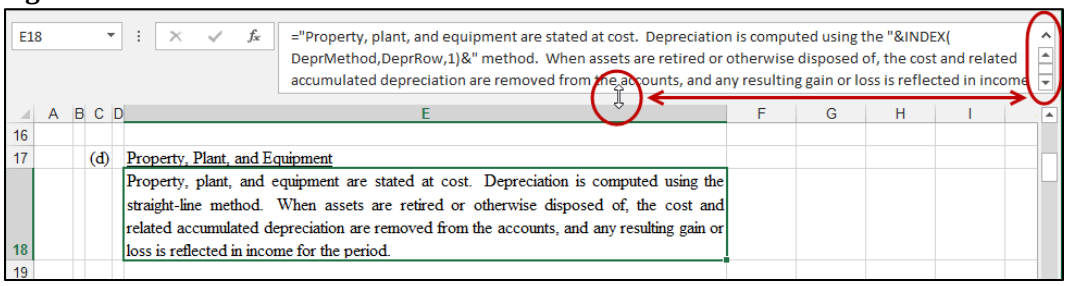


Figure 4 - Expanding the Formula Bar or Scroll to See More Lines

If the formula or text exceeds three lines, use the scroll buttons to scroll through the formula or to drag the bottom edge of the formula bar to expand its size. To split a formula into separate lines for easy reading or troubleshooting, position the cursor in the formula and press **ALT + Enter** as shown in **Figure 5**.

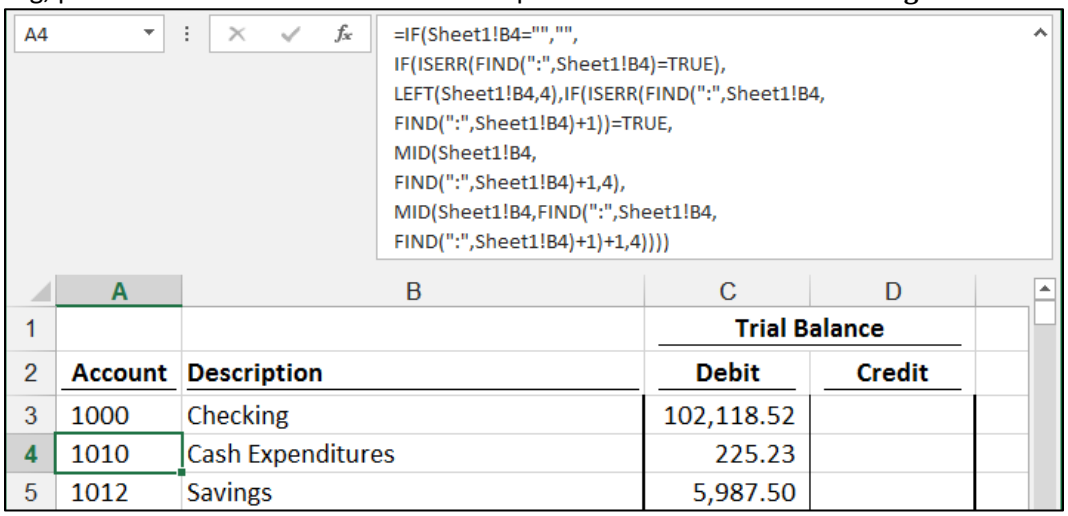


Figure 5 - Using ALT + Enter to Split a Formula

The complex formula shown in **Figure 5** parses an exported Trial Balance from QuickBooks into multiple columns. If the formula was run together in the Formula Bar, it would be much more difficult to understand but using **ALT + Enter** to split the formula makes reading and understanding it easier.

The Formula Tab

The **Formula** tab of the Ribbon, shown in **Figure 6**, contains four standard groups – **Function Library**, **Defined Names**, **Formula Auditing**, and **Calculation**. The Function Library contains organized lists of all functions available in Excel. The Defined Names group contains the Name Manager and other tools for defining and using defined names.

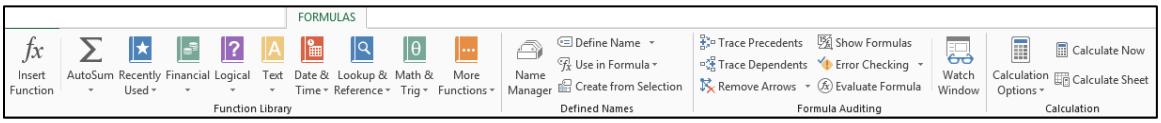


Figure 6 - The Formula Tab in Excel

The Formula Auditing group contains tools for auditing formulas or watching results, and the Calculation group allows you to calculate a workbook or to adjust calculation settings. The Solutions group appears only when specific add-ins are enabled.

Accruals and Deferrals with Date Arithmetic

In Excel, a date is just another number formatted to look like a date. To be precise, a date is a serial number that represents the number of days since January 1, 1900, with 1 representing January 1, 1900, and 2 representing January 2, 1900, etc. Similarly, 41,274 represents December 31, 2012. Excel's method of treating dates as numbers makes it easy to deal with dates in formulas. For example, to compute the number of days between two

dates, simply subtract the earlier date from the later date. Accountants and other financial professionals can take advantage of this feature in computing items such as accrued or deferred revenues and expenses. **Figure 7** displays a worksheet that uses date arithmetic to compute accrued interest on notes.

	A	B	C	D	E
1	**ABC Holding Company**				
2	**Accrued Interest on Notes**				
3	**For the Date Ended:**				
4	**January 25, 2016**				
5					
6	**Days in Year:**	**365**			
7	**Date Issued**	**Due Date**	**Amount**	**Interest Rate**	**Accrued Interest**
8	6/19/2012	6/19/2016	$ 360,000.00	4.20%	$ 54,473.42
9	7/21/2012	7/21/2016	210,000.00	3.90%	28,788.41
10	8/15/2012	8/15/2016	150,000.00	4.30%	22,230.41
11	10/1/2012	10/1/2016	450,000.00	4.40%	65,692.60
12	10/15/2012	10/15/2016	850,000.00	4.60%	128,226.58
13	10/21/2012	10/21/2016	320,000.00	4.80%	50,119.89
14			$ 2,340,000.00		$ 349,531.32

Figure 7 - Using Date Arithmetic to Compute Accrued Interest

Parsing and Combining Labels with String Arithmetic

Excel's text functions can be used to combine or parse labels. For example, simple formulas that use the ampersand (**&**) as the operator can add labels to other labels or to numbers. This can be useful to accountants in building integrated workbooks that separate assumptions from analyses and reports. **Figure 8** shows how text functions and string arithmetic can be used to tie report headings to common assumptions. As the assumptions are changed, the report headings are changed. In this case, the reporting period and reporting date update by simply changing the values in the assumptions section. The **TEXT** function is used to embed the report date in the report heading. The TEXT function converts values (numbers) into strings (labels) and allows you to stipulate the format in which the value is displayed.

