

# Sum Product

NEWSLETTER #84 - November 2019

[www.sumproduct.com](http://www.sumproduct.com) | [www.sumproduct.com/thought](http://www.sumproduct.com/thought)



*It's a bumper edition this month!*

It only came out in August, but **XLOOKUP** has already had a makeover in the public domain before it becomes Generally Available – only once before, with the **RANDARRAY** function, has this happened! We provide the details below.

But there's more: there's not one, but two, news articles for Power Query as Microsoft strives to both get it on Mac Excel and improve its analytical capabilities.

Further, the first anniversary of dynamic arrays has been and gone – but it's still not Generally Available – we take a look why that might be below.

Talking of things that have been out a while – we have good news for any American readers considering buying our Financial Modelling book: it's now out in the US through Mr. Excel! If you have been in two minds to buy it before, why not give it a go now..?

With Excel Summits in Bulgaria and Slovenia this month (yours truly will be there), I still have time to convey the regulars: Keyboard Shortcuts, Visual Basics, Power Pivot Principles, Power Query Pointers and the Power BI updates.

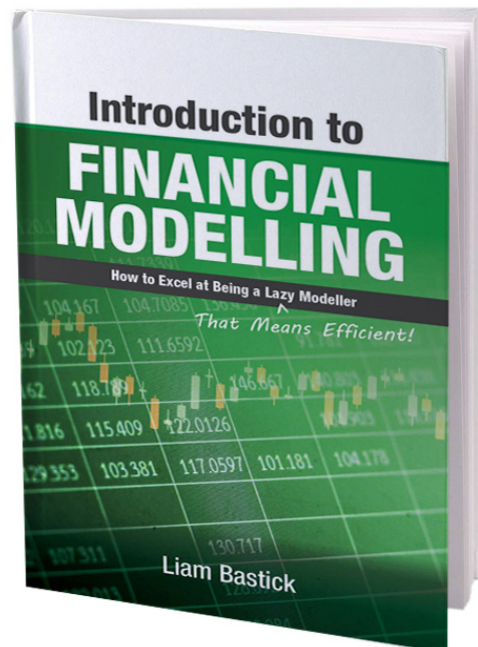
Busy, busy, busy!

*Liam Bastick*, Managing Director, SumProduct



## Financial Modelling Book Now Available in the US

Thanks for being patient. SumProduct is proud to announce, in conjunction with Mr. Excel, our debut book on financial modelling is now available directly in the United States.



Now in a US reprint, SumProduct's flagship book, *An Introduction to Financial Modelling* by Liam Bastick, is more readily available for the US and elsewhere than its original Australian counterpart. We even managed to keep some of the original spelling too...

You can read more about the book at [www.mrexcel.com/products/introduction-to-financial-modelling/](http://www.mrexcel.com/products/introduction-to-financial-modelling/) and buy it at [www.mrexcel.com/store/index.php?l=product\\_detail&p=372](http://www.mrexcel.com/store/index.php?l=product_detail&p=372).

So what's stopping you!?



## Bulgaria Excel Days 2019 Reminder



It's nearly upon us. With 10 lectures by leading world experts in Microsoft Excel, Power BI, Data Analysis and Visualisation, Sofia's Inter Expo Center is the place to be on Thursday 21st November. With six masterclasses and opportunities for business networking, this major event is likely to be another huge success. Don't let the fact SumProduct's **Liam Bastick** will be in attendance put you off!

Last year saw more than 250 delegates attend – and more is expected this year. For more details and to register, check out the website at [www.exceldays.itraining.bg/en/home-en/](http://www.exceldays.itraining.bg/en/home-en/).

## Excel Olympics Conference & Masterclasses, Slovenia – Another Reminder

The following week (Tuesday 26th and Wednesday 27th November) sees a similar event at the Austria Trend Hotel in Ljubljana, capital of Slovenia – the only country in the world to have “love” in its name.

**Conference and Masterclasses**  
26. and 27.11.2019, Ljubljana, Slovenia

Special offer: **Buy 4, Get 1 Free**

<https://conference.excelolympics.com>

Advanced formulas & functions  
DAX  
Dashboards  
Financial analysis  
Master your data

Liam Bastick   Leila Gharani   Matt Allington   Charles Williams   Ken Puls

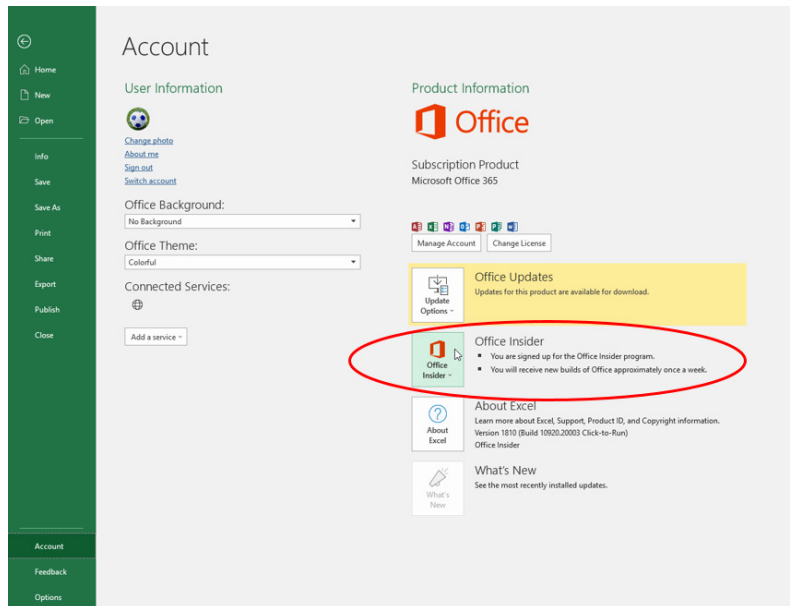
Five murky looking presenters (including that SumProduct guy again) depicts another expert group present Excel / Power BI sessions and masterclasses. For more details and to register, check out [www.conference.excelolympics.com](http://www.conference.excelolympics.com).

## Dynamic Arrays: One Year On

It's 12 months since Microsoft rocked the calculation boat with its new range of dynamic array functions and features. Those of us lucky enough to get it in Excel Insider Fast raved incessantly. But what's actually happened one year later..? We take a look.

365 days later and dynamic arrays are still in “Preview” mode, *i.e.* it's not

yet “Generally Available” but it has moved into all of Office 365 Insider Fast and some of Insider Slow and Monthly Targeted (version 1906 Builds 11901.20080 and later). You can register for Insider in **File** -> **Account** -> **Office Insider** in Excel's backstage area.



The world has now realised that it won't be coming to the perpetual licence versions such as Excel 2016 and Excel 2019. That wasn't immediately clear, but users have come to realise this. It's not so much that it has been in Preview for so long – we get that the calculations are revolutionary and there are still issues to work out such as:

- being able to have totals at the bottom of spilled arrays
- allowing spilled arrays to accept formatting
- having calculations generate the same results if calculated in a different order (*see below*).

