

Using Formulas in Excel

An Introductory Guide

Table of Contents

Formulas in a Nutshell	3
<i>Introduction to Formulas.....</i>	<i>3</i>
<i>Using Functions</i>	<i>6</i>
Referencing Cells, Sheets and Workbooks in a Formula	12
<i>Referencing an Entire Column or Row.....</i>	<i>12</i>
<i>Relative, Absolute and Mixed References</i>	<i>12</i>
<i>Referencing other Sheets and Workbooks</i>	<i>15</i>
<i>Using Named Ranges</i>	<i>15</i>
<i>Formatting a Range as a Table</i>	<i>20</i>
Solving Formula Errors.....	24
<i>Formula Errors Explained</i>	<i>24</i>
<i>Common Formula Problems</i>	<i>26</i>
<i>Preventative Measures.....</i>	<i>29</i>
<i>Troubleshooting Formulas.....</i>	<i>35</i>
Dynamic Array Formulas in Excel	41
<i>Writing a Dynamic Array Formula.....</i>	<i>41</i>
<i>Spill Range and Spill Error.....</i>	<i>42</i>
<i>How to Reference the Spill Range</i>	<i>44</i>
<i>Dynamic Array Formulas and Table Data.....</i>	<i>45</i>
<i>Dynamic Arrays Cannot Be Used in Tables</i>	<i>46</i>
<i>New Dynamic Array Functions</i>	<i>47</i>
<i>Dynamic Array Formulas with Other Excel features.....</i>	<i>48</i>
Contact Details	52

Formulas in a Nutshell

In this chapter, we introduce the basic rules of creating formulas and using functions in Excel. The information in this chapter will prepare you for everything else that is to come in the rest of the book.

This chapter is intended for those with little or no experience with formulas in Excel, however this information will benefit most Excel users of any level and serve as a good reference point throughout the book.

This chapter covers;

- An introduction to writing formulas.
- The different symbols (operators) used in formulas.
- How to determine the order of calculation.
- An introduction to using functions.

Introduction to Formulas

Getting Started with Formulas

A formula is a calculation in Excel. In its simplest form, a formula can be used to add or multiply values like you would on a calculator. A more advanced formula may include the use of different functions, and link data between different worksheets.

A formula always begins with the equal sign (=). When this sign is entered Excel immediately assumes a calculation (or formula) is coming.

In addition to the equal sign, a formula could also include some or all of the following.

- **Operators:** A formula can consist of different mathematical and logical operators such as + (addition), * (multiplication) or > (greater than). See Table 1: List of formula operators.
- **Cell references:** A formula is certain to need to refer to a cell or range of cells on a worksheet. There are various techniques to referencing worksheet ranges and these are all covered later in this chapter.
- **Parentheses:** These are used to control the order of calculation. This order is more commonly known as the rule of BODMAS, where the parentheses are referred to as brackets.
- **Values and text strings:** Different values and/or text strings can be used in a formula. When entering text into a formula it is entered within double quotes like "Yellow".
- **Functions:** Functions can be included in a formula. A function is a predefined formula to achieve a specific result such as to sum a range of values, or to calculate an average.

Table 1: List of formula operators

Operator	Definition
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Percent
&	Text concatenation
^	Exponentiation
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

Entering a Formula

To enter a formula, simply click in the cell where you want the result to be displayed. Type the equal sign followed by the different formula elements required. Then press Enter when you have finished typing the formula.

When you reference cells in a formula, you can do this by typing the reference, or by selecting the cells with the mouse. Excel will change the colour of the cells in use to make it easier to track your formula progress.

Here are some formula examples;

Adding the values from cells A2 and B2.

```
=A2+B2
```

Multiplying the values from cells A2 and B2.

```
=A2*B2
```

Dividing the value in A5 by 2 (a fixed value)

```
=A5/2
```

This formula includes the SUM function which has been used to total the values in range A5:A10. This total is then subtracted from the value in A3.

```
=A3-SUM(A5:A10)
```

Calculating 10% of the value in A2.

```
=A2*10%
```

Editing a Formula

Once a formula has been entered, only the result will be visible in the cell. If you click on the cell to select it, the formula will be displayed in the Formula Bar.

To edit the formula you can make the required changes in the Formula Bar. Or to display the formula on the worksheet either double click on the cell, or press F2.

The Order of Calculation

When entering a formula you need to be aware of the order of calculation. And more importantly how you can control it by using parentheses (or brackets).

Take the example below for instance.

```
=A2+B2*C2
```

This formula has been written with the intention of performing A2+B2 first, and then multiplying the total by C2. However Excel will calculate B2*C2 first and then add A2 to the total.

This order of calculation is commonly known as the rule of BODMAS. This acronym reminds us of the order of the mathematical operators.

Brackets

Order

Division

Multiplication

Addition

Subtraction

Put simply, it means that Excel will always divide and multiply before it adds and subtracts. You can control the order of operation though by using brackets.

So the previous example can be corrected by adding brackets like the formula below.

```
=(A2+B2)*C2
```

In more complex formulas you may have nested parentheses like the one below.

```
= ( (A2/2) + (B2/2) + (C2/5) ) *D10
```

In this example the deepest nested parentheses are evaluated and then the formula works its way out. This means that the three division calculations are performed, and then the adding of the three values and then the multiply by the value in D10.

Fortunately unless these kind of formulas are required from you, they are not commonly found on most spreadsheets.

Most formulas make use of predefined functions, where although the concept of the order of calculation still applies, most of the hard work is built into the function simplifying the process.

Calculating Percentages

A common requirement of writing formulas in Excel is to calculate percentages. Examples include to calculate customer discounts, VAT, inflation and salesperson commission.

As listed in Table 1: List of formula operators, the % sign can be used when writing formulas. This means that typing 10% in a formula is the equivalent of typing 0.1.

Here are some examples of formulas that perform common percentage calculations in Excel.

Finding 15% of the value in cell A2.

```
=A2*15%
```

Subtracting 15% from the value in cell A2.

```
=A2*85%
```

Adding 15% to the value in cell A2.

```
=A2*115%
```

Calculating the percentage for the value in cell B2 of the value in cell A2 (The cell containing the formula must be formatted as a percentage).

```
=B2/A2
```

The percentage increase, or decrease, of the value in cell B2 to the one in cell A2. Notice the use of parenthesis to calculate the subtraction part of the formula first. The cell containing this formula must also be formatted as a percentage.

```
=(B2-A2)/A2
```

Using Functions

A function is a predefined formula. Excel contains more than 450 functions. These range from basic mathematical calculations to more complex financial and engineering calculations.

When writing formulas, these built-in functions are your new best friend. They make it easier to create powerful and meaningful formulas that would otherwise have seemed impossible.

The good news is that all of the functions in Excel follow the same structure. So once you understand the different elements of a function, learning them becomes much easier.

The Anatomy (Syntax) of a Function

Every function in Excel is written using the functions name, and then followed by a set of parentheses like in the example below.

```
=NOW ( )
```

If you do not include the parentheses with the function name when entering it, you may encounter the #NAME? error. More on formula error values later in this chapter.

The parentheses are used to store the functions arguments. Put simply, its arguments are the information that the function needs to do its job.

A function may require no arguments, a specific number of arguments or have a mixture of mandatory and optional arguments.

The previously mentioned NOW function is an example of a function that requires no arguments. The example below shows the NETWORKDAYS function. These functions will be covered fully later in the book. For now we want to concentrate on its syntax.

```
=NETWORKDAYS(start_date, end_date, [holidays])
```

This function has 3 arguments. You can think of these arguments as question for you, so that the function can do what you need it to. Each argument is separated using a comma.

The NETWORKDAYS function calculates the number of working days between two dates. It is probably no surprise then that it asks you for the start date, end date, and if you want to include any holidays (any dates to be exempt from the calculation).

Because the holidays argument is displayed in square brackets ([]), this identifies it as being an optional argument.

Throughout the course of this book, whenever a function is mentioned, its syntax will be explained.

Entering a Function

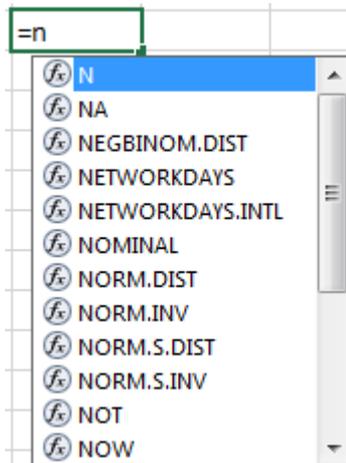
There are 3 different ways of entering a function into a formula.

Typing the Function

The most common approach is just to type the function into the formula using its name (Figure 0-1). This is the quickest way if it is a function you are familiar with.

As you type, a list appears to help. You can continue typing, or double click on the function name in the list.

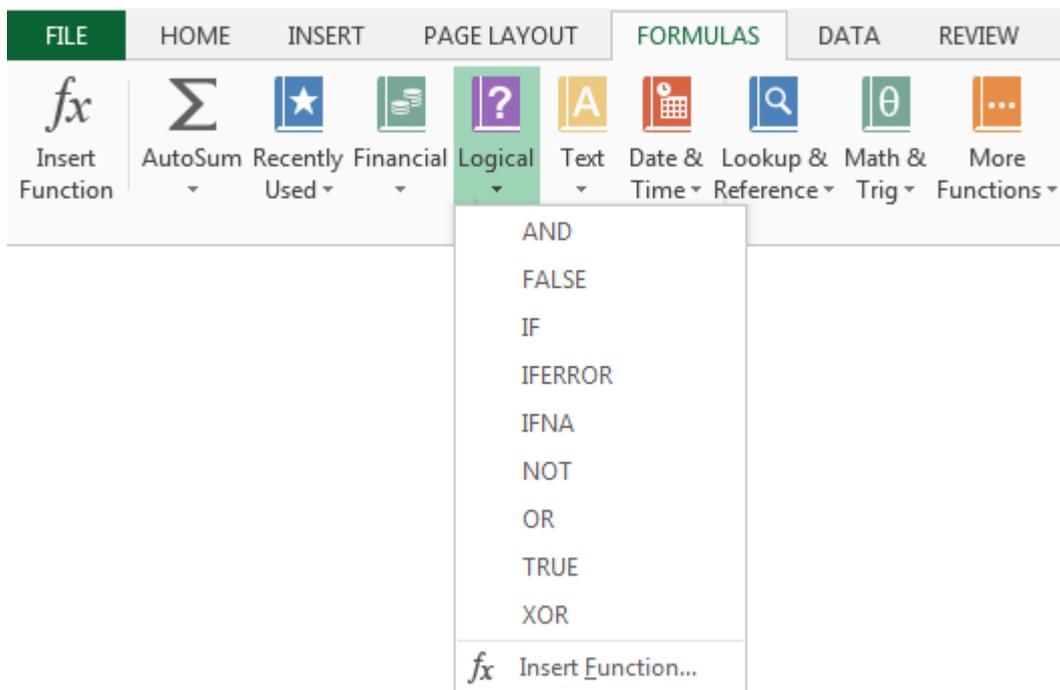
Figure 0-1



Using the Categories on the Formulas Tab

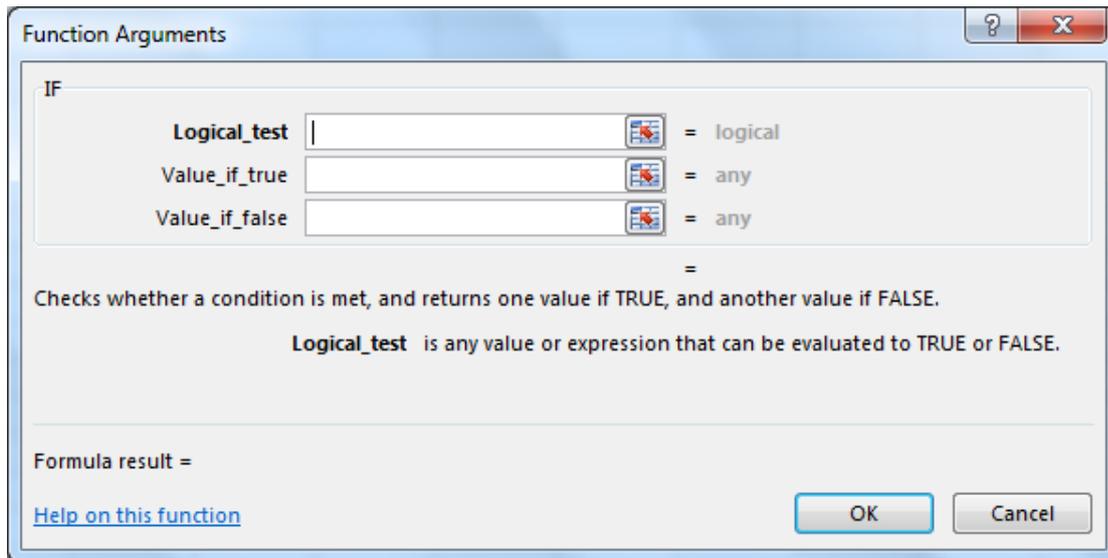
Another way of entering a function is to use the categories provided on the **Formulas** tab. Figure 0-2 shows the Logical category being explored.