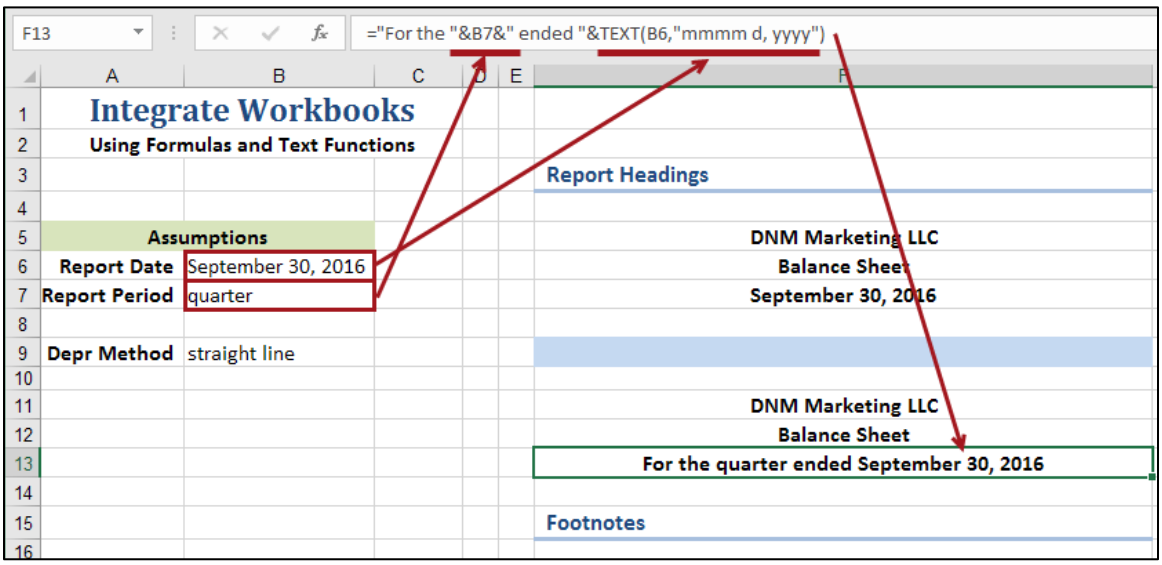


Figure 8 - Using String Arithmetic to Update Report Headings Easily

The example in **Figure 9** shows how string arithmetic can be used to tie footnotes to the financial reports or analysis to which they apply. In this case, the **DOLLAR** and TEXT functions are used to convert values in a report to a string that is embedded within the report's surrounding text.

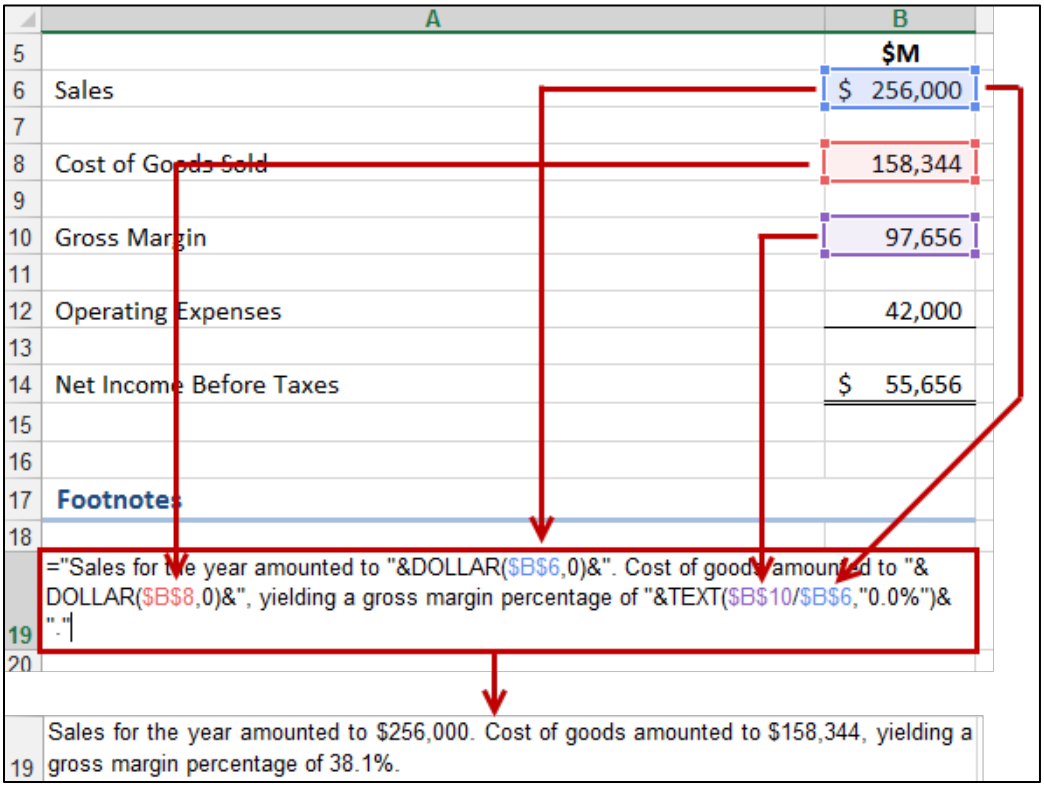


Figure 1 - Using the Dollar and Text Functions to Create Footnotes Linked to Financial Data

Just as functions can be used to combine labels and numbers, functions can also be used to parse, or divide, text into its various components. The example formulas displayed in **Figure 10** parse the combined account number and account name of a QuickBooks trial balance into two columns, one for the account number and the other for the account name. The formula that parses the account number uses the **LEFT** function, while the one that parses the account name uses the **RIGHT** and **LEN** functions together.