It's the fixation with Office 365 only that worries me. Whilst this helps the clamour for the subscription model, this may be a mistake. Modellers are going to be loath to use functions and features that the majority of

end users will not be able to take advantage of. It may mean adoption will be slow for features people should be using in their millions. Similar judgment calls were made with Power Pivot, and eventually, Microsoft relented and made it available to all in Windows versions of Excel. I can't help but think a similar perspective should be taken here.

Currently, we therefore have two types of Excel:

1. **DA Excel:** Excel that supports dynamic arrays, its functions and features
2. **Legacy Excel:** the “traditional” Excel that is still wrapped in the world of **CTRL + SHIFT + ENTER** and does not support dynamic arrays.

This leads to compatibility issues.

### Compatibility Issue #1: @ and SINGLE

You may recall the **SINGLE** function that was released last year.

In the past, if you entered **=A\$1:A\$10** anywhere in rows 1 through 10, the formula would return only the value from that row. In DA Excel, typing this formula would create a Spilled Array Formula. To protect existing formulae, the **SINGLE** function was created to return a single value using logic known as implicit intersection. **SINGLE** may return a value, single cell range or an error:

**=SINGLE(value).**

For example, the two **SINGLE** formulae here are supplied a range, **H13:H27**, and return the values in cells **H17** and **H22** respectively, corresponding to the rows that were held in common:

First Name	Last Name	Points
Ivan	Idea	717
Amanda	Hugankiss	885
Artie	Detoo	976
Blake	Seven	247
Piper	Pied	978
Ivana	Tinkle	508
Artie	Chokes	300
Mike	Stand	778
Shelley	Ack	954
Blade	Runner	203
Sheikh	Spear	711
Mike	Robe	305
Daley	News	839
Hugo	There	611
Mimi	Selfish	197

**SINGLE**

**978**

**203**

Later, after its initial release, **SINGLE** was replaced with **@** as follows:

The screenshot shows an Excel spreadsheet with a table of names and points. The formula bar at the top displays '=H13:H27'. To the right of the table, there is a cell containing '@'. Below the table, there are two cells containing the values 978 and 203, which correspond to the points for Piper and Blade respectively.

First Name	Last Name	Points
Ivan	Idea	717
Amanda	Hugankiss	885
Artie	Detoo	976
Blake	Seven	247
Piper	Pied	978
Ivana	Tinkle	508
Artie	Chokes	300
Mike	Stand	778
Shelley	Ack	954
Blade	Runner	203
Sheikh	Spear	711
Mike	Robe	305
Daley	News	839
Hugo	There	611
Mimi	Selfish	197

Now, I mention this history with good reason. Excel will only remove **@** from a formula where previous Excel versions would have used implicit intersection (as described above) to return a single value from a range, a named range or function parameter.

On the positive side, if you attempt to enter such a formula, Excel will warn you and do its utmost to stop you. It is still possible to cause an issue though. For example, in DA Excel, you could create the following formula:

The screenshot shows an Excel spreadsheet with two columns: 'Value' and 'DA Formula'. The formula bar at the top displays '=A2'. The 'Value' column contains numbers 1 through 5, and the 'DA Formula' column contains the same numbers. A red error message '=A2' is visible in cell E2.

	A	B	C	D	E
1	Value		DA Formula		
2	1		1		=A2
3	2		2		
4	3		3		
5	4		4		
6	5		5		

In Legacy Excel, this would appear as:

The screenshot shows an Excel spreadsheet with two columns: 'Value' and 'DA Formula'. The formula bar at the top displays '=\_xlfm.SINGLE(A2)'. The 'Value' column contains numbers 1 through 5, and the 'DA Formula' column contains the same numbers. A red error message '=\_xlfm.SINGLE(A2)' is visible in cell E2.

	A	B	C	D	E	F
1	Value		DA Formula			
2	1		1		=_xlfm.SINGLE(A2)	
3	2		2			
4	3		3			
5	4		4			
6	5		5			

Notice the error message is **=\_xlfm.SINGLE(A2)**, not **=\_xlfm.@(A2)**. This is confusing if you don't know the history of the **@** operator. Worse comes if you try to evaluate this formula:

The screenshot shows an Excel spreadsheet with two columns: 'Value' and 'DA Formula'. The formula bar at the top displays '=\_xlfm.SINGLE(A2)'. The 'Value' column contains numbers 1 through 5, and the 'DA Formula' column contains the same numbers. A red error message '#NAME?' is visible in cell C2, and another red error message '=\_xlfm.SINGLE(A2)' is visible in cell E2.

	A	B	C	D	E	F
1	Value		DA Formula			
2	1		#NAME?		=_xlfm.SINGLE(A2)	
3	2		#NAME?			
4	3		#NAME?			
5	4		#NAME?			
6	5		#NAME?			

It generates an **#NAME?** error, which is far from ideal.



### Compatibility Issue #2: Spill References

A formulaic reference to an entire dynamic array may use the spill reference suffix operator (that's a mouthful), e.g.

	A	B	C	D	E	F	G	H	I
1	Value	DA Formula					Sum		
2	1	1		=A2:A6			15	=SUM(C2#)	
3	2	2							
4	3	3							
5	4	4							
6	5	5							

The formula =SUM(C2#) sums the entire spilled range emanating from cell C2. Unfortunately, this is not converted as you might hope in Legacy Excel:

	A	B	C	D	E	F	G	H	I	J	K
1	Value	DA Formula					Sum				
2	1	1		{=A2:A6}			15	=SUM(_xlfn.ANCHORARRAY(C2))			
3	2	2									
4	3	3									
5	4	4									
6	5	5									

The formula in cell G2 is now =SUM(\_xlfn.ANCHORARRAY(C2)) and, again, will not evaluate, viz.

	A	B	C	D	E	F	G	H	I	J	K
1	Value	DA Formula					Sum				
2	1	1		{=A2:A6}			#NAME?	=SUM(_xlfn.ANCHORARRAY(C2))			
3	2	2									
4	3	3									
5	4	4									
6	5	5									

### Compatibility Issue #3: Uninvited CTRL + SHIFT + ENTER (CSE) Array Formulae

Take another look at that last example:

	A	B	C	D	E	F
1	Value	DA Formula				
2	1	1		=A2:A6		
3	2	2				
4	3	3				
5	4	4				

In DA Excel, the formulae in cells C2:C6 contain the spilled formula =A2:A6. In Legacy Excel, these formulae have been replaced as follows:

	A	B	C	D	E	F
1	Value	DA Formula				
2	1	1		{=A2:A6}		
3	2	2				
4	3	3				
5	4	4				
6	5	5				

The formula `{=A2:A6}` (created by using **CTRL + SHIFT + ENTER**) has been entered in each cell in cells **C2:C6**. This is because any formula that DA Excel thinks could return an array (even if it is actually only returning a single value) is converted to a **CTRL + SHIFT + ENTER** (CSE) array formula in previous versions. This can impact memory and is creating unnecessarily complex formulae. This can cause several issues:

- a single cell containing a non-spilling formula is made unnecessarily complex and may not evaluate correctly
- a DA Excel dynamic array formula that is either spilling or blocked (**#SPILL!**) will be converted to a fixed CSE formula in earlier versions of Excel
- a formula entered in DA Excel as a multi-cell CSE formula may have backward compatibility issues.