Figure 0-2



When you select a function from a category list, you are taken to the function arguments dialog box (Figure 0-3).

Figure 0-3



This approach is slower than typing the function into the formula, but it provides greater assistance.

Figure 0-3 shows the *logical_test* argument in bold indicating that it is a mandatory argument. The others are optional.

Required data types of the answer are also displayed to the right of each argument box, with an explanation provided below.

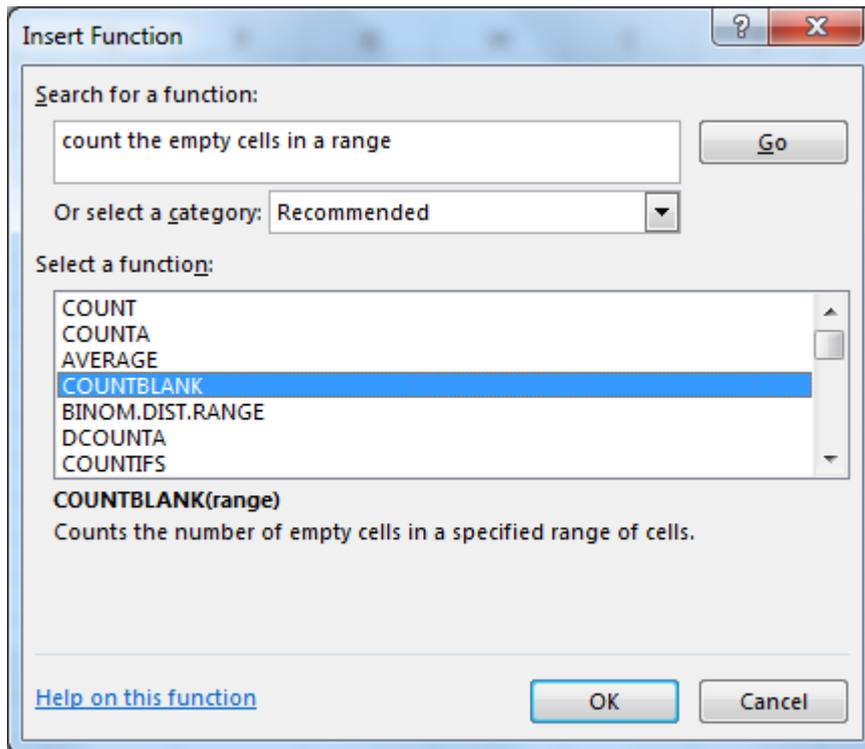
If more help is still required, a **Help on this function** link can be found at the bottom. This link will take you to the Microsoft Help page on that function explaining it in greater detail and with examples.

Using the Insert Function Dialog

The final approach to entering a function is to use the Insert Function dialog. You can open this dialog by clicking the **Insert Function** button on the **Formulas** tab, or by clicking the **fx** icon to the left of the Formula Bar.

If you are not sure of the function you require, you can enter some keywords in the **Search for a function** box and click **Go** (Figure 0-4). A list of functions meeting your criteria is returned. As you select a function, its description is displayed at the bottom of the dialog.

Figure 0-4



Once you have found the function you need you can click **Ok**. This will take you to the Function Arguments dialog explained previously.

You can also open the Insert Function dialog when editing a formula to view the Function Arguments dialog and to get some help.

The SUM Function

The SUM function is the most commonly used function of all. Its purpose is to sum a range of values.

Its syntax is;

```
=SUM(number1, [number2], ...)
```

number1: The range of cells containing the values you want to sum.

The SUM function below adds all the values in range B2:B20.

```
=SUM(B2:B20)
```

Much faster and more efficient than using a formula for B2+B3+B4+B5 and so on. As you can see, when entering a range of cells the colon (:) is used to specify every cell from B2 to B20.

This shorthand notation means that even when summing 20000 cells the formula would just look like below.

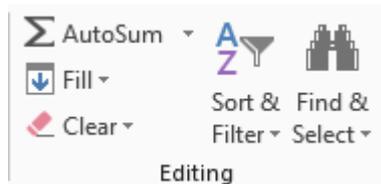
```
=SUM(B2:B20001)
```

The SUM function can accept multiple cell ranges. The SUM function below is being used to sum 3 different cell ranges. Each range is provided as a different argument and therefore separated by a comma.

```
=SUM(B2:B20, D2:D20, F2:F20)
```

You can enter this function by using any of the techniques mentioned previously, or you can click the **AutoSum** button (Figure 0-5) on the **Home** tab.

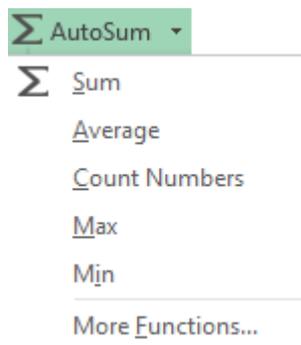
Figure 0-5



Other Aggregate Functions

In addition to the SUM function, there are 4 more functions that can be accessed by clicking the arrow next to the **AutoSum** button on the **Home** tab (Figure 0-6).

Figure 0-6



The good news is that they also have the same syntax as the SUM function, so if you are confident with SUM, you just learnt 4 more functions.

The AVERAGE function below calculates the mean average of cells A2:A20.

```
=AVERAGE(A2:A20)
```

The COUNT function counts how many values (numbers) occur in range A2:A20.

```
=COUNT(A2:A20)
```

The MAX function returns the largest value in cells A2:A20.

```
=MAX(A2:A20)
```

The MIN function returns the smallest value in cells A2:A20.

```
=MIN(A2:A20)
```

As well as being spreadsheet functions, these five functions are also built into many of Excel's reporting tools including PivotTables and Subtotals.

Referencing Cells, Sheets and Workbooks in a Formula

When writing formulas in Excel you will need to reference values that exist in cells on the same worksheet, a different worksheet, or even a different workbook entirely.

Excel provides a few different techniques to do this. This chapter will look into some of these different techniques and why you would use them.

This chapter covers;

- Relative, absolute and mixed references.
- Referencing a cell on a different sheet or workbook.
- Using named ranges to simplify cell references in formulas.
- Formatting a range as a table for dynamic and structured references.

Referencing an Entire Column or Row

All the formulas mentioned so far in this book have used a reference to a specific range of cells e.g. A2:A10. However, you may find it useful to refer to an entire column or row.

For example the following formula sums all of the values in column D.

```
=SUM(D:D)
```

The advantage to referring to an entire column is that your formula will continue to work if the range being summed changes. So if more values are added to the list, or even values removed, the formula continues to return the correct result.

Referring to an entire column does not impact the speed that Excel calculates. Excel keeps a note of the last used cell in a column, and therefore will not calculate every cell in the entire column.

Be careful to ensure that you do not write the formula inside the same column that you are referencing, as this will cause a circular reference (more on this later).

Relative, Absolute and Mixed References

There are four different ways of referencing a cell when using the cell address in a formula. These are known as relative, absolute and the mixed references (row absolute and column absolute).

The different types of cell reference can be identified by the use of the dollar sign (\$). This dollar sign is used to lock the column and/or row of a reference so that it does not move if the formula is copied.

The dollar signs can be typed into the formula, or the **F4** key can be used to convert a cell reference when writing a formula.

Relative References

A relative cell reference does not contain any dollar signs. This means that if the formula is moved or copied, the cells being referred to are moved in relation to the cell containing the formula.

Take the following formula that has been entered into cell B7 in Figure 0-1.

```
=SUM(B3:B5)
```

Because relative references have been used, if the formula is copied into cells C7:E7, the cells being referred to will adjust in relation to the movement of the cell containing the formula.

So the formula in cell C7 would become =SUM(C3:C5). The formula was copied to the right, so all the cells being referenced also move to the right.

Figure 0-1

	A	B	C	D	E
1					
2		January	February	March	April
3	Games	£ 2,400.00	£ 1,425.00	£ 2,400.00	£ 3,100.00
4	Movies	£ 1,280.00	£ 2,000.00	£ 5,100.00	£ 2,200.00
5	Music	£ 3,000.00	£ 1,600.00	£ 2,400.00	£ 2,400.00
6					
7	Total	£ 6,680.00			

Absolute References

An absolute cell reference contains a dollar sign before the column and row in the address. This type of reference is fully locked, and will not change if the formula is copied in either direction.

The formula below has been entered into cell D3 in Figure 0-2.

```
=C3*$G$2
```

The reference to cell C3 is relative, whilst the reference to cell G2 has been made absolute.

This has been done to ensure that when the formula is copied down to the other cells of the table, the reference to C3 changes to cell C4, C5 and so on, and the reference to cell G2 does not change.

Figure 0-2

	A	B	C	D	E	F	G
1							
2	Item No	Item	Sales	VAT		VAT	20%
3	B24	Coffee Table	£1,024.00	£204.80			
4	A23	Wardrobe	£745.00				
5	C11	Garden Tables	£2,300.00				
6	C12	Garden Chairs	£2,670.00				
7	A21	Bed	£600.00				

Mixed References

There are two types of mixed reference; the row absolute and the column absolute. These are partially absolute references and the dollar sign is used to lock the required column or row.

Row absolute: Example B\$1. The column will adjust when a formula is copied, but the row will not.

Column absolute: Example \$A2. The row will adjust when a formula is copied, but the column will not.

The formula below has been entered into cell B2 in Figure 0-3. This formula was then copied across and down to complete the times table.

= \$A2 * B\$1

Figure 0-3

	A	B	C	D	E	F	G	H	I	J
1		2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Referencing other Sheets and Workbooks

The cells and ranges that you reference in a formula will commonly be on other worksheets, or even a different workbook.

You can select these cells in the same manner that you do with cells on the same sheet. Nothing more is required from you and Excel will write the reference.

Understanding how to write these references can be helpful though, especially if you work with other peoples spreadsheets often, and need to understand their formulas.

When referring to a cell on a different worksheet, you simply enter the sheet name followed by an exclamation mark (!) to the left of the cell reference.

The example below references cell B2 on a sheet called *Oxford*.

```
=Oxford!B2
```

If the sheet name contains spaces, it must be enclosed in single quotation marks (').

```
= 'Oxford Sales' !B2
```

If the cell you are referencing is on a different workbook, the workbook name must be entered inside square brackets to the left of the sheet name.

```
= [Monthly Expenses.xlsx]Oxford Sales' !B2
```

If the other workbook is open, only the workbook name is shown. If the other workbook is closed, the file path is also included in the reference.

```
= 'H:\Stores\2015\[Monthly Expenses.xlsx]Oxford Sales' !B2
```

You must be careful when working with workbooks that link to other workbooks. Users renaming files, or changing directories can create unexpected and unwanted problems. Unless you feel it absolutely necessary you are better working within one workbook.

Using Named Ranges

It is possible to assign a name to a cell or range of cells in a workbook. This practice can simplify the writing and understanding of formulas.

Compare for example even a simple formula like the one below.

```
=B22-B5
```

And then the equivalent formula when the cells have been named with something meaningful.

```
=Income-Expenses
```

This formula is more descriptive both for yourself, but also for others who may use this workbook.