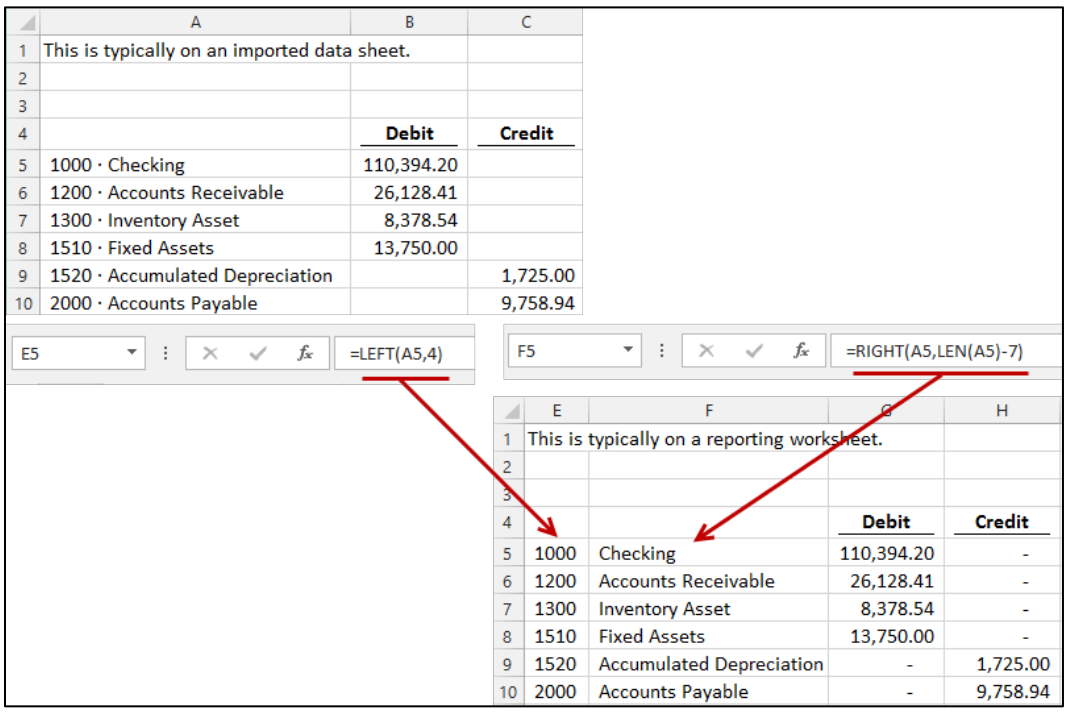


Figure 10 - Parsing Account Numbers and Titles Using String Functions

Other more advanced text functions can be used to parse data when the text components have less common characteristics than the data in the trial balance example above. For example, a common problem that accountants face is grouped or combined data fields – data that is a combination of multiple data elements in a single database field. Many casual users of Excel will routinely combine data elements, such as city, state, and zip code, in a single column, which may hinder the effective use of the data. Excel's text functions can parse the combined data elements into individual columns or extract a particular element for use. **Figure 11** displays formulas using functions LEFT, RIGHT, **MID**, **FIND**, and LEN to parse contact details into individual columns for city, state, and zip code.

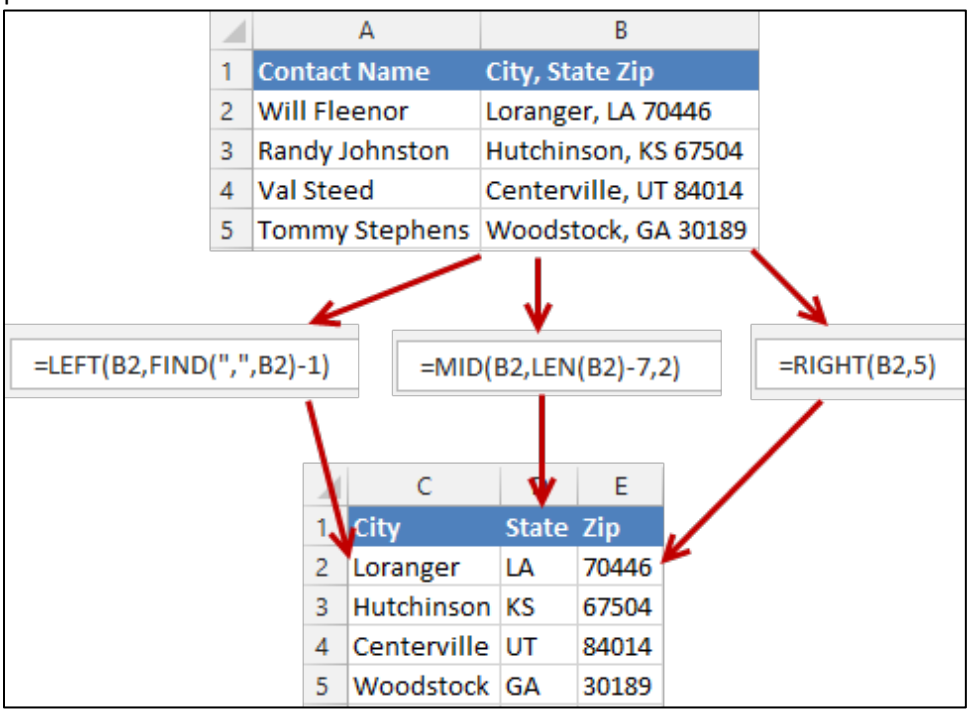


Figure 11 - Parsing Contact Details

Parsing Data with Text to Columns

Parsing text with Excel functions requires knowledge and skill to produce satisfactory results. The **Text to Columns** command automates the process of converting text that is a combination of several data elements into individual columns of data. Select the data to parse and click **Text to Columns** on the **Data** tab of the Ribbon as shown in **Figure 12** to begin the process.