You need to take great care when working with such instances.

### Another Issue: Calculation Order Concern

There is another problem that is an issue in DA Excel. When I calculate something in Excel, if I use the same formula, I must get the same answer, right? Well – not necessarily. Consider the following:

The screenshot shows two examples of calculations in Excel. Both examples use the `SEQUENCE` and `TRANSPOSE` functions. In both, 'Value for Formula 1' is 3 and 'Value for Formula 2' is 4. In 'Calculation 1', Formula 1 (cell H12) is 3 and Formula 2 (cell H13) is 4, resulting in a `#SPILL!` error in cell K16. In 'Calculation 2', Formula 1 (cell H27) is 3 and Formula 2 (cell H28) is 4, resulting in a `#SPILL!` error in cell I33. The formulas are `=SEQUENCE(H12)` and `=TRANSPOSE(SEQUENCE(H13))` for Calculation 1, and `=SEQUENCE(H27)` and `=TRANSPOSE(SEQUENCE(H28))` for Calculation 2.

In the example above, Calculations 1 and 2 are identical but deliver different results (*i.e.* different **#SPILL!** errors). Why?

- In Calculations 1 and 2, both values for Formula 1 and Formula 2 were originally set to 1. This causes no **#SPILL!** errors
- In Calculation 1, the value for Formula 2 (cell **H13**) was then changed to 4 with no error
- Then, in Calculation 1, the value for Formula 1 (cell **H12**) was changed to 3. This caused the resultant **#SPILL!** error in cell **K16**
- Next, in Calculation 2, the value for Formula 1 (cell **H27**) was changed to 3 with no error
- Then, in Calculation 2, the value for Formula 2 (cell **H28**) was changed to 4. This caused the resultant **#SPILL!** error in cell **I33**.

I am not sure what the solution is for this problem. Technically, **#SPILL!** is working correctly, but it doesn't seem right that two results may be generated in this instance depending upon what input I change first. The jury remains out on this one.

### Not All Gloom and Doom

**RANDARRAY** is one of the original functions in DA Excel. Because it's not yet Generally Available, it's had a facelift in the last six months. This is an important thing to note – this was the first time *ever* a function had been released publicly by Microsoft and then had its syntax changed. This is something that is possible to do before a function or feature becomes Generally Available – "Preview" means Microsoft reserves the right to change something as they see fit. That's a *good* thing here.

Originally, the **RANDARRAY** function returned an array of random numbers between 0 and 1. However, there was a general sense of underwhelm with this function and the new and improved version has just been released. It now allows you to set your own maximum and minimum and decide whether you want the values returned to be decimals (*e.g.* 17.4381672...) or integers (whole numbers).

The new syntax for the function is now as follows:

**=RANDARRAY([rows], [columns],[min],[max],[integer]).**

The function has five arguments, all supposedly optional (but upon testing, we weren't quite as convinced):

- **rows**: this specifies how many **rows** the results should spill over. If omitted, the default value is 1
- **columns**: this specifies how many **columns** the results should spill over. If omitted, the default value is also 1
- **min**: this is the minimum value that may be selected randomly. If this is not specified, it is assumed to be zero (0)
- **max**: this is the maximum value that may be selected randomly. If this is not specified, it is assumed to be 1
- **integer**: if this is set to TRUE, only integer outputs are allowed; the default value (FALSE) provides non-integer (decimal) results.

Other points to note:

- if **rows** or **columns** refers to a blank cell reference, this will generate the new **#CALC!** error
- if **rows** or **columns** are entered as decimals, the values used will be truncated to the number before the decimal point (e.g. 3.9999999 will be treated as 3)
- if **rows** or **columns** is a value less than 1, **#CALC!** will be returned
- if **integer** is set to TRUE and either **min** or **max** is not an integer, this will generate an **#VALUE!** error
- **max** must be greater than or equal to **min**, else the error **#VALUE!** is returned.

When we originally discussed the **RANDARRAY** function, we used this rather comprehensive example to create a list of random integers between two values:

Excel screenshot showing the formula in cell F44: `=ROUNDDOWN(RANDARRAY(H36,H37)*(H39-H38+1),0)+INT(H38)`. The inputs are: No. of Rows: 5, No. of Columns: 8, Start Number: 1, End Number: 10. The output is a 5x8 grid of random integers.

6	2	1	3	8	9	1	2
6	10	8	3	1	6	4	2
8	7	5	7	3	4	2	10
6	4	1	5	5	10	4	2
3	9	6	1	1	10	10	9

Originally, the formula in cell **F44** was

**=ROUNDDOWN(RANDARRAY(H36,H37)\*(H39-H38+1),0)+INT(H38)**

and my past article explained how this worked. However, it's much easier now:

Excel screenshot showing the formula in cell F45: `=RANDARRAY(H36,H37,H38,H39,H40)`. The inputs are: No. of Rows: 5, No. of Columns: 8, Start Number: 1, End Number: 10, Integer: TRUE. The output is a 5x8 grid of random integers.

1	2	8	6	1	7	8	2
1	1	2	1	6	1	3	10
2	8	7	4	7	5	5	1
8	10	10	4	6	10	9	7
3	8	4	4	4	3	6	3

The “new improved” formula in cell **F45** (it's moved down a row due to the additional argument required in cell **H40**) is simply

**=RANDARRAY(H36,H37,H38,H39,H40).**

This is much simpler – and pretty cool.

### Impact of **XLOOKUP** and **XMATCH**

Since the announcement of dynamic arrays, there has been another major announcement of two new functions (which we covered last month): **XLOOKUP** and **XMATCH**. At the time of writing, these two functions are both in Preview mode in Office 365 Insider Fast only.

**XLOOKUP** has the following syntax:

**XLOOKUP(lookup\_value, lookup\_vector, results\_array, [match\_mode], [search\_mode])**

On first glance, it looks like it has too many arguments, but often you will only use the first three:

- **lookup\_value**: this is required and defines what value you want to look up
- **lookup\_vector**: this reference is required and is the row or column of data you are referencing to look up **lookup\_value**
- **results\_array**: this is where the corresponding item is you wish to return and is also required (even if it is the same as **lookup\_vector**). This does not have to be a vector (*i.e.* one row or one column of cells): it may be an array (with at least two rows and at least two columns of cells). The only stipulation is that the number of rows / columns must equal the number of rows / columns in the column / row vector – but more on that later
- **match\_mode**: this argument is optional. There are four choices:
  - o **0**: exact match (default)
  - o **-1**: exact match or else the largest value less than or equal to **lookup\_value**
  - o **1**: exact match or else smallest value greater than or equal to **lookup\_value**
  - o **2**: wildcard match. You should use the special character ? to match any character and \* to match any run of characters.