This chapter will serve as only an introduction to the use of named ranges and the role they may play in your formulas. They do however have much more potential than what we cover here.

The key advantages to using named ranges are;

- They enable you to use descriptive names within formulas making references easier to identify and formulas easier to understand.
- A named range is usually unique to a workbook and can therefore be referenced from any sheet. This simplifies referencing cells on different worksheets.
- Named ranges can be absolute so the use of dollar signs is not necessary in the formula.
- Formulas can be faster to write when using these meaningful names.
- Formula problems can be easier to resolve when named ranges are used. A tool called the Name Manager will list the value in every cell in one place.

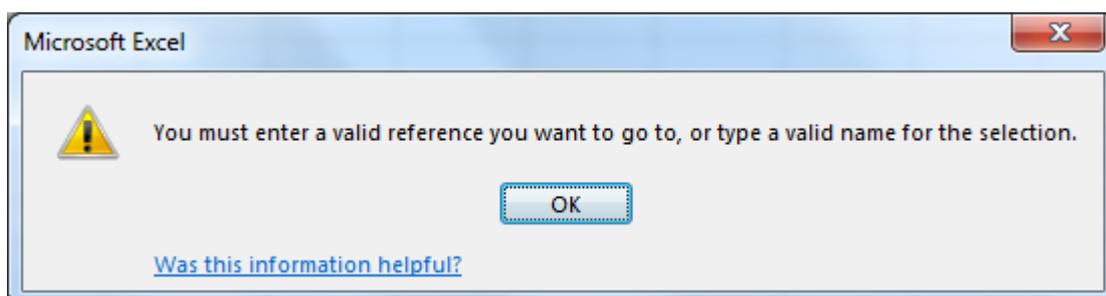
Creating Named Ranges

There are a few different ways to create a named range in Excel. We will cover two ways of doing this shortly, but first there are some rules to naming ranges you should be aware of.

- **The name cannot contain any spaces.** You may want to use an underscore character to separate the words such as *March_Sales*, or join them together as one word using a capital letter to identify the change in word such as *MarchSales* (this is a programming technique called *Camel Casing*).
- **Names can only start with a letter or an underscore.** Numbers are allowed in named ranges, but not as the first character.
- **Symbols such as %, \$, @, & and ! are not allowed in a name.** Most symbols are not allowed in the name of a range. In fact I believe the only ones that are allowed are the underscore (_), question mark (?), period (.) and backslash (\).
- **You cannot use the address of a cell as a name.** This one may seem obvious but is very easily done by mistake. For example, the following innocent looking named ranges would not be allowed as they are also the address of a cell *CAR2*, *TAX2015*, *VAT15*, *BAG100*.

If you violate any of these rules when naming a range, the error in Figure 0-4 is displayed.

Figure 0-4

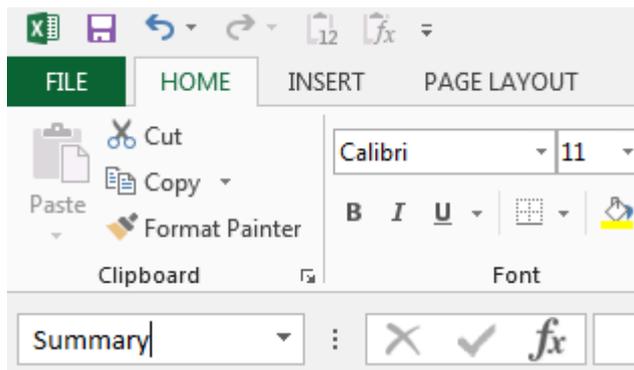


Create a Name Using the Name Box

This is the quickest way to create a named range, but it is limited and cumbersome when creating names for more advanced uses.

1. Select the range of cells you want to name (this can be a single cell, range or even an entire column).
2. Click in the Name Box and type the name you want to use (Figure 0-5).
3. Press **Enter** when finished to confirm your entry

Figure 0-5



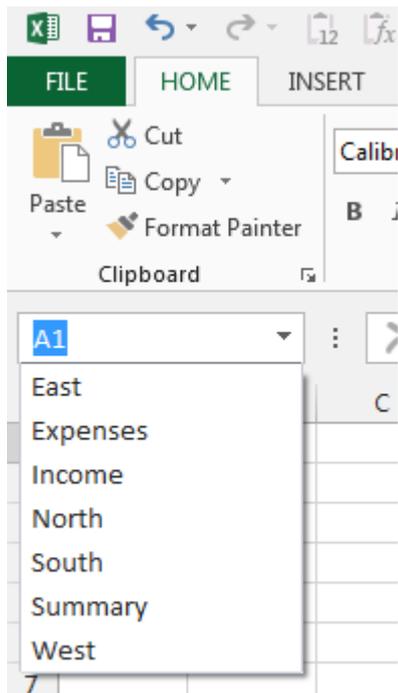
You must press Enter when you have finished typing the name in the Name Box. Simply clicking on the worksheet will not create the name.

When creating a named range using the steps prescribed above, this name will be unique across all sheets of the workbook. This name will also be an absolute reference.

This is normally what users want from their named ranges, but it is important to know so that you can edit the name if necessary to meet your requirements.

The Name Box can also be used to quickly select a named range. Click the arrow on the end of the Name Box and choose the name from the list (Figure 0-6).

Figure 0-6

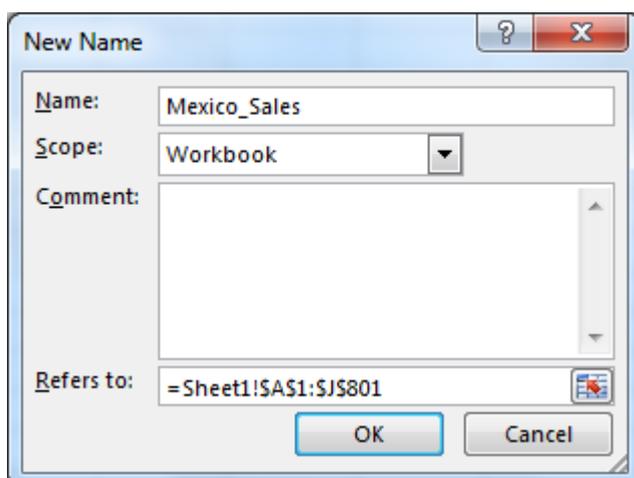


Using the Define Name Button

Using the Define Name button takes a little more effort, but it gives you greater control over the settings for your named ranges.

1. Select the range of cells you want to name.
2. Click the **Formulas** tab and then the **Define Name** button.
3. Complete the dialog box (see Figure 0-7) and click **Ok**.

Figure 0-7



Name: Enter a name for the range.

Scope: By default, named ranges are at workbook level. This means that a name is unique across the whole workbook and can therefore be referenced from any sheet. This is typical of what users want from their named ranges.

This can be changed to create a name at worksheet level, meaning the name is local to that specific sheet. By doing this the same name can also be created on the other sheets of the workbook. This can be a really useful technique.

Comment: Enter a description for the named range.

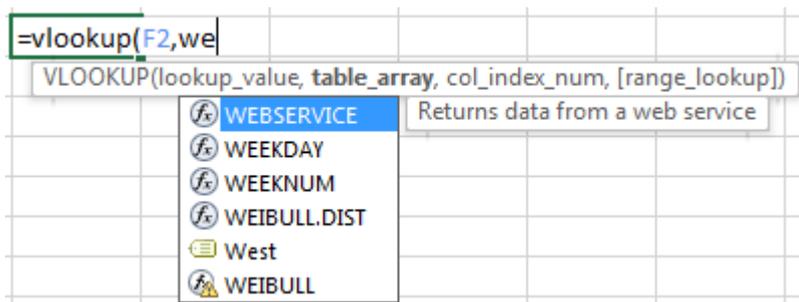
Refers to: This field will contain the address of the selected range. You can use this field to edit the range, remove the dollar sign for a relative reference, or even enter a formula.

Using Named Ranges in a Formula

Once named ranges have been set up, they can be used within your formulas. To use a named range, simply type its name when entering or editing a formula.

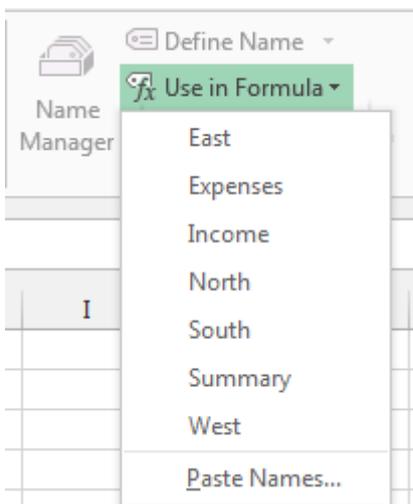
Figure 0-8 shows the Formula AutoComplete list to assist you in referencing your defined names and functions. Notice the *West* named range in the list with the name tag icon to the left of its name.

Figure 0-8



If you cannot remember the name that has been used for a range, you can click the **Use in Formula** button on the **Formulas** tab to select the name from a list.

Figure 0-9

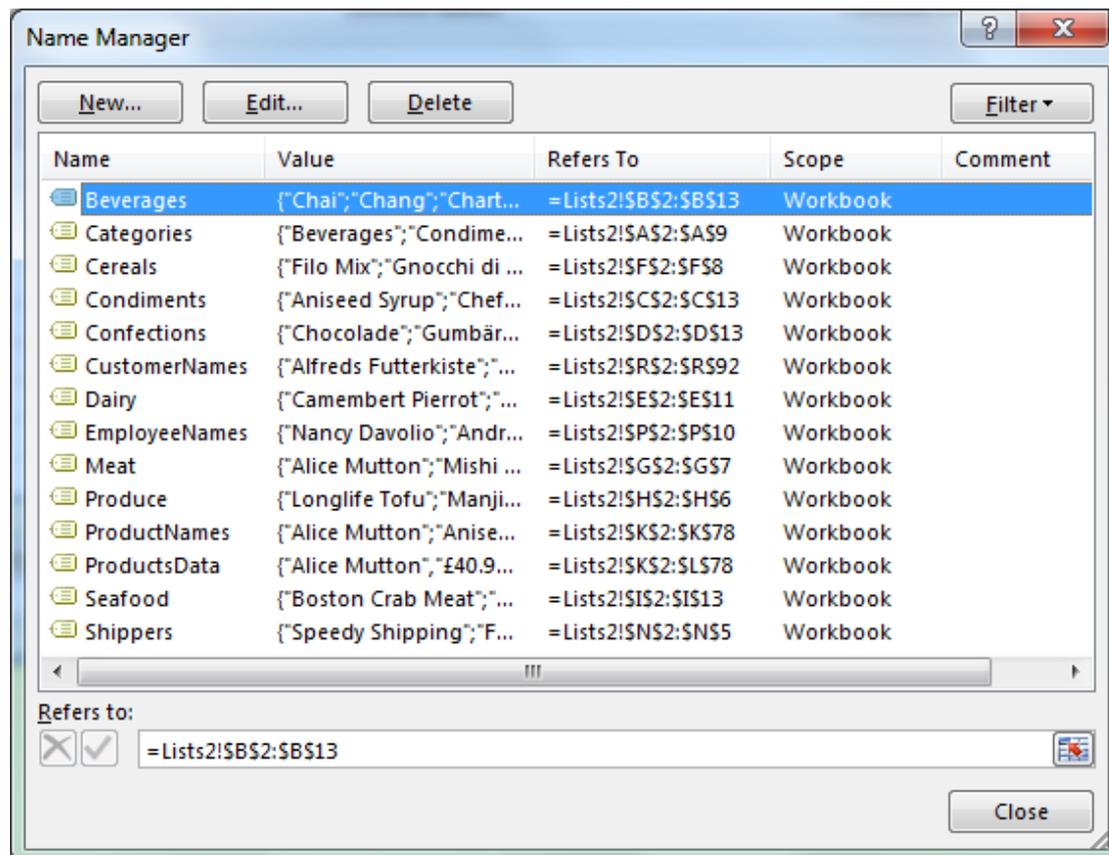


Viewing, Editing and Deleting Named Ranges

Once they have been setup, any viewing or modifying of named ranges can be done from the Name Manager. Click the **Name Manager** button on the **Formulas** tab to open it.