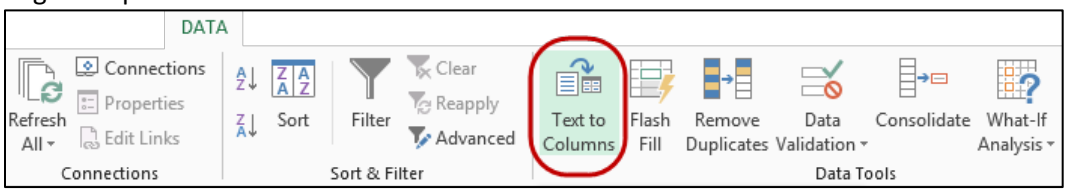


Figure 12 - Text to Columns Available on the Data Tab of the Ribbon

In this case, the process will require two steps – the first to parse the city from the state and zip code, and the second to parse the state from the zip code. The first step uses the **comma** as the parsing delimiter; the second uses the **space** as the delimiter. In parsing the state from the zip code, make sure to select **Text** as the **column data format** for the zip code as shown in **Figure 12**. Otherwise, Excel will remove leading zeros from any zip codes that begin with "0," such as those often found in the northeastern part of the United States.

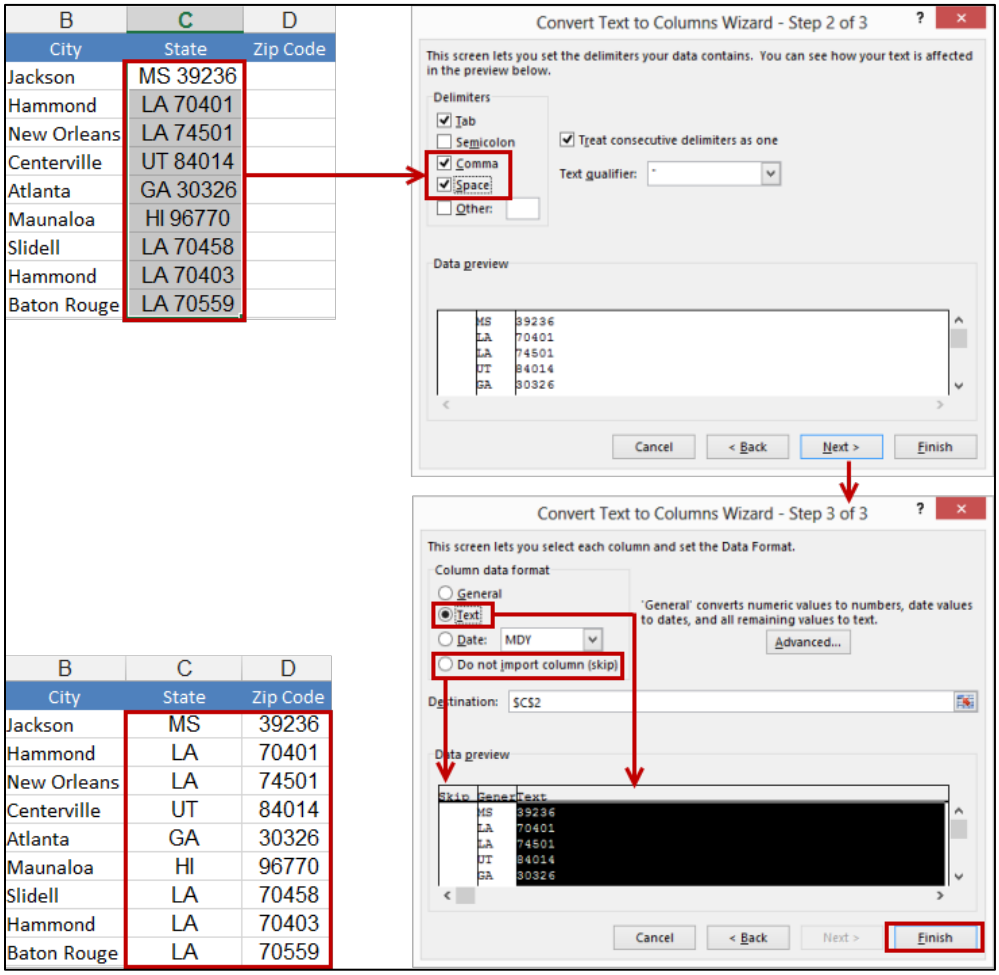


Figure 12 - Parsing State and Zip Code Using Text to Columns

Using Defined Names

Defined names, referred to as "named ranges" in Lotus 1-2-3, can be used in place of cell references when building formulas. For most Excel users, defined names are more easily remembered than corresponding cell references, constants, or formulas when building worksheet models.

Defined names can be created and used in almost any manner imaginable. They can represent:

- Numeric constants, such as a tax rates;
- Text constants, such as a company name;
- Absolute, relative, or mixed cell or range references;
- Three-dimensional cell or range references; or
- Formulas.

Defined names can be created in numerous ways in Excel. The easiest method is to highlight the cell or range of cells to be named and then to type the desired name in the **Range Box** on the **Formula Bar** just below the Ribbon as shown in **Figure 13**.

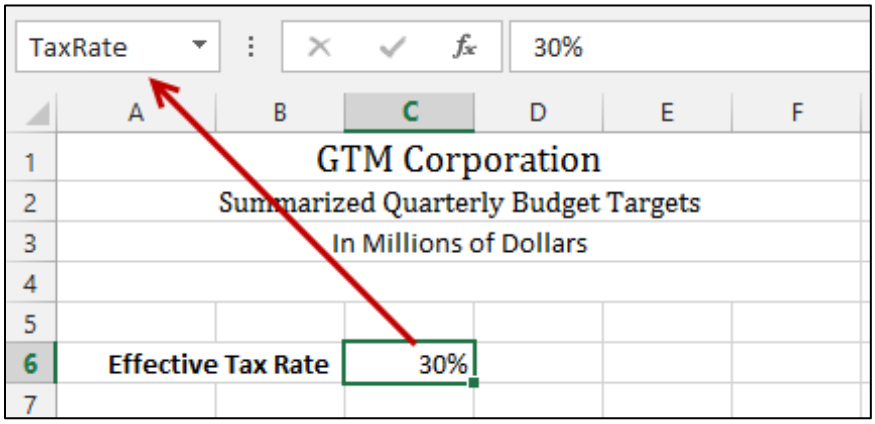


Figure 13 - Using the Name Box to Create a Defined Name

> Excel has naming rules for defined names. A partial list of these rules appears below.
>
> - Names must begin with a letter, backslash, or underscore.
> - No spaces are allowed in a defined name.
> - Names that resemble cell references cannot be used.
> - Single letters cannot be used for names except for R, r, C, c.
> - Names can contain up to 255 characters and are *not* case-sensitive.