What's impressive, though, is that for certain selections of the final argument (**search\_mode**), you don't need to put your data in alphanumerical order! As far as I am aware, this is a first for Excel

- **search\_mode**: this argument is also optional. There are again four choices:
  - o **1**: search first to last (default)
  - o **-1**: search last to first
  - o **2**: what is known as a binary search, first to last (requires **lookup\_vector** to be sorted). Just so you know, a binary search is a search algorithm that finds the position of a target value within a sorted array. A binary search compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value, and repeating this until the target value is found
  - o **-2**: another binary search, this time last to first (and again, this requires **lookup\_vector** to be sorted).

**XLOOKUP** is a powerful substitute for both **VLOOKUP** and **INDEX MATCH**:

	B	C	D	E	F	G	H	I	J	K	L	M	N
35													
36													
37													
38													
39													
40													
41													
42													
43													
44													
45													
46													
47													
48													
49													
50													
51													
52													
53													
54													
55													
56													
57													
58													
59													

You can clearly see the **XLOOKUP** function is shorter:

**=XLOOKUP(H52,F41:F47,G41:G47)**

Only the first three arguments are needed, whereas **VLOOKUP** requires both a fourth argument, and, for full flexibility, the **COLUMNS** function as well. **XLOOKUP** will automatically update if rows / columns are inserted or deleted. It's just *simpler*.

Further, the sister function **XMATCH** has the following syntax:

**XMATCH(lookup\_value, lookup\_vector, [match\_mode], [search\_mode])**

where:

- **lookup\_value**: this is required and defines what value you want to look up
- **lookup\_vector**: this reference is required and is the row or column of data you are referencing to look up **lookup\_value**
- **match\_mode**: this argument is optional. There are four choices:
  - o **0**: exact match (default)
  - o **-1**: exact match or else the largest value less than or equal to **lookup\_value**
  - o **1**: exact match or else smallest value greater than or equal to **lookup\_value**
  - o **2**: wildcard match. You should use the special character ? to match any character and \* to match any run of characters.

Again, for certain selections of the final argument (**search\_mode**), you don't need to put your data in alphanumerical order

- **search\_mode**: this argument is also optional. There are again four choices:
  - o **1**: search first to last (default)
  - o **-1**: search last to first
  - o **2**: this is a binary search, first to last (requires **lookup\_vector** to be sorted)
  - o **-2**: another binary search, this time last to first (and again, this requires **lookup\_vector** to be sorted).



As you can see, it's a fairly straightforward addition to the **MATCH** family. It acts similarly to **MATCH** – just with heaps more functionality.

	B	C	D	E	F	G	H	I	J	K
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										

It's clear to see both of these functions would work well with dynamic arrays and spilled references. Since the announcement of these two formulae, people have been knocking down the door to use them in a Generally Available environment.

### Word to the Wise

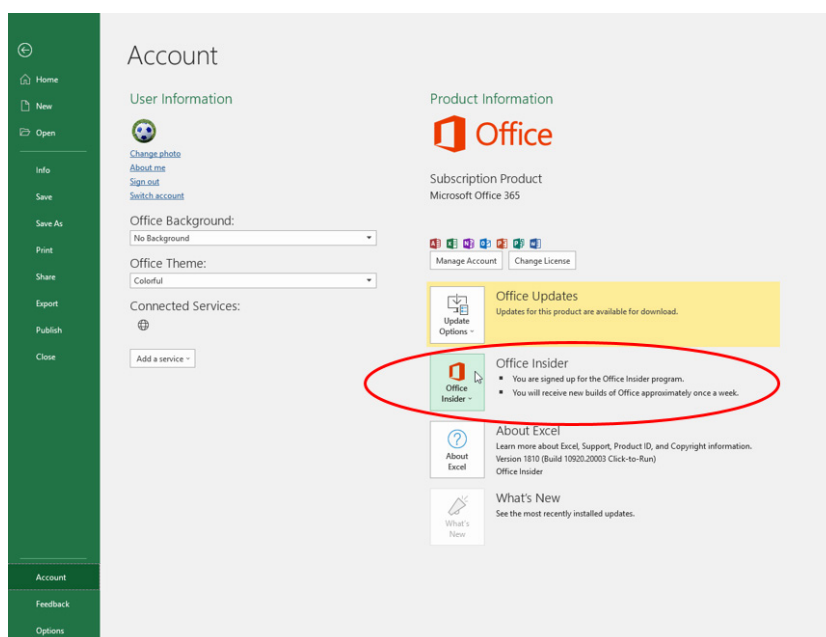
I know no more than you, dear reader, but I have an inkling that Microsoft might be changing things up shortly. Dynamic arrays were publicly announced at Microsoft Ignite last year. This same conference is to be held in Florida in early November this year. I can't help thinking...

## XLOOKUP Revised: And Then There Were Six...

It was only in late August 2019 that Microsoft added **XLOOKUP** to its lookup fold, and already it has had a makeover. Back in February, **RANDARRAY** became the first function ever to have a syntax change after release – and now **XLOOKUP** joins this elite group of two.

### Revising XLOOKUP

As we explained last month, like its sister function **XMATCH**, **XLOOKUP** is still only available in the "Office Insider" programme, which is an Office 365 fast track. You can register in **File -> Account -> Office Insider** in Excel's backstage area:



Please note that there is still some debate as to whether all Insiders Fast subscribers have access to these functions.

I think Microsoft has decided to make a change before pulling the pin and making both functions Generally Available. The current line of thinking is that there should be an error trap for when a value cannot be found. This has meant a sixth argument has now been added:

**XLOOKUP(lookup\_value, lookup\_vector, results\_array, [match\_mode], [search\_mode], [if\_not\_found])**

It still means most of the time you will only require the first three arguments:

- **lookup\_value**: this is required and defines what value you want to look up
- **lookup\_vector**: this reference is required and is the row or column of data you are referencing to look up lookup\_value
- **results\_array**: this is where the corresponding item is you wish to return and is also required (even if it is the same as lookup\_vector). This does not have to be a vector (*i.e.* one row or one column of cells): it may be an array (with at least two rows and at least two columns of cells). The only stipulation is that the number of rows / columns must equal the number of rows / columns in the column / row vector – but more on that later
- **match\_mode**: this argument is optional. There are four choices:
  - o **0**: exact match (default)
  - o **-1**: exact match or else the largest value less than or equal to **lookup\_value**
  - o **1**: exact match or else smallest value greater than or equal to **lookup\_value**
  - o **2**: wildcard match. You should use the special character ? to match any character and \* to match any run of characters.