The Name Manager (see Figure 0-10) will display all the names that can be found in the workbook. It lists their current value, where they can be found and their scope.

Figure 0-10



The **Edit** and **Delete** buttons can be used to edit or delete existing names.

You cannot edit or delete a named range from the Name Box mentioned earlier. This can only be used to create names and to navigate to them.

Formatting a Range as a Table

Excel provides a 'Format as Table' feature so that you can convert a range of cells into a table. This can make some common Excel tasks easier, including formulas.

Not every range of cells will work as a table. Ranges of cells that you are using for a list such as a list of sales, employees, inventory, or expenses, are perfect for this feature. Figure 0-11 shows such a list. This list is probably large and has a single header row along the top making it perfect for a table.

Functions such as VLOOKUP, AGGREGATE, SUMIFS and features such as a PivotTable work well with tables.

Really though they are just another alternative for referencing cells. They have their benefits, but they also have their negatives, and you need to use what is right for you and your team.

Figure 0-11

	A	B	C	D	E	F
1	ProductID	ProductName	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder
2	17	Alice Mutton	20 - 1 kg tins	£ 39.00	0	0
3	3	Aniseed Syrup	12 - 550 ml bottles	£ 10.00	13	70
4	40	Boston Crab Meat	24 - 4 oz tins	£ 18.40	123	0
5	60	Camembert Pierrot	15 - 300 g rounds	£ 34.00	19	0
6	18	Carnarvon Tigers	16 kg pkg.	£ 62.50	42	0
7	1	Chai	10 boxes x 20 bags	£ 18.00	39	0
8	2	Chang	24 - 12 oz bottles	£ 19.00	17	40
9	39	Chartreuse verte	750 cc per bottle	£ 18.00	69	0
10	4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	£ 22.00	53	0
11	5	Chef Anton's Gumbo Mix	36 boxes	£ 21.35	0	0
12	48	Chocolade	10 pkgs.	£ 12.75	15	70
13	38	Côte de Blaye	12 - 75 cl bottles	£ 263.50	17	0
14	58	Escargots de Bourgogne	24 pieces	£ 13.25	62	0

Benefits of Using Tables

Let's have a look at some of the benefits of formatting a range as a table.

- You do not need Freeze Panes to keep the header row visible when scrolling down the worksheet. The header row disappears, but the headers and filter replace the column letters at the top of the spreadsheet (Figure 0-12).
- A table is automatically assigned a name. The first table being *Table1*. This name can be changed and then easily referred to just like named ranges mentioned previously.
- You can quickly and easily control the formatting of the entire table using styles.
- A table automatically expands when new data is added at the bottom, or to its right. Anything using the table such as charts, PivotTables, named ranges and formulas are then automatically resized.
- If necessary, a table can easily be resized by clicking and dragging the bottom right corner of the table.
- From Excel 2013, Slicers (a powerful filter tool) can be used on tables.

Figure 0-12

	ProductID	ProductName	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	G
19	56	Gnocchi di nonna Alice	24 - 250 g pkgs.	£ 38.00	21	10	
20	31	Gorgonzola Telino	12 - 100 g pkgs	£ 12.50	0	70	
21	6	Grandma's Boysenberry Spread	12 - 8 oz jars	£ 25.00	120	0	
22	37	Gravad lax	12 - 500 g pkgs.	£ 26.00	11	50	
23	24	Guaraná Fantástica	12 - 355 ml cans	£ 4.50	20	0	
24	69	Gudbrandsdalsost	10 kg pkg.	£ 36.00	26	0	
25	44	Gula Malacca	20 - 2 kg bags	£ 19.45	27	0	
26	26	Gumbär Gummibärchen	100 - 250 g bags	£ 31.23	15	0	
27	22	Gustaf's Knäckebröd	24 - 500 g pkgs.	£ 21.00	104	0	
28	10	Ikura	12 - 200 ml jars	£ 31.00	31	0	
29	36	Inlagd Sill	24 - 250 g jars	£ 19.00	112	0	
30	43	Ipoh Coffee	16 - 500 g tins	£ 46.00	17	10	
31	41	Jack's New England Clam Chowde	12 - 12 oz cans	£ 9.65	85	0	

These benefits are great and there are a lot of tools built-in to tables. But that is a conversation for another time.

We want to focus on how formatting a range as a table will impact our formulas. And that means the key benefit as far as this book is concerned is the introduction of structured references.

What are Structured References

Structured references provide a meaningful and durable way of referring to data that is stored in a table.

Take the table in Figure 0-12 for example. When used as a range of cells, the following formula is used to calculate the total price of stock.

```
=D2 * E2
```

But with structured references it would look like.

```
=[@UnitPrice] * [@UnitsInStock]
```

Another great thing with structured references is that they do not change the way they look (like cell references) when the table is added to. For example, the formula below would look the same regardless of the size of the table.

```
=SUM(Inventory[UnitsInStock])
```

Creating a Table

So, before you can go using structured references, you will need to format the range of cells as a table.

1. Select the range of cells you want to format.
2. Click the **Home** tab and then the **Format as Table** button.
3. A list of formatting styles are shown for you to choose from. Select the one you want.

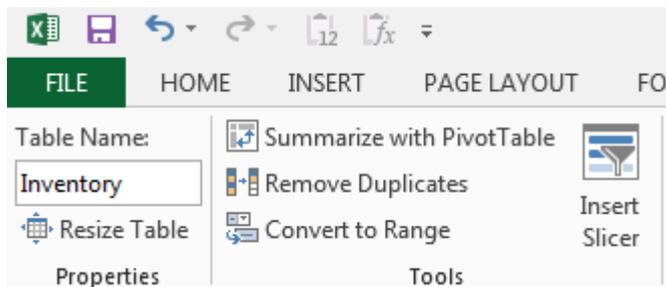
It is possible to create your own table style. This may be of interest if you have to conform to a company standard, or you just want more scope for creativity.

Naming a Table

Naming a table will make it easier to reference from within a formula. The table already has a name, but a name like *Table1* or *Table2* is probably not much use to you.

1. Click in the table and then on the **Design** tab on the Ribbon.
2. Click in the **Table Name** box and enter the name you want to use (see Figure 0-13).

Figure 0-13



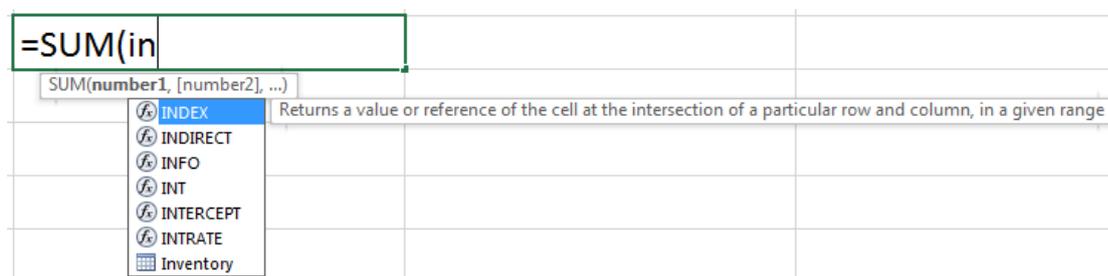
Referencing Table Cells in a Formula

When writing a formula, referencing cells of a table is similar to how you reference named ranges. This approach is in addition to selecting the cells by clicking on them which is still possible.

Let's look at the example of using the SUM function on the *UnitsInStock* column mentioned earlier.

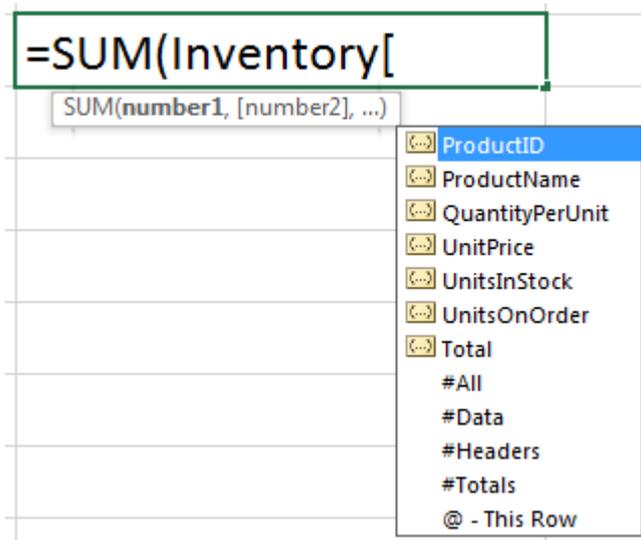
If you are on a different worksheet when typing the formula, begin by entering the name of the table (see Figure 0-14). This will appear in the Formula AutoComplete list like functions and named ranges do. Notice the table icon to the left of the name.

Figure 0-14



Once the table is selected, type an opening square bracket ([). This indicates that you want to reference a field (column) within the table. Excel responds by displaying a list of all the fields for you to choose from (see Figure 0-15).

Figure 0-15



After a field is selected, type the closing square bracket (]) and continue with the formula as necessary.

Formatting a range as a table can simplify your formulas especially when you need to reference across different worksheets, and need to look at whole columns of the table. They provide meaningful names that make formulas easier to read and digest also.

Structured references are not used in any of the formulas throughout this book as I wanted a consistent way of referencing cells in the formula examples. However, feel free to test the examples out with tables to get a better feel of how they might work for you.

Solving Formula Errors

When writing formulas in Excel, you will encounter formula errors. These errors can occur for many reasons. This chapter explores some of these reasons, common types of formula error and techniques for solving them.

This chapter covers;

- An explanation for the different types of formula error.
- Common mistakes that cause these errors and how they could be prevented.
- Excel's formula auditing tools to assist you in solving formula problems.

Formula Errors Explained

Not all problems with formulas return an error message. The formula may be syntactically correct, but returns the wrong answer. This could be because you are summing the wrong range of cells, or you have used incorrect logic with your IF function.

However you have probably seen some formula error messages before. The purpose of these errors is to get your attention, and also explain the root of the problem.

Let's have a look at these error messages and what they mean.

#VALUE!

The formula uses data of the incorrect data type. For example, a formula is adding values but one of the values is stored as text.

When faced with a #VALUE! Error you should check the arguments of the function. You may have provided information using the wrong data type.

You may also see this error if you forget to press Ctrl + Shift + Enter when using an array formula.

#N/A

The formula references a value that is not available to it yet.

This error is commonly associated with lookup functions such as VLOOKUP and MATCH. In this scenario users commonly want to hide the error.

The formula below shows the IFNA function (new to Excel 2013) being used to make the cell appear blank, instead of displaying the #N/A error.

```
=IFNA(VLOOKUP($H$4,$A$2:$F$78,4,FALSE),"")
```

#NAME?

The formula contains text that it does not recognise.

Commonly displayed because a function or a range name has been misspelt. Also if a text entry is not enclosed in quotation marks.

#DIV/0?

The formula is trying to divide by 0 or a blank cell.

This error is typical when the formula has been entered in a table but the values the formula uses are not available yet. In this example the cells would be blank and cause a #DIV/0? Error.

This formula does not need to be calculated and that is why the error has occurred. The error can be hidden using the IFERROR function like in the example below.

```
=IFERROR(B2/C2,"")
```

#REF!

The formula uses an invalid cell reference.

This error will typically appear if you delete a row, column or sheet that is being used by the formula.

For example, the #REF! error is displayed for the following formula if the *Boston* sheet is deleted.

```
=Boston!D5
```

#NULL!

An intersection of two cell ranges has been specified but do not intersect.

The intersection operator in formulas is a space. Therefore this error is commonly seen when a space has been accidentally entered into a formula.

In the example below, an operator has been accidentally omitted and a space entered. This will result in a #NULL! Error.

```
=B2*C2 D2
```

#NUM!

The function contains an invalid argument.