Another way to create a defined name is to highlight the cell or range of cells to be named and then to click **Defined Name** on the **Formulas** tab of the Ribbon to open the **New Name** dialog box shown in **Figure 14**. Type in the desired **Name**, select the **Scope**, enter a **Comment**, and modify the **Refers to** information as required. Click **OK** to create the defined name.

Note that the reference in the **Refers to** box normally begins with an equal sign (=), but that is *not* a requirement. If you type an entry that does not begin with an equal sign, it is treated as a *text constant*. For example, you could create "ORDC" as a defined name to represent your company name, Opihi River Development Corporation. To use the name anywhere in the workbook, you could just type in "=ORDC".

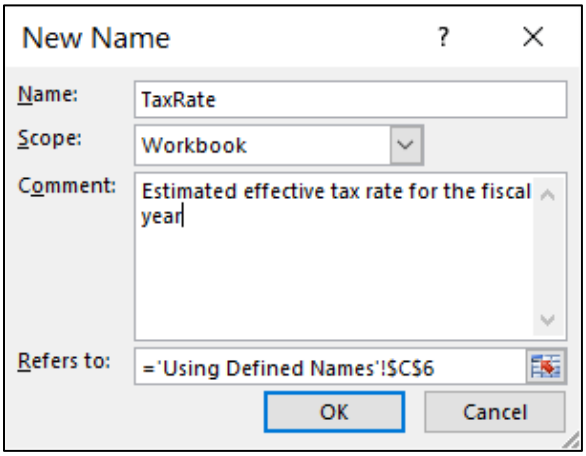


Figure 14 - Using the New Name Dialog to Create a Defined Name

Defined names can also be created from adjacent labels. Just highlight the range, including the labels, and from the **Formulas** tab of the Ribbon, select **Create from Selection** as shown in **Figure 16**. In the **Create Names from Selection** dialog box, check the boxes that correspond with the position of the labels in the selected range and click **OK** to create the names automatically. Any spaces or special characters in the labels will be replaced with underscores in the defined names. Defined names created from labels that begin with a number such as **1Q2012**are preceded with an underscore.

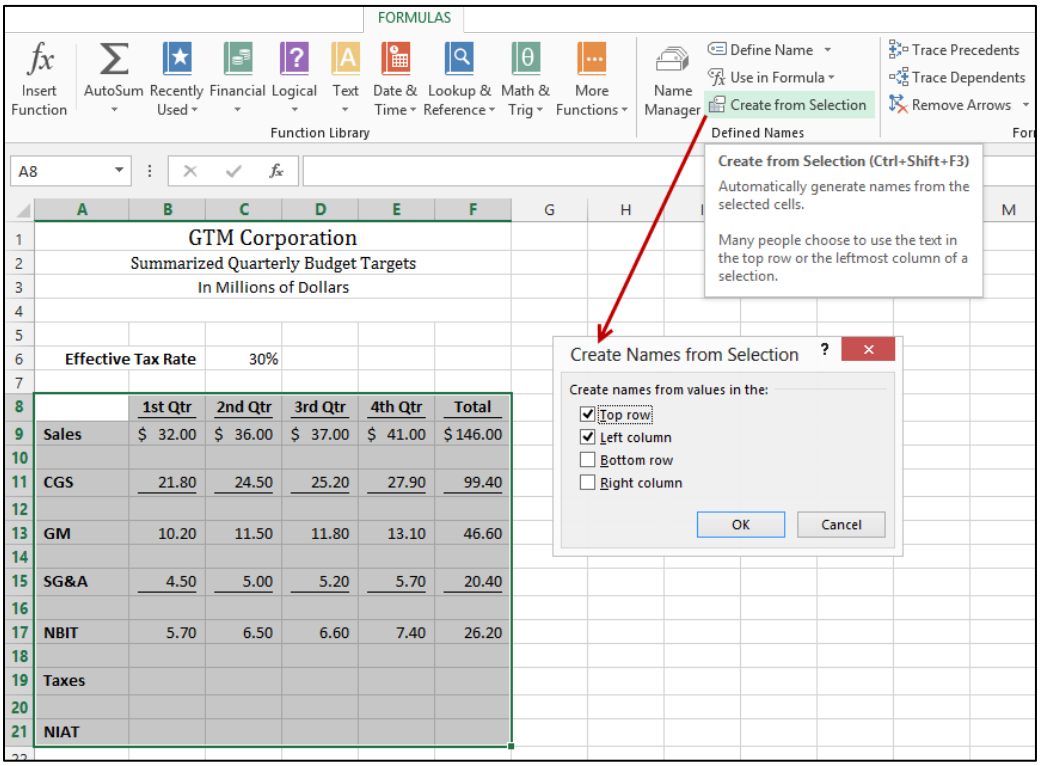


Figure 2 - Creating Defined Names from Adjacent Labels

> With the dramatic increase in the size of the worksheet grid beginning in Excel 2007, names such as **TAX2005** that were formerly acceptable are no longer allowed because the name is now identical to a valid cell reference. These names will continue to function as defined if the workbook is opened in *compatibility mode*. If a workbook containing such names is converted to an Excel 2007 or newer workbook, Excel changes the names so that they are preceded with an underscore, such as **_TAX2005**.

Using Names in Formulas

One of the great advantages of using defined names is that they create *self-documenting* formulas. In other words, the formula explains the calculation by the names used in it. For example, =**Sales-CGS** explains more clearly the calculation of Gross Margin than =**B4-B6**. There are at least three ways to use defined names in formulas.

1. Enter the defined names from the keyboard while creating formulas.

2. Enter the first few letters from the keyboard while creating formulas and then use Formula AutoComplete to select the name.

3. Click **Use in Formula** on the **Formulas** tab of the Ribbon to select a defined name while building a formula. Note that **Use in Formula** only lists the defined names relevant to the active worksheet.