What's impressive, though, is that for certain selections of the final argument (search\_mode), you don't need to put your data in alphanumerical order! As far as I am aware, this is a first for Excel

- **search\_mode**: this argument is also optional. There are again four choices:
  - o **1**: search first to last (default)
  - o **-1**: search last to first
  - o **2**: what is known as a binary search, first to last (requires **lookup\_vector** to be sorted). Just so you know, a binary search is a search algorithm that finds the position of a target value within a sorted array. A binary search compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value, and repeating this until the target value is found
  - o **-2**: another binary search, this time last to first (and again, this requires lookup\_vector to be sorted)
- **if\_not\_found**: this new optional argument allows you to replace the usual return of #N/A with something more informative like an alternative formula, text or a value.

Having this error case as the final argument is consistent with other existing functions such as **IFERROR** and **IFNA**, as well as the dynamic array function **FILTER**. It seems like a good idea as having this final argument can tell you when something in your spreadsheet is not working as you might think.

It should be noted that this additional argument *won't* be coming to **XMATCH**. Whilst these functions came out at the same time and work well together, there are some subtle differences:

- **XLOOKUP** could retrieve #N/A from the return range even though the search item was found, so #N/A does not necessarily mean “not found”. There is no such ambiguity with **XMATCH**. Having this distinction for **XLOOKUP** is therefore useful
- It's likely **XMATCH** will be commonly used with **INDEX**, instead of **MATCH**. An **[if\_not\_found]** argument does not help here as users will require the error trap test outside of the **INDEX** calculation.

## Get & Transform (Power Query) in Excel for Mac: Phase 1



For several years now, the Windows platform has introduced a powerful set of Get & Transform Data / Power Query tools, starting way back with Excel 2010. We have a Power Query blog each Wednesday that keeps you apprised of all the things you can do with the Extract, Transform and Load (ETL) tool. But it's been missing from Mac Excel.

Back in June, there was a little teaser of what was to come with the ability to undertake limited Power Query query refreshes for CSV, JSON, Text, XLSX and XML files only. Well, just recently, Microsoft has announced the first step towards full support for Power Query in Excel for Mac.

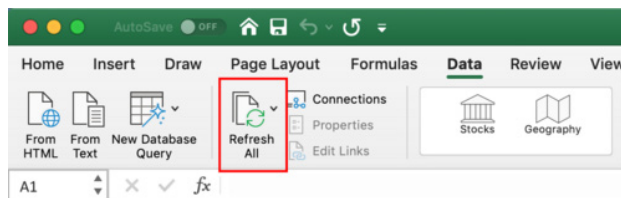
*[Given they have officially called it “Power Query in Excel for Mac” does this mean the Get & Transform moniker is to be retired..? We shall have to wait and see!]*

This “Phase 1” release allows you to refresh your Power Query queries on Mac from Text, CSV, XLSX, XML and JSON files (yes, we’ve just said most of that has been around from June), but you can author queries with VBA.

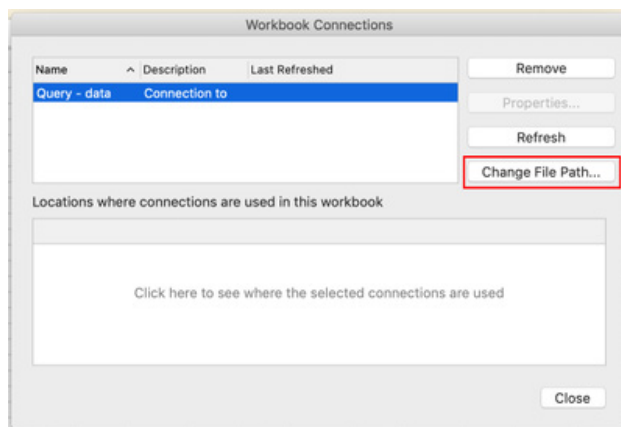
This new functionality is live for all Office 365 subscribers in Excel for Mac running version 16.29 (19090802) and later. If you are running an earlier version, you will need to update.

To refresh queries generated from Text, CSV, XLSX, XML or JSON files, triggering the Refresh command is done the same way as previously, e.g.

- Click on the Data tab of the Ribbon, then choose ‘Refresh All’
- Right-click on your query table, then click Refresh
- Right click on your PivotTable, then click ‘Refresh Data’
- Use a VBA script
- etc.



It should be noted that the first time you try to refresh your workbook queries, you may need to update the location of the data source so that it works on your Mac. To do this, **click the Data tab on the Ribbon -> Connections -> select the desired connection -> click Change File Path button** to update it.

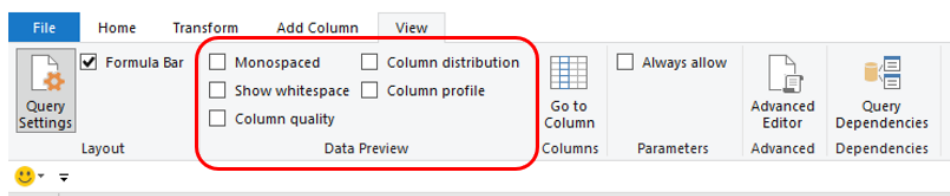


Furthermore, you may also create and manage Power Query queries in your workbook using VBA. Any existing macros and VBA scripts that reference **Workbook.Queries** and **WorkbookQuery** entities in the Excel's object model will work in Excel for Mac as expected. You may still need to adjust your scripts to reflect the notion of a file path on a Mac (e.g. /Users/USERNAME/Desktop/data.csv instead of C:\Users\USERNAME\Desktop\data.csv).

Remember, this is only the start (“Phase 1”): with Microsoft pushing to have “one Excel”, you can bet your bottom dollar further features will be coming soon. After all, Mac Excel is still a long way behind. Maybe they will Get the picture soon and Transform it shortly.

## New Views for Power Query

The Artist Formerly Known as Power Query – Get & Transform – is having a bit of a makeover, with some useful new features added to the View tab of the Power Query Editor, viz.