This could be caused because the function is passed a non-numeric argument when a numeric argument was expected, or because the value is too large or too small.

Common Formula Problems

This list describes some common formula problems which may be causing your formula to produce an error value.

The Value is Too Big for the Column

If the value returned by a formula is too large for its column then the cell will be filled with hash symbols (###). To correct this you need to increase the width of the column.

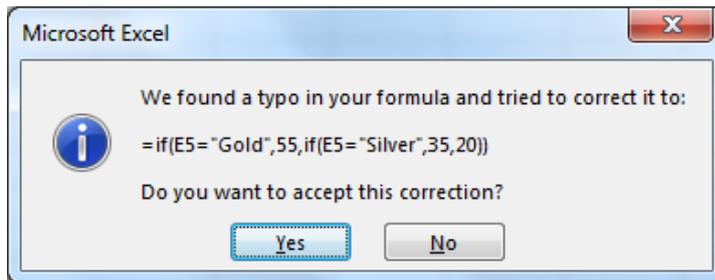
Hash symbols are only displayed for values, and will not appear if the formula result is text that does not fit in the cell.

These hash symbols will also be displayed if a formula returns a negative date or time value. A negative date value is a date preceding 1900.

Number of Parentheses do not Match

A formula must contain an equal number of left and right parentheses. Excel typically will refuse the formula if it identifies a misuse of the parentheses, however Excel can also suggest a correction (see Figure 0-1).

Figure 0-1



Make sure you check the suggested correction before clicking Yes as Excel does not always interpret your formulas correctly.

When parentheses are entered into a formula, each matching set is assigned a colour. Keep an eye on this to ensure you have an equal number, and also that they are typed in the correct order.

Incorrect Number of Commas

Commas are used to separate the arguments in a function. If you enter too many, or too few commas then you will be alerted (see Figure 0-2).

Figure 0-2



Check your formula to ensure that commas have been entered in the correct positions. These kind of typing mistakes will always happen, but you will get efficient and identifying and correcting them.

Formula Text is Displayed Instead of Result

When you press Enter on typing a formula, if the formula text is displayed instead of its result, the cell is probably formatted as text.

To solve this problem, you will need to format the cell correctly and then re-write the formula. The formula will not instantly correct itself when the cell formatting has changed.

Formulas are not Calculating

If you are changing values on a workbook and your formulas are not updating, it could be that formula calculations are set to manual.

This may have been done because a workbook is quite large and constantly calculating formulas takes time.

Or maybe because the workbook contains volatile functions. These are functions such as TODAY and RAND that update even when not involved with the value that was changed.

To calculate all the formulas on a worksheet press **Shift + F9**. To calculate the formulas in the entire workbook press **F9**. These options can also be found on the **Formulas** tab.

To switch back to automatic calculations; click the **Formulas** tab, **Calculation Options** button and then **Automatic**.

Entering Spaces in a Formula

You should not enter spaces in a formula.

A space is used in formulas as the intersection operator. If a space is used in the wrong situation this could result in the #NULL! error value, or the wrong result being displayed.

You should also be careful of extra spaces being entered into cells on a spreadsheet. If you have a formula comparing text entries, or counting the occurrences of text in a list, then it needs to be entered precisely. Any leading or trailing spaces will cause Excel to recognise that text differently.

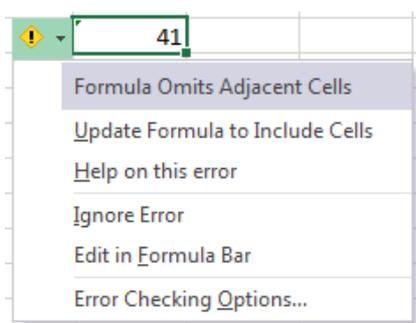
Leading and trailing spaces in a cell can be removed using the TRIM function, or the Flash Fill feature of Excel.

Cell Referencing Problems

Cell referencing problems are a very common cause of formula errors. And there are many different reasons for these problems.

Figure 0-3 shows a user being alerted to a potential problem with their formula. And on further investigation Excel is warning the user about adjacent values that have been omitted.

Figure 0-3



In this example, the user has attempted to sum a range of cells but has not selected the entire range. Excel has noticed the additional values that are omitted from the sum, and is querying your formula.

Another issue that can occur is the incorrect use of absolute, relative and mixed references discussed in the previous chapter.

If when copying a formula you see error values, or results that do not look right, check that references used in the formula are correct.

Using Enter on Array Formulas

When entering (or changing) an array formula you must press **Ctrl + Shift + Enter** when finished to enter the formula. If you only press **Enter**, Excel will not recognise the array formula and will return an incorrect result, or a formula error.

Preventative Measures

Validating Data Entry

On occasions when you see a formula error, the issue may not be with the formula. Quite often the problem originates from the cells that the formula is referencing.

These cells could be formatted incorrectly or contains typos, and these mistakes will affect the functionality of your formulas.

Excel contains a Data Validation tool that can be used to assist data entry on a spreadsheet and limit mistakes.

Lets' look at two examples of its use.

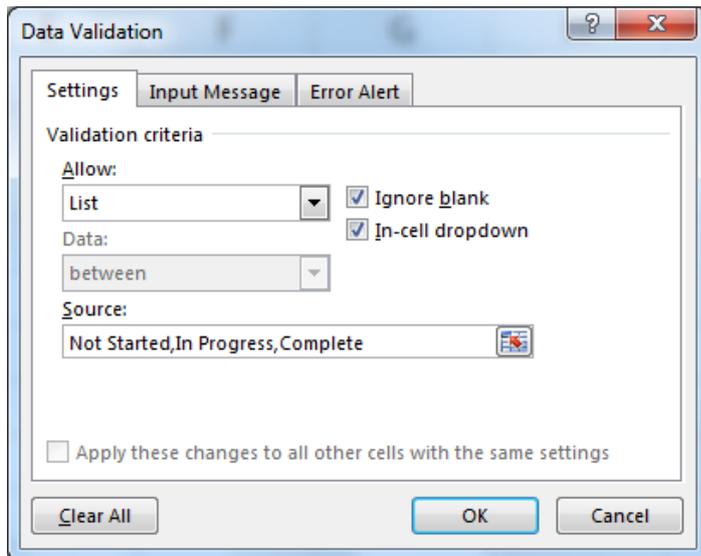
Validate Text Entries by Creating a List

Lets' imagine you have a spreadsheet with a column where the current status of a task is recorded. There are 3 statuses that are used; *Not Started*, *In Progress* and *Complete*.

To ensure that these 3 statuses are input accurately by users, a list can be created so that the user can just select the required status.

1. Select the cells where you want to apply the data validation.
2. Click the **Data** tab on the Ribbon and then **Data Validation**.
3. Select **List** from the **Allow** list.
4. Type the options you want to display in the list into the **Source** box, separating each one with a comma (see Figure 0-4).

Figure 0-4



5. Click **Ok** to create the list. Figure 0-5 shows the list being used to enter data.

Figure 0-5

Task	Status
Task 1	Not Started
Task 2	In Progress
Task 3	

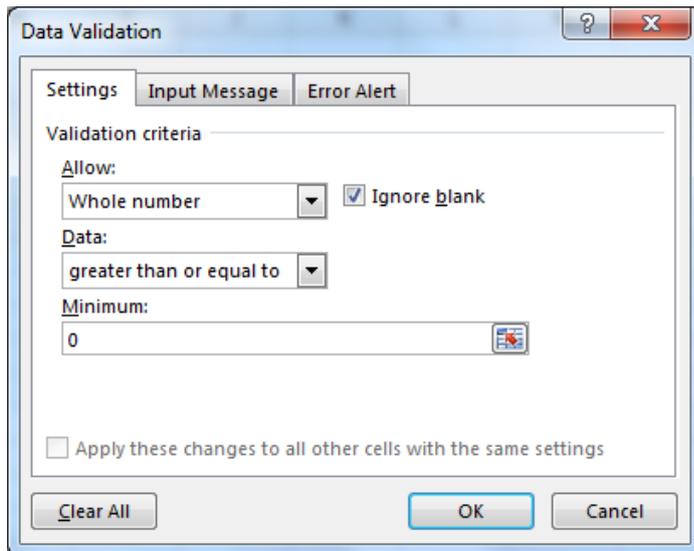
The table shows a dropdown menu for the 'Status' column of 'Task 3' with the following options: Not Started, In Progress, Complete.

Validating the Entry of Values

You can also use Data Validation to limit mistakes on the entry of values. You can validate the entry of integers, decimals, dates, times and even create validation rules using a formula.

In Figure 0-6 a validation rule has been set up to ensure the entry of positive whole numbers only. The benefits of this validation rule are twofold; it ensures that the entry is numeric and also ensures that the number is positive.

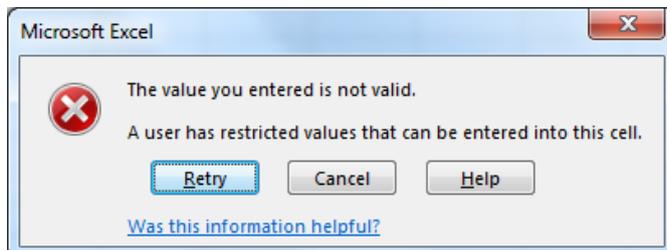
Figure 0-6



Creating your own Error Alert

If a validation rule is violated when entering data, the standard Data Validation error alert is displayed (see Figure 0-7). This informs the user that they made a mistake, but leaves the user in the dark on what went wrong.

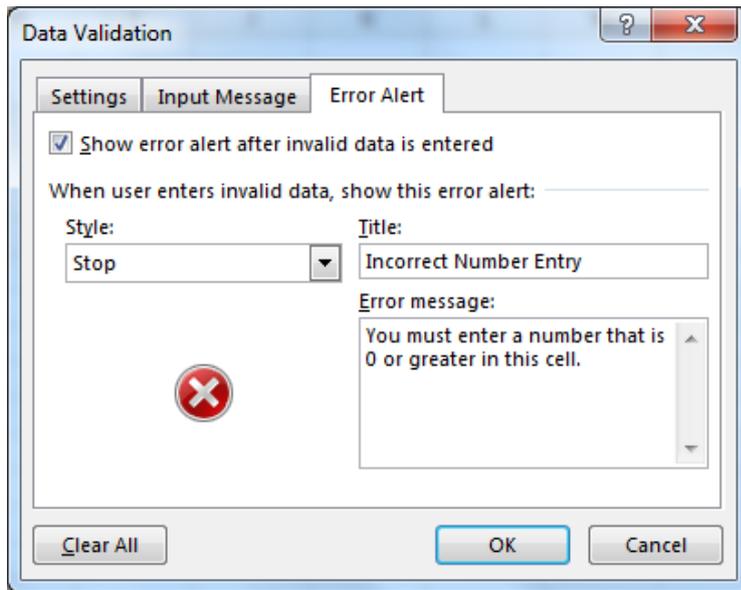
Figure 0-7



Fortunately, you can create your own error alert. This enables you to communicate with the user, explain what is required and where they may have made a mistake.

Click the **Error Alert** tab in the Data Validation dialog and complete the required fields (see Figure 0-8).

Figure 0-8



This error alert is now displayed when a mistake is made on entering data (see Figure 0-9).

Figure 0-9



Hiding Formula Error Messages

Sometimes a formula error message is displayed when there is not really a problem with the formula. This can occur because the formula uses a value that is not available to it yet (maybe a cell has not been completed).

This kind of scenario is typical with lookup formulas and those that need to divide values. However it is not limited to these reasons.

Take the example in Figure 0-10. Cell D4 displays a #DIV/0 error. This is because there have been no sales for product C yet and you cannot divide by 0.

Figure 0-10

	A	B	C	D
1	Product	Sales Total	No of Sales	Av per Sale
2	A	£ 230.00	2	£ 115.00
3	B	£ 400.00	1	£ 400.00
4	C	£ 120.00	0	#DIV/0!
5	D	£ 50.00	3	£ 16.67
6	E	£ 70.00	1	£ 70.00

Although there is not a problem with the formula itself, these error messages are bad news. Therefore you can get the formula to take a different action to the error, such as to hide it. This can be done using the IFERROR function.