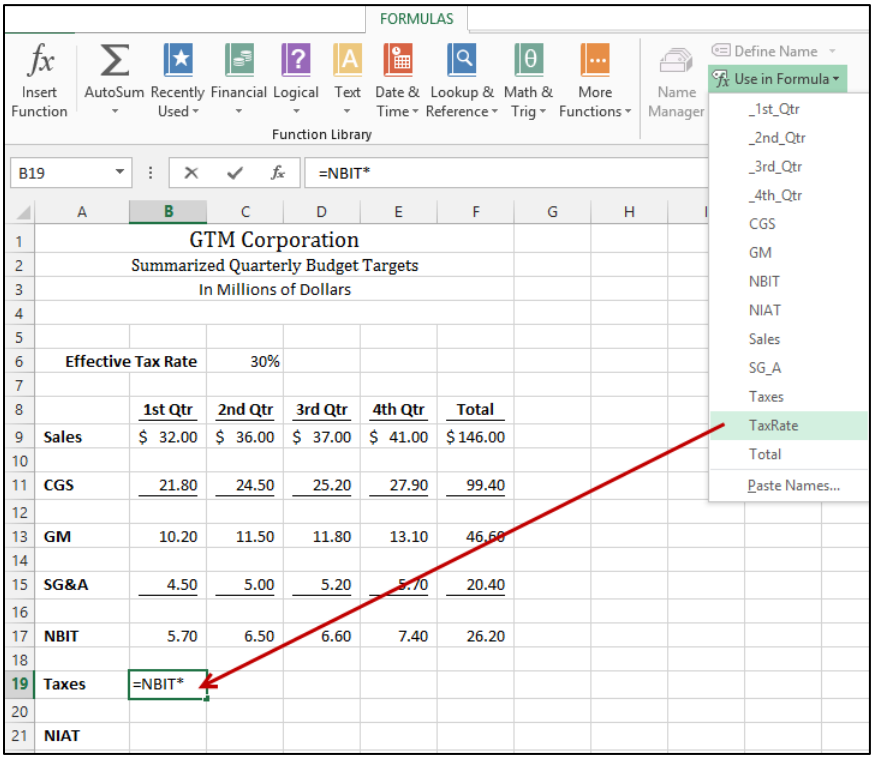


Figure 3 - Inserting Defined Names with Use in Formula

In the example worksheet, we need to create formulas to compute **Taxes** and **NIAT**. Position the cursor in cell B19 and then complete one of the following procedures to produce the first formula.

1. Type in **=NBIT*TaxRate** and press **Enter**.

2. Alternatively, begin the formula by typing **=NB** and then double-click on the defined name **NBIT** in the AutoComplete list. Type in ***** followed by **Tax**. Double-click on the defined name **TaxRate** and press **Enter**.

3. As another option, begin the formula by typing =. From the **Formulas** tab of the Ribbon, click **Use in Formula** and then click on **NBIT**. Type ***** and then click on **Use in Formula**, this time selecting **TaxRate** as shown in **Figure 101**. Press **Enter**.

4. Create a similar formula in cell B21 for NIAT. Copy the formulas across to complete the worksheet.

If you subsequently define names for ranges already used in formulas, you can replace the cell references with their defined names using a single command. From the **Formulas** tab of the Ribbon, click on the drop-down arrow

to the right of **Define Name** and select **Apply Names**. Select the names to be applied and click **OK**. The command can be applied to a specified range or to the entire workbook. *CAUTION!* If a defined name is deleted, formulas referencing the deleted name will return the **#NAME** error message.

Conditional and Boolean Calculations

The **IF** function is used to make calculations based on whether specified conditions are met. The two latest versions of Excel support up to sixty-four (64) nested IFs in each formula. The syntax of the IF function follows.

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

The value returned by the function depends on the results of the logical test. In the example in **Figure 17**, a formula based on the IF function is used to calculate a budget variance percentage, but only if the budget amount is not equal to zero, thereby avoiding the **#DIV/0** error. If the budget amount is equal to zero, the formula returns a null string, which displays a blank cell. To return a null string, use two quotation marks entered side-by-side as shown in the sample formula.

```
=IF($C7<>0,($D7-$C7)/$C7,"")
```

Figure 17 - Formula to Return Null String If Budget Equals Zero

The Boolean functions – **AND**, **OR**, and **NOT** – can be used in combination with other functions to produce compound conditional formulas. For example, to create a formula that requires multiple conditions to be met in order to return a specified result, use AND in conjunction with the IF function as shown in **Figure 103**.

Larry's Landscaping Services
Actual to Budget Performance
Baton Rouge Location
For the Month Ended December 31, 2016

		Budget	Actual	% Variance	$ Variance	'AND'	'OR'	'NOT'
6230	Auto Fuel	$ 21,800.00	$ 19,696.58	-9.65%	$ (2,103.42)	TRUE	TRUE	TRUE
6240	Auto Maintenance	21,800.00	19,696.58	-9.65%	(2,103.42)	TRUE	TRUE	TRUE
6400	Bank Service Charges	100.00	89.00	-11.00%	(11.00)		TRUE	TRUE
6550	Payroll Expenses	340,000.00	344,865.36	1.43%	4,865.36		TRUE	
6600	Delivery Fee	1,500.00	3,292.95	119.53%	1,792.95	TRUE	TRUE	
6900	Insurance	32,500.00	39,457.85	21.41%	6,957.85	TRUE	TRUE	
7100	Equipment rental	7,300.00	7,373.28	1.00%	73.28			
7200	Miscellaneous	8,000.00	17,227.74	115.35%	9,227.74	TRUE	TRUE	
7300	Office Supplies	18,000.00	17,227.74	-4.29%	(772.26)			TRUE
7500	Rent	77,000.00	77,373.28	0.48%	373.28			
7553	Equipment Repairs	10,500.00	9,556.66	-8.98%	(943.34)		TRUE	TRUE
7700	Tools and Misc. Equipment	3,500.00	3,708.05	5.94%	208.05		TRUE	
7751	Gas and Electric	8,300.00	9,256.00	11.52%	956.00		TRUE	
7753	Telephone	3,000.00	2,941.22	-1.96%	(58.78)			TRUE
7752	Water	1,865.00	1,686.38	-9.58%	(178.62)		TRUE	TRUE
9000	Interest Expense	6,000.00	7,433.98	23.90%	1,433.98	TRUE	TRUE	
	Total Operating Expenses	$ 561,165.00	$ 580,882.65		$ 19,717.65			

```
=IF(AND(ABS($E7)>0.05,ABS($F7)>1000),"TRUE","")
=IF(OR(ABS($E7)>0.05,ABS($F7)>1000),"TRUE","")
=IF(NOT($F7>0),"TRUE","")
```