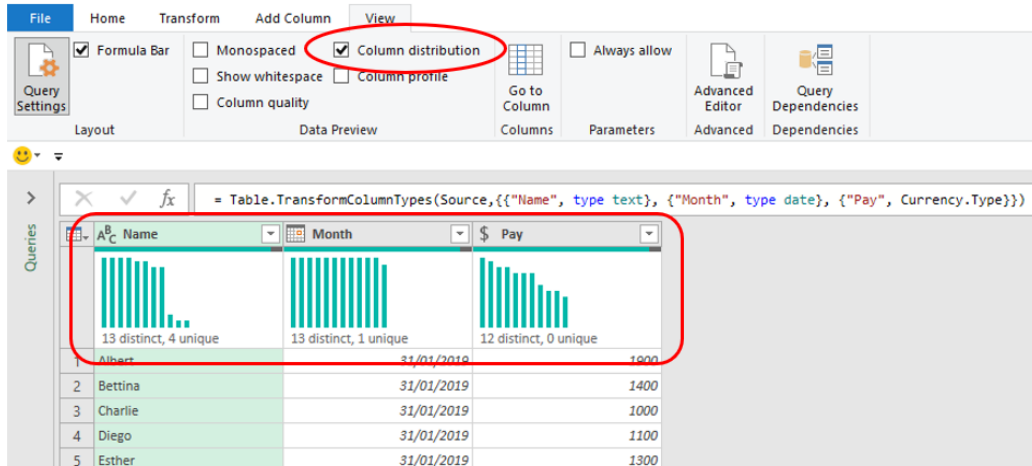
This has popped up “on the quiet” from Microsoft so we aren’t sure yet what versions of Excel this is in, or when it will be propagated to all versions (if at all). Situated on the View tab in the ‘Data Preview’ grouping, there’s three new features of note:

1. Column distribution
2. Column profile
3. Column quality.

Let’s take a look at each in turn.

## Column distribution

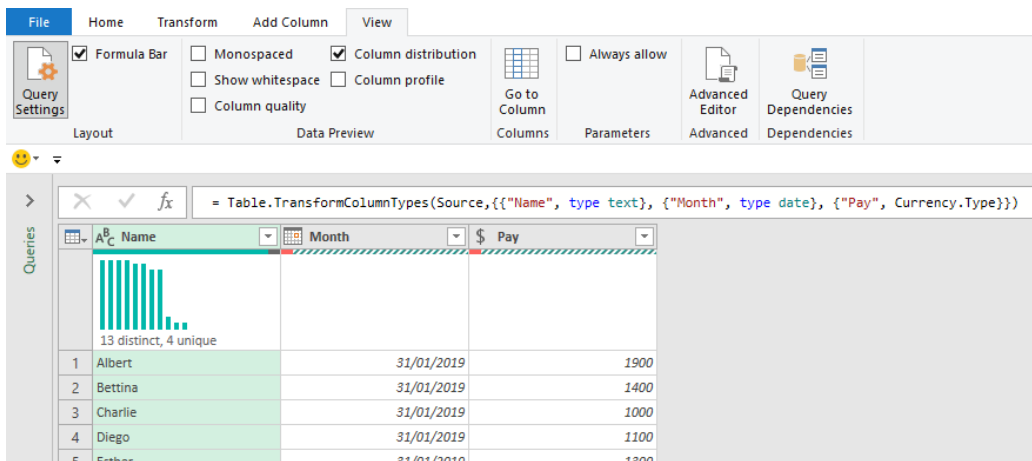
With the 'Column distribution' checkbox ticked, graphics are displayed at the top of each column (field), highlighting the number of distinct and unique items in that field:



To be clear:

- **distinct** means the number of different items in that field, so "A", "B", "B", "A" and "C" would be five items (five records), but only three distinct items – "A", "B" and "C". Further, remembering that **M** is case sensitive, the items "Alpha", "alpha", "ALPHA" and "Alpha " (with an extra space) would be four distinct items
- **unique** means the number of items that occur once and only once in that field. Using my example "A", "B", "B", "A" and "C" again would still be five items, but only "A" and "C" occur once and only once, so there are two unique items.

I am not sure if it's by default or by design, but the graphics and the counts of unique items will not show if there are any errors in that field. For illustration, this example has errors in both the Month and Pay fields, hence the blank 'Column distribution' sections:



Given this can be particularly unhelpful, it's probably a good idea to have a brief aside to discuss the different types of errors in Power Query...

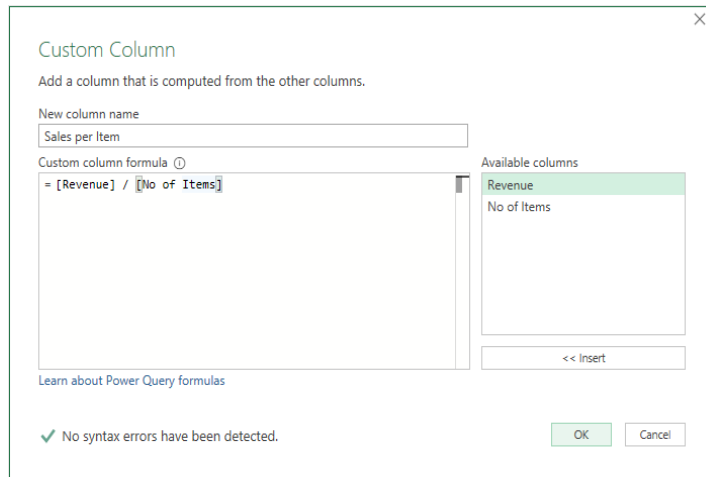
## ASIDE: Errors in the system

Consider the following Table is loaded from Excel into Power Query (Get & Transform) in the usual way:

	A	B
1	Revenue	No of Items
2	100	10
3	-100	10
4	100	0
5	-100	0
6	100	
7	0	
8		0
9		
10	100 ten	
11	hundred	10
12	hundred	ten
13	0	0



Once loaded, I can add a calculated column by clicking on the 'Custom Column' icon in the General grouping of the 'Add Column' tab, viz.



Essentially, I am calculating the 'Sales per Item' by taking the value in the Revenue field and dividing it by the corresponding number in the 'No of Items' field. It's not rocket science, but it does generate the following results:

	ABC 123 Revenue	ABC 123 No of Items	ABC 123 Sales per Item
1	100	10	10
2	-100	10	-10
3	100	0	Infinity
4	-100	0	-Infinity
5	100	null	null
6	0	null	null
7	null	0	null
8	null	null	null
9	100 ten		Error
10	hundred	10	Error
11	hundred	ten	Error
12	0	0	NaN

Only the first two records generate values we may use. The rest is a litany of errors and issues:

- **Infinity:** diving a positive number by zero gives rise to this error. Technically, such a calculation is mathematically indeterminate, but it is ascribed the error **Infinity**
- **-Infinity:** similar to the last error, this is the error returned when you divide a negative number by zero
- **NaN:** this does not mean your grandmother wrote the calculation, **NaN** means that it is not a number, *i.e.* it is either undefined or unrepresentable, especially in floating-point arithmetic, such as that prevalent in Power Query
- **null:** this means your formula attempts to perform a calculation with a *null* value. Good luck with that – it ain't gonna happen
- **Error:** this is a general catch-all when the error cannot be described specifically. In the example above, we are trying to use text in a formula – that's a type mismatch and Power Query will justifiably spit the proverbial dummy instead.