The formula below could be entered into cell D2 and copied to the other cells in the 'Av Per Sale' range to hide the error values.

```
=IFERROR(B2/C2,"")
```

Protecting the Formulas on a Sheet

You might want to consider protecting the formulas on a worksheet to prevent them from being accidentally damaged.

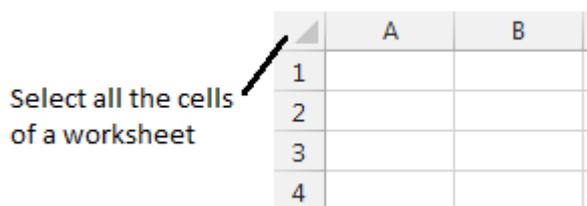
Unfortunately it is a fact of Excel life that people will make mistakes. Formulas do not require regular attention from users. Protecting formulas will ensure they are not vulnerable to accidents when spreadsheets are edited.

By default, all of the cells of a worksheet are locked. This means that when you protect a worksheet, the whole sheet is protected and becomes read only.

To protect only the formulas on a worksheet, you need to unlock all of the cells except the ones containing formulas. To do this, we will first unlock all the cells, and then lock those with formulas again.

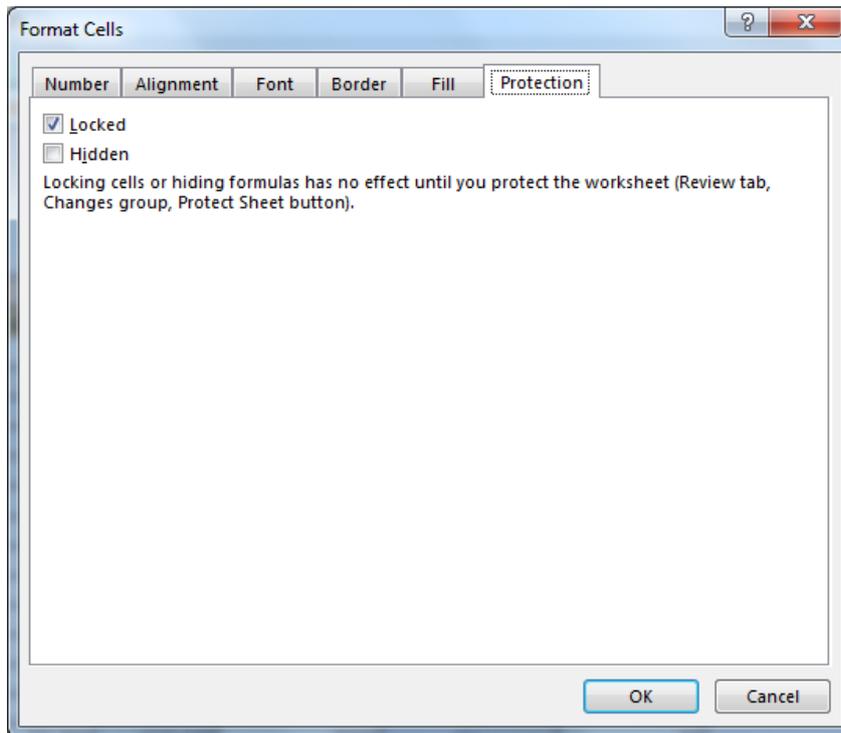
1. Select all the cells of a worksheet by clicking the arrow in the top left corner at the intersection of the columns and rows (see Figure 0-11).

Figure 0-11



2. Open the Format Cells dialog and click the **Protection** tab (see Figure 0-12). Uncheck the **Locked** box and click **Ok**.

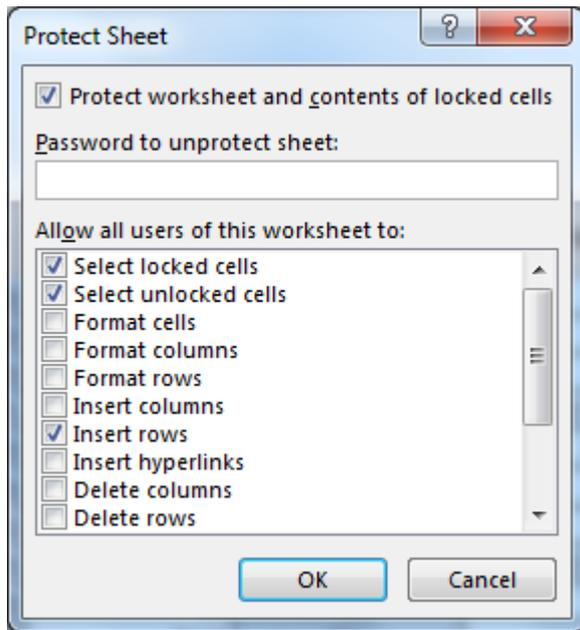
Figure 0-12



3. Select all of the cells on a worksheet that contain a formula by clicking the **Find & Select** button on the **Home** tab, and then **Formulas**.
4. Open the **Protection** tab of the Format Cells dialog, and check the **Locked** box.
5. Now that the formula cells are locked, the sheet can be protected. Click the **Review** tab and then the **Protect Sheet** button.
6. The Protect Sheet dialog opens (see Figure 0-13). Entering a password is optional and is often neglected, as when only protecting against accidents using a password is not necessary. A password will provide better security if required.

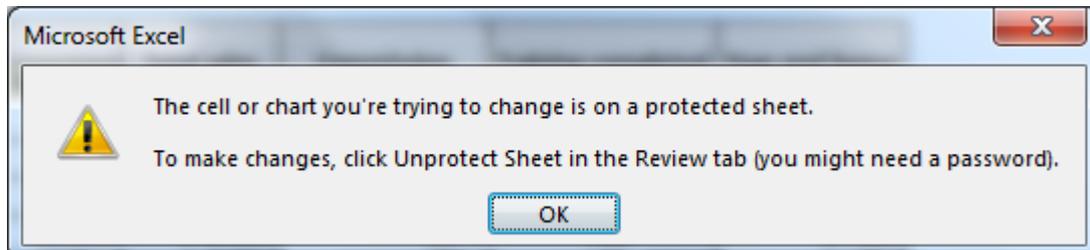
A checklist of common Excel features such as formatting cells, inserting columns and sorting data is shown. This allows you to specify what users can, and cannot, do on a protected worksheet.

Figure 0-13



With the formulas protected. Anybody who tries to edit a cell containing a formula will see the message in Figure 0-14.

Figure 0-14



Although this example focuses on protecting the formulas of a worksheet, there are many ways that protecting a sheet can improve its accuracy.

Protecting a worksheet can also be used to disable functionality such as inserting columns, deleting rows and formatting cells. These steps could help to limit the mistakes that then affect your formulas.

Troubleshooting Formulas

Viewing the Formulas on a Worksheet

When troubleshooting formulas, it can be helpful to have them displayed on screen instead of its value.

To view the formulas;

1. Click the **Formulas** tab on the Ribbon.

2. Click the **Show Formulas** button, or press **Ctrl + `**.

The formula text is now displayed instead of the formula result (see Figure 0-15). The columns of the worksheet are wider to accommodate the formula text.

Figure 0-15

Qtr4	Total sales	Commission	Training completed
2700	=SUM(B9:E9)	=IF(F9>=\$B\$3,F9*\$B\$4,0)	No
2000	=SUM(B10:E10)	=IF(F10>=\$B\$3,F10*\$B\$4,0)	No
3700	=SUM(B11:E11)	=IF(F11>=\$B\$3,F11*\$B\$4,0)	No
3250	=SUM(B12:E12)	=IF(F12>=\$B\$3,F12*\$B\$4,0)	Yes
2700	=SUM(B13:E13)	=IF(F13>=\$B\$3,F13*\$B\$4,0)	Yes
2000	=SUM(B14:E14)	=IF(F14>=\$B\$3,F14*\$B\$4,0)	Yes
3700	=SUM(B15:E15)	=IF(F15>=\$B\$3,F15*\$B\$4,0)	No

Click the **Show Formulas** button again to switch back to viewing the value in each cell.

Tracing Precedents and Dependents

When trying to diagnose the problem with a formula, it can help if you understand the formulas relationship with other cells on the sheet.

Often the problem will be caused by an error with a cell that the formula is using, rather than the formula itself. The error may even be caused by a cell that is used by the cell that the formula is using.

These relationships between cells on a worksheet could be complex. Fortunately Excel provides a formula auditing tool that will visualise these links so they are easier to understand.

There are two types of relationship a cell can have with another cell. They are precedents and dependents.

Cell Precedents: These are the cells that are directly or indirectly used by the formula (see Figure 0-16). Only a cell that contains a formula can have precedents. You may want to see a cell precedents to help identify why a formula is not working correctly.

Cell Dependents: These are the cells dependent upon the cell. You may want to check a cell's dependents before you delete its contents so you know if anything is affected.

Figure 0-16

	A	B	C	D	E	F	G	H	I
1	Commission Report								
2									
3	Sales Target:	£8,500							
4	Commission:	5%							
5	Bonus:	10%							
6									
7		Sales per quarter							
8	Salesperson	Qtr1	Qtr2	Qtr3	Qtr4	Total sales	Commission	Training completed	Year-end bonus
9	Bill MacArthur	£1,500	£1,750	£1,500	£2,700	£7,450	£0	No	£0
10	Jamie Morrison	£3,560	£3,000	£1,700	£2,000	£10,260	£513	No	£0
11	Maureen O'Connor	£4,500	£4,000	£3,500	£3,700	£15,700	£785	No	£0
12	Rebecca Austin	£3,250	£2,725	£3,000	£3,250	£12,225	£611	Yes	£1,223
13	Paul Anderson	£2,520	£2,000	£2,500	£2,700	£9,720	£486	Yes	£972
14	Cynthia Roberts	£1,500	£1,700	£1,800	£2,000	£7,000	£0	Yes	£0
15	Rita Greg	£4,590	£4,050	£4,500	£3,700	£16,840	£842	No	£0
16	Trevor Johnson	£3,000	£3,200	£3,000	£2,250	£12,110	£600	Yes	£1,211
17	Kevin Meyers	£1,790	£1,800	£2,000	£2,200	£7,790	£0	Yes	£0

To view the precedents or dependents for a cell;

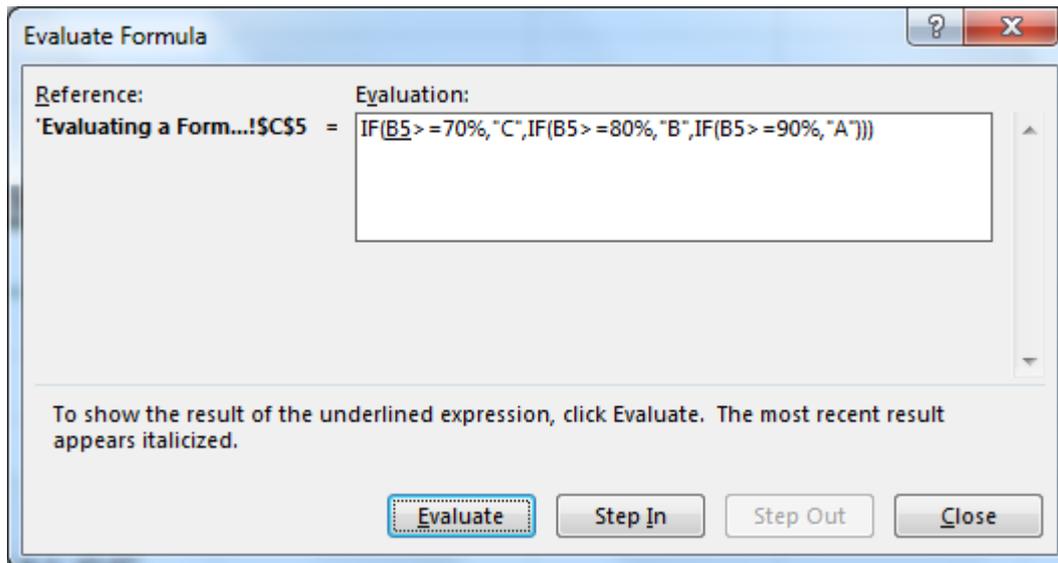
1. Select the cell that you want to audit.
2. Click the **Formulas** tab and then the **Trace Precedents** or **Trace Dependents** button.
3. An arrow is displayed to show the direct precedents or dependents to the cell. If you click the **Trace Precedents** or **Trace Dependents** button again, it will display the next line of precedents or dependents. This can be continued until there are no more to display.
4. Click the **Remove Arrows** button when you have finished.