Figure 4 - Using Boolean Functions to Build Compound Conditional Formulas

Conditional IF Functions

Excel includes built-in functions to do conditional sums, counts, and averages – **SUMIF**, **SUMIFS**, **COUNTIF**, **COUNTIFS**, **AVERAGEIF**, and **AVERAGEIFS**. The singular versions of these functions (SUMIF, COUNTIF, and AVERAGEIF) only support a single conditional test and are similar in application, operation, and syntax. The plural versions of these functions (SUMIFS, COUNTIFS, and AVERAGEIFS) do conditional sums, counts, and averages with up to 127 conditions. Their syntax is slightly different. For example, compare the syntax of SUMIF with SUMIFS.

SUMIF(range, criteria, sum range)

SUMIFS(sum range, criteria range1, criteria1, criteria range2,criteria2…)

Note that the *sum range* is the last criterion in the SUMIF function, but the first criterion in the SUMIFS function. SUMIFS can support criterion references for up to 127 criteria. Similar to SUMIF, criteria can be entered as numbers, cell references, or text surrounded by quotation marks and can contain wildcard characters. However, criteria ranges *must* be the same shape and size as the sum range.

GTM Manufacturing Company compiles a list of weekly sales by region and product. The CFO needs to produce a report each month that sums the sales by product, by region, and by product within region. In the past, she has sorted and summed the data multiple times in order to create and paste the desired totals to her report sheet. In Excel, she can create the report with three simple formulas – one SUMIF to total sales by region, one SUMIF to total sales by product, and one SUMIFS to sum sales by product within region. The SUMIF formulas will use *absolute* addresses to define the sum and criteria ranges so that the formula can be copied down without error. The SUMIFS formula will use *absolute* addresses to define the sum and criteria ranges, but it will use *mixed* addresses for the criteria so that the formula can be copied down *and* across. **Figure 19** displays the worksheet used to produce the required totals.

	A	B	C	D	E	F	G	H	I	J	K
1	GTM Manufacturing Inc										
2	Sales Data in Thousands										
3	For the period 07/01 through 07/28										
4											
5											
6	**Date**	**Region**	**Product Line**	**Amount**							
7	7/7/2016	Northeast	Lotions	623.12		**Region**					
8	7/7/2016	Northeast	Cremes	486.03		Northeast	=SUMIF($B$7:$B$42,F8,$D$7:$D$42)				
9	7/7/2016	Northeast	Masks	218.09		South					
10	7/7/2016	South	Lotions	654.28		Midwest					
11	7/7/2016	South	Cremes	510.34							
12	7/7/2016	South	Masks	229.00							
13	7/7/2016	Midwest	Lotions	672.97		**Product Line**					
14	7/7/2016	Midwest	Cremes	524.92		Lotions	=SUMIF($C$7:$C$42,F14,$D$7:$D$42)				
15	7/7/2016	Midwest	Masks	235.54		Cremes					
16	7/14/2016	Northeast	Lotions	647.00		Masks					
17	7/14/2016	Northeast	Cremes	504.66							
18	7/14/2016	Northeast	Masks	226.45							
19	7/14/2016	South	Lotions	679.35		**Product Line Within Region**					
20	7/14/2016	South	Cremes	529.89			Lotions	Cremes	Masks		
21	7/14/2016	South	Masks	237.77		Northeast	=SUMIFS($D$7:$D$42,$B$7:$B$42,$F21,$C$7:$C$42,G$20)				
22	7/14/2016	Midwest	Lotions	698.76		South					
23	7/14/2016	Midwest	Cremes	545.03		Midwest					
24	7/14/2016	Midwest	Masks	244.57							

Figure 5 - Using SUMIF and SUMIFS to Produce a Sales Report

The formula to produce sales totals by product within region is shown in **Figure 20**. Note the use of mixed addresses in the formula for the criteria. In this formula, the criterion for ***Region*** is $F21, which is absolute as to column and relative as to row. As the formula is copied across columns, the formula will always look to column F for the ***Region*** criterion relative to the row in which the formula resides. The criterion for ***Product Line*** is G$20, which is relative as to column and absolute as to row. As the formula is copied down across rows, the formula will always look to row 20 for the ***Product Line*** criterion relative to the column in which the formula resides. Stipulating *mixed* addresses for the criteria is the magic that allows the formula to be copied down and across.

```
=SUMIFS($D$7:$D$42,$B$7:$B$42,$F21,$C$7:$C$42,G$20)
```

Figure 20 - Using Mixed Addresses in Formulas

Summarizing Data with SUBTOTAL

Many accounting workbooks have multiple levels of subtotals that are ultimately summarized into a grand total. The **SUBTOTAL** function makes this process easier and less prone to error, especially when using big datasets with many subtotals. Nested subtotals (subtotals within subtotals) are ignored to avoid double counting. Grand totals are always calculated on the detail data, not on any included intermediate (nested) subtotals. The SUBTOTAL function can summarize *columns* of data using any of eleven methods identified in **Table 1**.