Bizarrely, only **Error** is counted as an error for reporting purposes, but it still will not present the chart currently:

	ABC 123 Revenue	ABC 123 No of Items	ABC 123 Sales per Item
1	100	50	2
2	-100	50	-2
3	100	0	Infinity
4	-100	0	-Infinity
5	100	null	null
6	0	null	null
7	null	0	null
8	null	null	null
9	0	0	NaN

There is worse – *sometimes*. If you filter out some of the errors, it is possible to break our new ‘Column distribution’ reporting tool on occasion:

	ABC 123 Revenue	ABC 123 No of Items	ABC 123 Sales per Item
	An unexpected error occurred		
1	100	50	2
2	-100	50	-2

Some of the time, trying a second time, or clicking on another ‘Applied Steps’ item and then returning, fixes this – but not always. Sometimes, you’ll just have to close Excel and re-open it, perhaps two or three hundred times. I am sure this will be fixed by the Power that Be *very soon* – we have already reported this error!

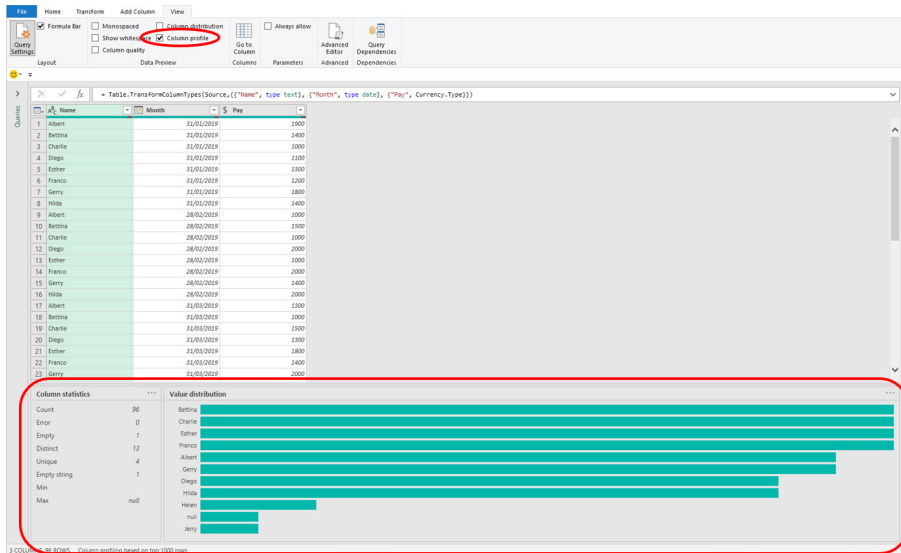
Filtering out all of the error types will provide the charts once more:

	\$ Revenue	123 No of Items	\$ Sales per Item
	2 distinct, 2 unique	1 distinct, 0 unique	2 distinct, 2 unique
1	100	50	2
2	-100	50	-2

If you don’t get these charts, do ensure that your data type isn’t set to ‘Any’ – that seems to upset the appletcart.

### Column profile

The ‘Column profile’ view adds statistics and a chart at the foot of the query in the Power Query Editor, viz.



The statistics are useful, and now that I have explained what is meant by distinct, unique and errors, fairly self-explanatory. However, it should be noted that certain fields may have both Empty items and Empty strings:

- **Empty:** this means the cell either contains nothing or Power Query is treating the value as *null*
- **Empty string:** this is the equivalent of the Excel formula =*""*, i.e. there is something there – it just happens to be, er, nothing.

The Statistics window varies depending upon the data type of the column selected (e.g. text, number) and the field’s contents. For example, in a numeric field, you may see something like the following instead:

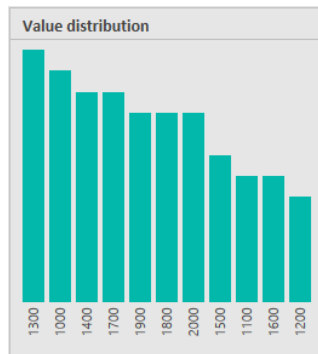
Column statistics	
Count	96
Error	0
Empty	2
Distinct	12
Unique	0
NaN	0
Zero	0
Min	1000
Max	2000
Average	1506.38...

Here, unlike a text field, Min and Max mean something, but there is no longer an 'Empty string' item. You should note also the scroll bar, highlighting 'Standard deviation' is off screen. Unless you change the screen resolution, it does not appear you can presently change the size of this Statistics dialog.

Regarding the 'Value distribution' dialog, you can get bar charts...



...and column charts:



The logic eludes me presently, but it is worrying that for bar charts you may not see all records displayed (only 11 of the 13 are shown in the example above). With no options to revise images currently, do bear this is mind when viewing such graphics.

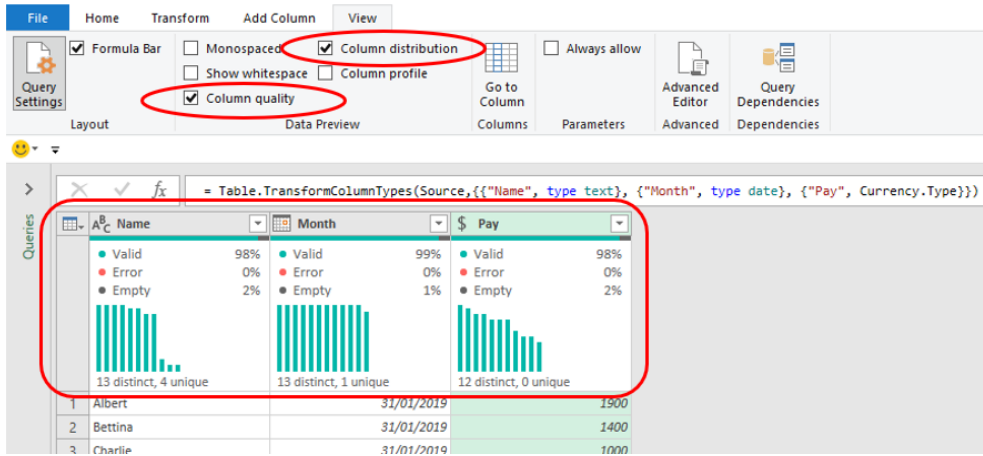
### Column quality

This again reports above the field data:

The screenshot shows the Power BI ribbon with 'Column quality' checked. Below, a data table displays quality metrics for three columns: Name, Month, and Pay.

Column	Valid	Error	Empty
Name	98%	0%	2%
Month	99%	0%	1%
Pay	98%	0%	2%

If the 'Column distribution' is checked as well, the graphic will appear below the statistics, viz.



The Data is separated into three elements:

1. **Error**: as explained above, this is any record that results in **Error**
2. **Empty**: this is a blank or *null* record
3. **Valid**: this is everything else (both figuratively and literally!). All specified error types (*see above*), *i.e.* those not classified as **Error**, will be included in **Valid**.

It is quite a basic analysis, but it is better than what we had previously (*i.e.* nothing).