Evaluating a Formula

The Evaluate Formula button is Excel's main tool for debugging large and complex formulas. It enables you to step through the formula, evaluating each part individually to identify the cause of the problem.

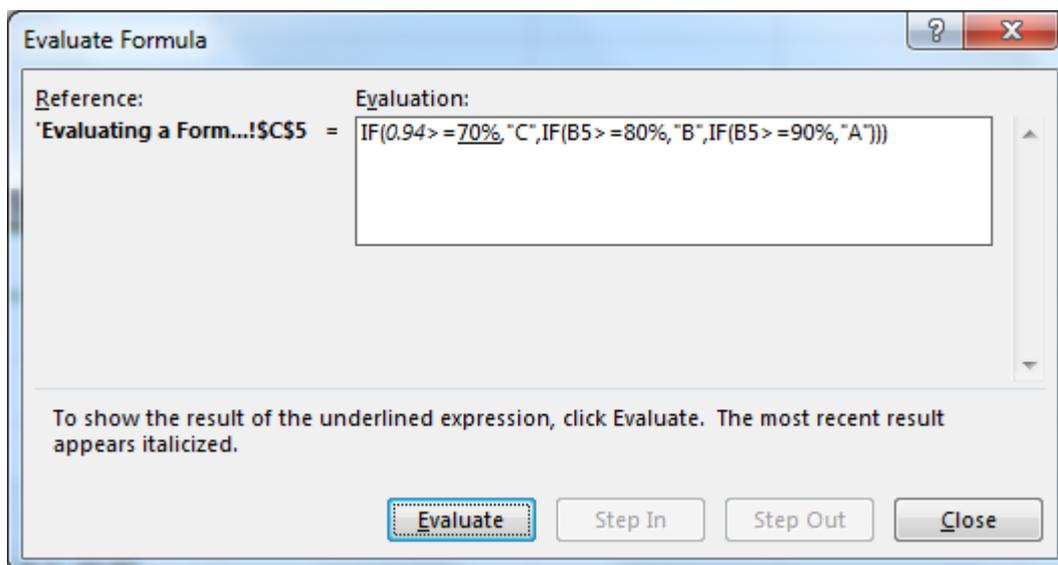
1. Select the cell that you want to evaluate.
2. Click the **Evaluate Formula** button on the **Formulas** tab.
3. The first step of the formula is displayed. The first part to be evaluated is underlined (see Figure 0-17).

Figure 0-17



4. Click the **Evaluate** button to calculate the result of that part and move onto the next.
5. The result is displayed and the next part to be evaluated is underlined (see Figure 0-18).

Figure 0-18



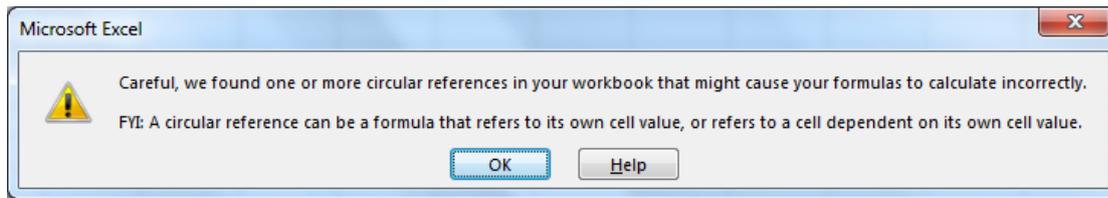
6. Continue until the formula is completely evaluated or you have finished debugging.

Handling Circular References

When working with formulas in Excel you may have come across a circular reference. This is when a formula refers to its own value.

When a circular reference is encountered, a message like the one in Figure 0-19 is displayed.

Figure 0-19



The message explains that the circular reference may have been caused by a direct, or indirect, reference to its own value.

A common example of a circular reference is when an entire column has been used as the reference in a formula, and that formula is written in the column that's being referenced.

For example the formula below, if entered into cell C10 would cause a circular reference.

```
=SUM(C:C)
```

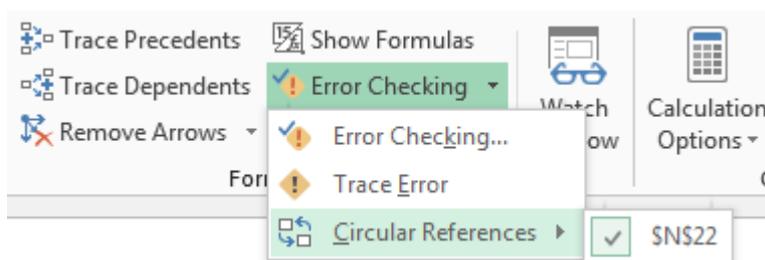
This type of circular reference is quite easy to identify. However if the formula is indirectly referencing its own value, then this can be harder to troubleshoot.

Excel does not stop the use of circular references, as they may be used intentionally. It just warns you of their existence.

If you have a worksheet that contains circular references, and you need to find them.

1. Click the **Formulas** tab.
2. Click the list arrow on the **Error Checking** button.
3. Select **Circular References** (see Figure 0-20).

Figure 0-20



Circular References in a worksheet are also mentioned on the Status Bar at the bottom of the screen (see Figure 0-21).

Figure 0-21



When troubleshooting circular references, it can be useful to turn on the tracer arrows to view the formulas precedents (*Read more on Tracing Precedents and Dependents*). This will help to locate the cause of the circular reference.

Dynamic Array Formulas in Excel

Dynamic array formulas were released in 2020 to Microsoft 365 users of Excel only. They are incredible and have changed the way that many formulas are written.

Using arrays in Excel formulas is not new. They have always been possible, but apart from a few exceptions, you would need to press Ctrl + Shift+ Enter to run them. This gave them the name CSE formulas. They could also be slow and awkward to use, and certainly not dynamic.

This article will explain what you need to know about these dynamic array formulas and how to use them effectively.

Writing a Dynamic Array Formula

Let's look at a simple example of how to write a [dynamic array formula](#).

We have a list of due dates and want to use the IF function to display "yes" if the date is due, and "no" if it is not.

	A	B	C
1	Delivery ID	Due Date	Due?
2	1561	22/06/2020	
3	1341	03/06/2020	
4	1201	31/05/2020	
5	1358	16/06/2020	
6	1018	13/07/2020	
7	1043	17/06/2020	
8	1662	01/06/2020	
9	1784	29/05/2020	
10	1896	10/07/2020	
11	1217	10/06/2020	
12	1379	15/07/2020	
13	1432	29/06/2020	
14	1455	10/07/2020	
15	1456	05/07/2020	

You could write the formula like this;

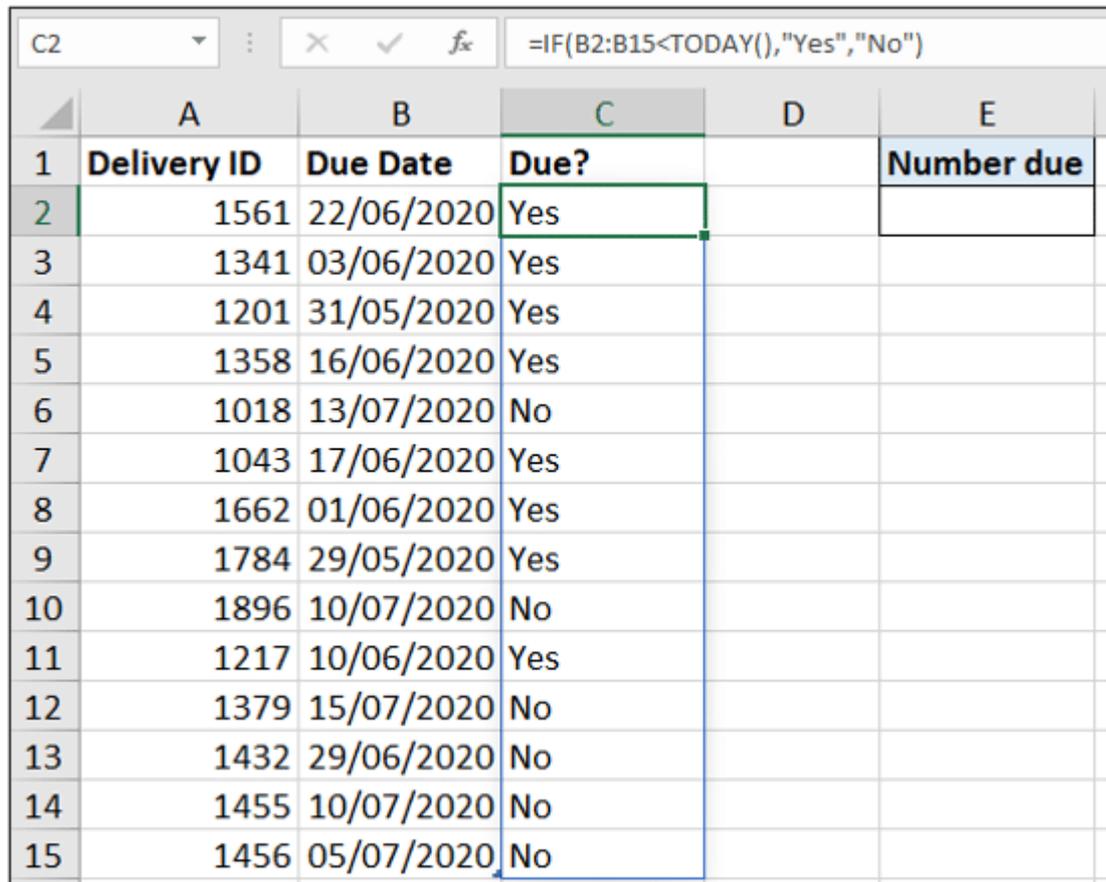
```
=IF(B2<TODAY(),"Yes","No")
```

But this does not take full advantage of the dynamic array engine.

If you write the formula like this;

```
=IF(B2:B15<TODAY(),"Yes","No")
```

It will automatically spill down the other cells in column C to row 15 because that is where the array B2:B15 ends.



	A	B	C	D	E
1	Delivery ID	Due Date	Due?		Number due
2	1561	22/06/2020	Yes		
3	1341	03/06/2020	Yes		
4	1201	31/05/2020	Yes		
5	1358	16/06/2020	Yes		
6	1018	13/07/2020	No		
7	1043	17/06/2020	Yes		
8	1662	01/06/2020	Yes		
9	1784	29/05/2020	Yes		
10	1896	10/07/2020	No		
11	1217	10/06/2020	Yes		
12	1379	15/07/2020	No		
13	1432	29/06/2020	No		
14	1455	10/07/2020	No		
15	1456	05/07/2020	No		

Spill Range and Spill Error

The formula was entered into cell C2 and it spilled down to cell C15.

If you need to edit that formula, you can only do so in cell C2. The other cells do not contain the formula.

From cell C3, in the Formula bar, the formula is visible, but is greyed out and cannot be touched.

=IF(B2:B16<TODAY(),"Yes","No")				
	B	C	D	E
	Due Date	Due?		Number due
51	22/06/2020	Yes		
41	03/06/2020	Yes		
01	31/05/2020	Yes		

From cell C2, the formula remains active and available for editing.

=IF(B2:B16<TODAY(),"Yes","No")				
	B	C	D	E
	Due Date	Due?		Number due
1	22/06/2020	Yes		
1	03/06/2020	Yes		
1	31/05/2020	Yes		

So, the formula can only be changed from the origin cell.

You may have also noticed that the spill range has a blue border to visually show the array perimeter. This is only shown if you click a cell within the spill range.