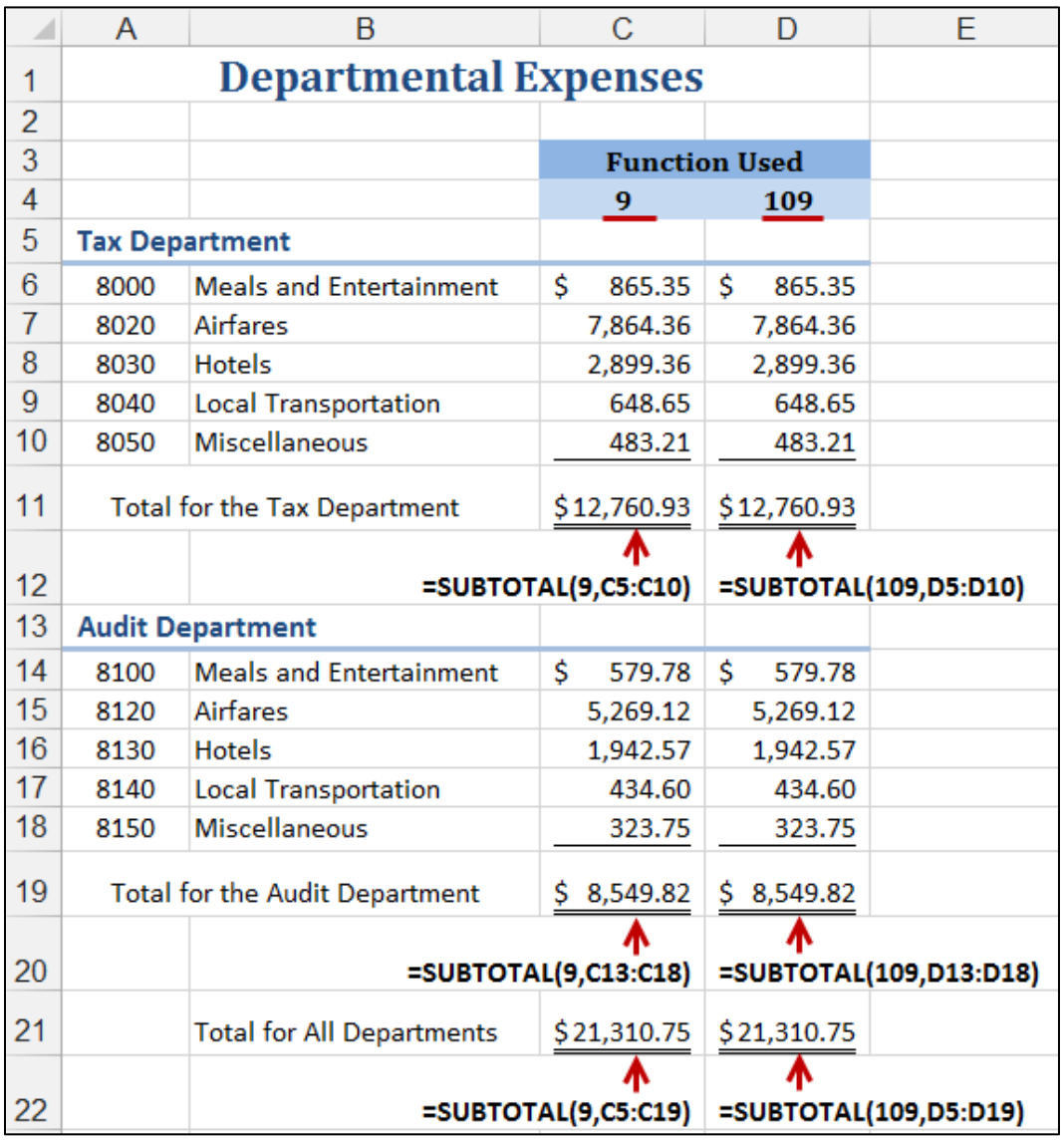
	Function Number	
Function	Includes Hidden Values	Ignores Hidden Values
AVERAGE	1	101
COUNT	2	102
COUNTA	3	103
MAX	4	104
MIN	5	105
PRODUCT	6	106
STDEV	7	107
STDEVP	8	108
SUM	9	109
VAR	10	110
VARP	11	111

Table 1 - SUBTOTAL Functions in Excel

Note that there are two columns of functions in the table. The first indicates that values on hidden rows in the subtotal are included in the calculation. The second column ignores values on hidden rows.

Data Filters alter the rules of the SUBTOTAL function. Summarizations calculated on filtered data always exclude values on hidden rows regardless of the function used. In other words, functions 1 and 101 do not include values on hidden rows in the calculation of subtotals. Furthermore, operation of the **AutoSum** button accessible on the **Home** tab of the Ribbon is altered when a *filter is active* so that SUBTOTAL functions are entered rather than simple summarization functions, such as SUM, AVERAGE, or COUNT, etc.

The example in **Figure 106** uses the SUBTOTAL function to produce subtotals on rows 11 and 19 and a grand total on row 21. The formulas in the left column of amounts uses function **9** in the SUBTOTAL function. These formulas include hidden values in the subtotals. The formulas in the right column of amounts use function **109** in the SUBTOTAL function. These formulas do not include hidden values in the subtotals. If function 109 is used, it is a simple task to create a report that excludes miscellaneous expenses. Just hide the two miscellaneous expense rows, and the calculations of the subtotals and grand total will immediately reflect their absence.



	A	B	C	D	E
1	**Departmental Expenses**				
2					
3			**Function Used**		
4			**9**	**109**	
5	**Tax Department**				
6	8000	Meals and Entertainment	$ 865.35	$ 865.35	
7	8020	Airfares	7,864.36	7,864.36	
8	8030	Hotels	2,899.36	2,899.36	
9	8040	Local Transportation	648.65	648.65	
10	8050	Miscellaneous	483.21	483.21	
11	Total for the Tax Department		$12,760.93	$12,760.93	
12			**=SUBTOTAL(9,C5:C10)**	**=SUBTOTAL(109,D5:D10)**	
13	**Audit Department**				
14	8100	Meals and Entertainment	$ 579.78	$ 579.78	
15	8120	Airfares	5,269.12	5,269.12	
16	8130	Hotels	1,942.57	1,942.57	
17	8140	Local Transportation	434.60	434.60	
18	8150	Miscellaneous	323.75	323.75	
19	Total for the Audit Department		$ 8,549.82	$ 8,549.82	
20			**=SUBTOTAL(9,C13:C18)**	**=SUBTOTAL(109,D13:D18)**	
21		Total for All Departments	$21,310.75	$21,310.75	
22			**=SUBTOTAL(9,C5:C19)**	**=SUBTOTAL(109,D5:D19)**	

Figure 21 - Using SUBTOTAL to Automate the Totaling Process