	A	B	C	D	E
1	Delivery ID	Due Date	Due?		Number due
2	1561	22/06/2020	Yes		
3	1341	03/06/2020	Yes		
4	1201	31/05/2020	Yes		
5	1358	16/06/2020	Yes		
6	1018	13/07/2020	No		
7	1043	17/06/2020	Yes		
8	1662	01/06/2020	Yes		
9	1784	29/05/2020	Yes		
10	1896	10/07/2020	No		
11	1217	10/06/2020	Yes		
12	1379	15/07/2020	No		
13	1432	29/06/2020	No		
14	1455	10/07/2020	No		
15	1456	05/07/2020	No		
16					



If anything was to interfere with the spill range, the #SPILL! error is shown.

	A	B	C	
1	Delivery ID	Due Date	Due?	
2	1561	22/06/2020	#SPILL!	
3	1341	03/06/2020		
4	1201	31/05/2020		
5	1358	16/06/2020		
6	1018	13/07/2020		
7	1043	17/06/2020		
8	1662	01/06/2020		
9	1784	29/05/2020	hello	
10	1896	10/07/2020		

This is fantastic as it makes it easy to notice if any cell in the spill range is affected. It is also easily fixed by removing the interference.

How to Reference the Spill Range

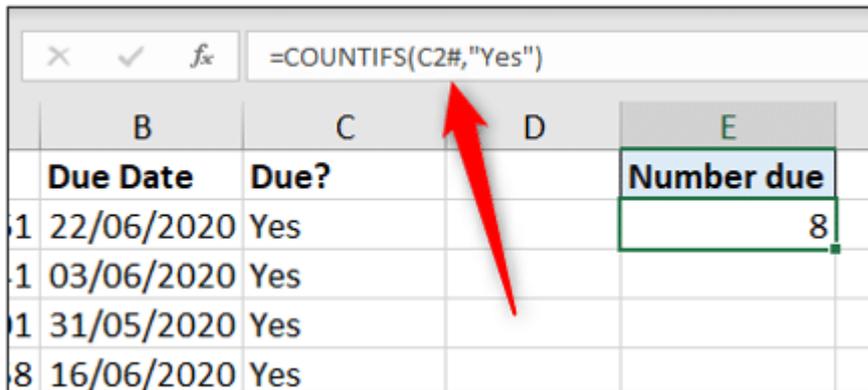
Something else that makes these dynamic array formulas so amazing is that you can reference the spill range.

This means that your formulas are referencing a dynamic range and makes it much easier to create dynamic reports and models in Excel.

For this example, let's use a COUNTIFS function in cell E2 to count the number of deliveries that are due.

When referencing a spill range, the hash (#) sign is used after the origin cell address. So for the range argument of COUNTIFS, C2# is entered.

```
=COUNTIFS(C2#, "Yes")
```



	B	C	D	E
	Due Date	Due?		Number due
1	22/06/2020	Yes		8
1	03/06/2020	Yes		
1	31/05/2020	Yes		
8	16/06/2020	Yes		

Often, if you select the range, this is written in for you.

Dynamic Array Formulas and Table Data

The dynamic array examples so far have been used on cell ranges, but these formulas are more reliable when working with data stored in tables.

Tables automatically expand when new rows or columns are added to them so if our formulas use table data, they too will dynamically expand.

This dynamic behaviour is the real game changer behind these formulas.

The data we have been using in the examples so far is now in a table named *datesdue*.

The following formula would work just like the previous examples.

```
=IF(datesdue[Due Date]<TODAY(), "yes", "No")
```

But now we have a true dynamic array because if another delivery was added to row 16, the IF function automatically spills to the added row.

12	1379	15/07/2020	No	
13	1432	29/06/2020	No	
14	1455	10/07/2020	No	
15	1456	05/07/2020	No	
16	1745	10/08/2020	No	
17				

Dynamic Arrays Cannot Be Used in Tables

Unfortunately, dynamic array formulas cannot be used within tables.

Tables are for storing raw data whilst dynamic arrays are used for creating dynamic outputs from that data.

If you use a dynamic array within a table the #SPILL! error is produced.

With column C included within the table, we are informed that you cannot spill within a table.

The screenshot shows an Excel spreadsheet with a table. The formula bar at the top displays the formula: `=IF([Due Date]<TODAY(),"yes","No")`. The table has columns B (Due Date), C (Due?), D, and E (Number due). Row 51 shows a date of 22/06/2020 in column B, and the corresponding cell in column C contains the error #SPILL!. A context menu is open over the error, with a red arrow pointing to the 'Spill range in table' option.

New Dynamic Array Functions

A bunch of new functions have appeared in Excel to take advantage of this dynamic array behaviour.

These include SORT, SORTBY, SEQUENCE, FILTER, UNIQUE and RANDARRAY.

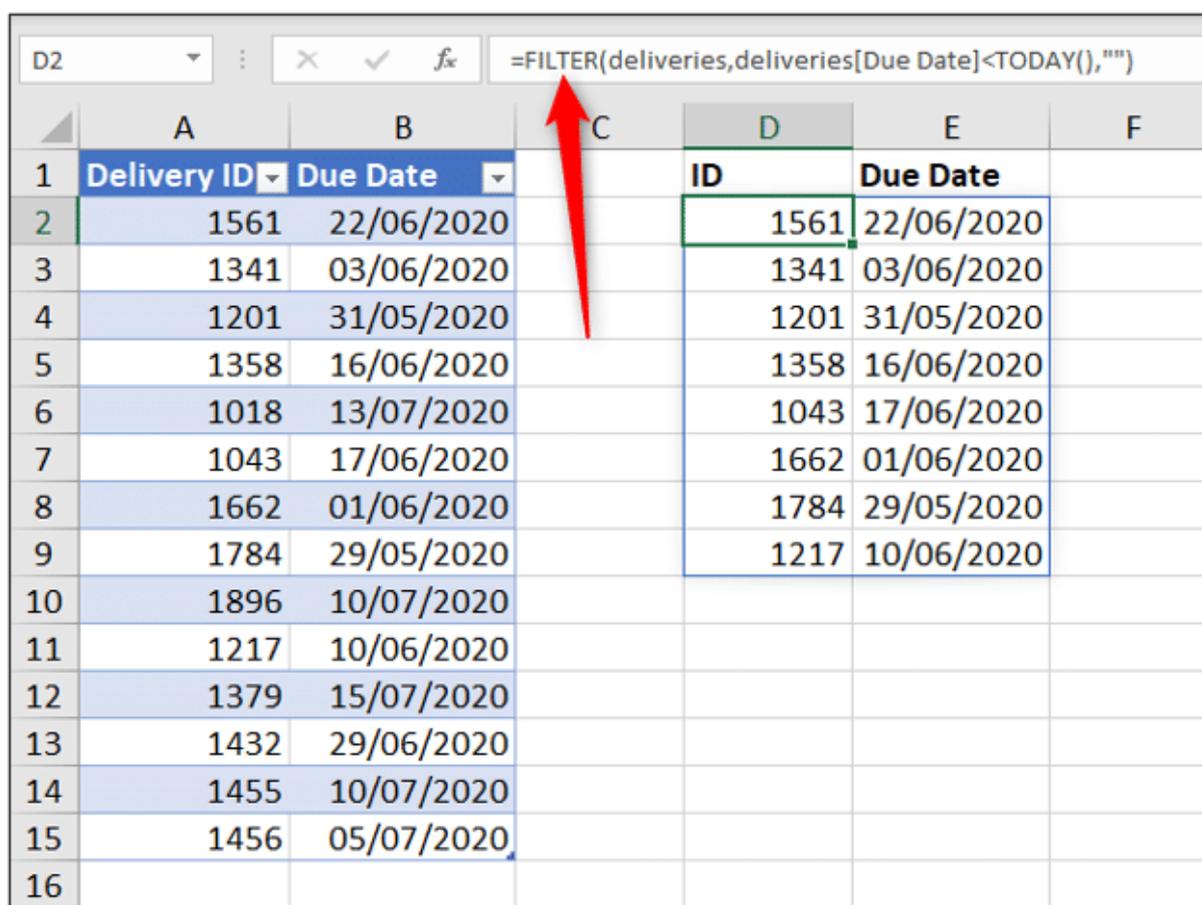
Let's see an example of the FILTER function in action.

Keeping with the same data, it is now stored in a table named *deliveries*.

We will use the FILTER function to filter the list and return the deliveries that are due.

The following formula will do this;

```
=FILTER(deliveries,deliveries[Due Date]<TODAY(),"")
```



The screenshot shows an Excel spreadsheet with a table of deliveries. The formula bar at the top displays the formula `=FILTER(deliveries,deliveries[Due Date]<TODAY(),"")`. A red arrow points from the formula bar to the filtered data in the spreadsheet. The table has columns for Delivery ID, Due Date, ID, and Due Date. The filtered data is shown in columns D and E, with rows 2 through 9.

	A	B	C	D	E	F
1	Delivery ID	Due Date		ID	Due Date	
2	1561	22/06/2020		1561	22/06/2020	
3	1341	03/06/2020		1341	03/06/2020	
4	1201	31/05/2020		1201	31/05/2020	
5	1358	16/06/2020		1358	16/06/2020	
6	1018	13/07/2020		1043	17/06/2020	
7	1043	17/06/2020		1662	01/06/2020	
8	1662	01/06/2020		1784	29/05/2020	
9	1784	29/05/2020		1217	10/06/2020	
10	1896	10/07/2020				
11	1217	10/06/2020				
12	1379	15/07/2020				
13	1432	29/06/2020				
14	1455	10/07/2020				
15	1456	05/07/2020				
16						

This formula returned an array two columns wide to match the *deliveries* table which was provided as the array to filter.

You will still need to format cells just like with any formula, so in this example the date cells were formatted in advance.

It is typical to format a larger range than expected as the formulas are dynamic. And if they expand you want to new cells to be readily formatted.

Dynamic Array Formulas with Other Excel features

Unfortunately, dynamic array formulas cannot be used directly inside Excel features such as Data Validation and Conditional Formatting.

However, you can refer to a spill array from these features. So, the dynamic arrays become a stage for the other features to work from.

For example, we want to create a Data Validation list from the unique values from this table of *countries*.

	A	B	C	D
1	Countries		Unique List	
2	France			
3	Germany			
4	France			
5	UK			
6	France			
7	Spain			
8	USA			
9	Germany			
10	Bulgaria			
11	UK			
12	Norway			
13	India			
14	China			
15				

To get a unique list we can use another of the new dynamic array functions named UNIQUE.

The following formula can be used in cell C2.

```
=UNIQUE(countries[Countries])
```

	A	B	C	D	E
1	Countries		Unique List		Drop Down
2	France		France		
3	Germany		Germany		
4	France		UK		
5	UK		Spain		
6	France		USA		
7	Spain		Bulgaria		
8	USA		Norway		
9	Germany		India		
10	Bulgaria		China		
11	UK				
12	Norway				
13	India				
14	China				
15					

As it uses data in a table, if more countries were added, the formula would dynamically expand and accept them. Equally, if countries were removed, the dynamic array would shrink.

We could take things a step further and add the [SORT function](#) to sort the countries in A to Z order.

	A	B	C	D	E
1	Countries		Unique List		Drop Down
2	France		Bulgaria		
3	Germany		China		
4	France		France		
5	UK		Germany		
6	France		India		
7	Spain		Norway		
8	USA		Spain		
9	Germany		UK		
10	Bulgaria		USA		
11	UK				
12	Norway				
13	India				
14	China				

And then create the Data Validation list from the prepared spill range.

Don't forget to use the spill reference #.

And the Data Validation list is set up.

C	D	E
Unique List		Drop Down
Bulgaria		
China		
France		
Germany		
India		
Norway		
Spain		
UK		
USA		

- Bulgaria
- China
- France
- Germany
- India
- Norway
- Spain
- UK

Contact Details

Email: admin@computergaga.com

LinkedIn: <https://www.linkedin.com/in/alanmurray-computergaga/>

X: <https://www.twitter.com/computergaga1>

Blog: <https://www.computergaga.com>

YouTube: <https://www.youtube.com/computergaga>