### Word to the Wise

There is a classic "gotcha" in here too... The data profiling is only performed on the first 1,000 records by default. If you have more than this ensure you select the entire data set. To do this, you have to LEFT-click at the foot of the Power Query Editor and select 'Column profiling based on entire data set' instead (*pictured*):



Finally, as I have explained these new views, I may have come across as a little negative. This is not my intention. The features are brand new, and you should be aware of the issues we discovered. You may find plenty

of your own too. This is often the case with new features and functions. Microsoft *will* fix them as they are reported, so don't let this put you off trying out these new views – assuming you can access them.



## Visual Basics

We thought we'd run an elementary series going through the rudiments of Visual Basic for Applications (VBA) as a springboard for newer users. This month, we consider Events.

While most VBA scripts tend to be triggered by buttons or commands that a user can execute, it can also be programmed to run when a particular action is triggered elsewhere in the Excel workbook. We referred to this earlier in the series and will now address the process of creating these events.

These triggers can be found in the VBA Editor and navigating to a Sheet or to **ThisWorkbook**.

### Worksheet events

There are a couple of key worksheet events generally used:

- **Activate**
- **SelectionChange**.

The Activate event will trigger whenever the worksheet in question is activated (e.g. if a user clicks on the worksheet tab in Excel).

```
Private Sub Worksheet_Activate()  
  
    'Run some code here  
    Application.Calculate
```

```
End Sub
```

This sort of event can be used to trigger an update or clean-up of data on a sheet. It is commonly used to refresh PivotTables when a sheet is selected, with the code cycling through available PivotTables and refreshing each one.

The **SelectionChange** triggers whenever a new cell is selected within the

worksheet, e.g. when a cell is clicked on or when the selection range is expanded or reduced. This event is a bit different, in that there is also an additional parameter that becomes available, initially referred to as "ByVal Target as Range". This allows you to pass the selected area through to the VBA script being run.

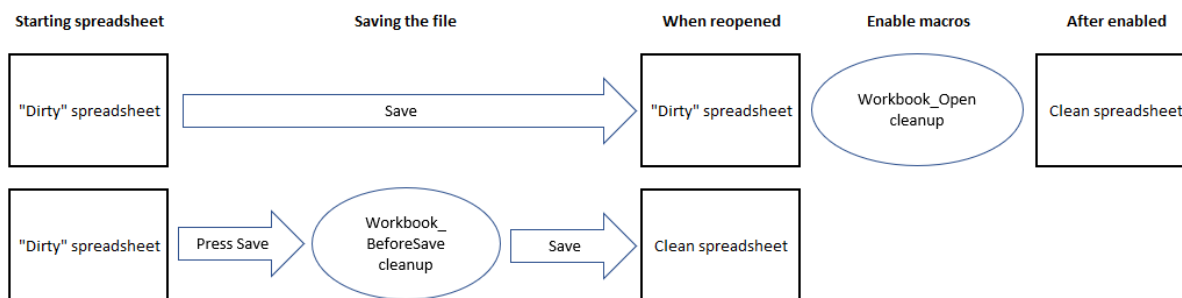
### Workbook events

There are several key workbook events that are commonly used in practice:

- **Activate**
- **BeforeClose**
- **BeforeSave**
- **Open**.

In particular, the **Open** event is commonly used to set up disclaimer messages, user guides and instructions, or for model clean-up. However, this relies upon the user enabling macros prior to the macro attempts to

run. For a more robust process, any clean-up or disclaimer script should be written prior to the file being saved, so that any user will be required to respond to the disclaimer before the file can be used.



This is where the **BeforeSave** event is particularly useful. The event is triggered upon a **Save** or **SaveAs** command, and allows users to run any clean-up prior to the file being saved. The macro can be set to restore

the file after it has been saved, so that the user won't need to go through the disclaimer process, resulting in a seamless save action.

### Disabling events

While events can be useful, at times you may need to save the file as part of another macro script, assuming that the event will not be triggered. In these instances, you need to include the command:

```
Application.EnableEvents = False
```

This will prevent any events from being triggered until you restore the actions using:

```
Application.EnableEvents = True
```

If you decide to switch off events at the start of your macro to improve the speed and efficiency of your macro run, remember to switch it back on subsequently after your macro finishes running.

More next month.

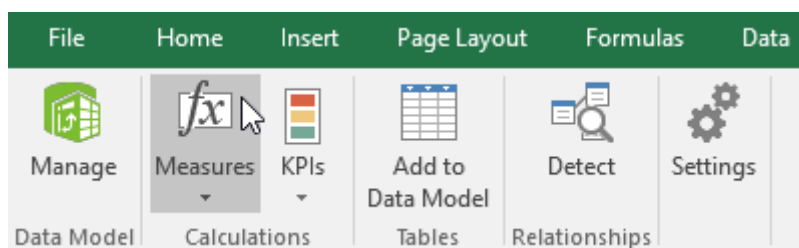
## Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we look at measures.

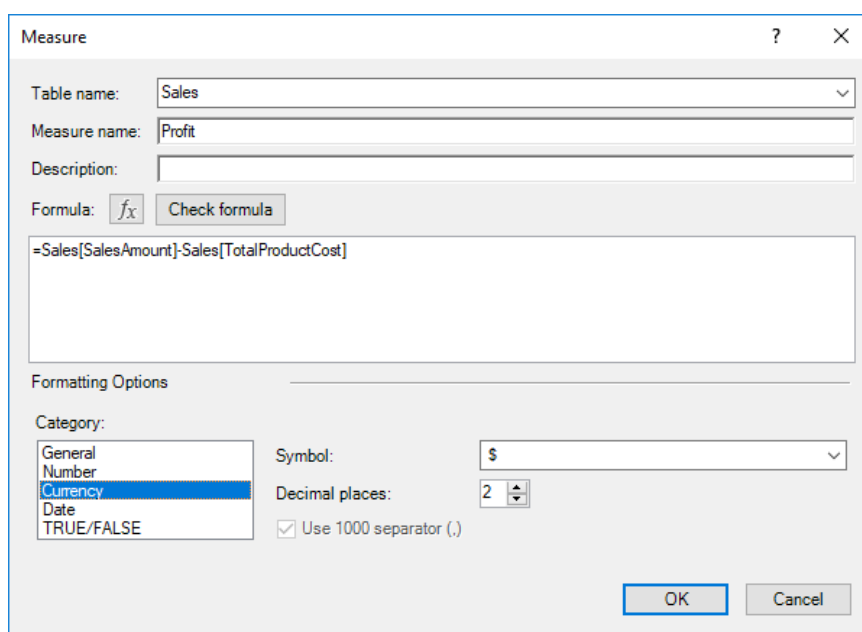
A **measure** is a calculation that can be created to be used in PivotTable analysis. When measures are created, they sit in the values area of the PivotTable ready to be used in a PivotTable. Unlike creating formulae in Excel, a measure in Power Pivot is saved and can be used time and time again.

### Example

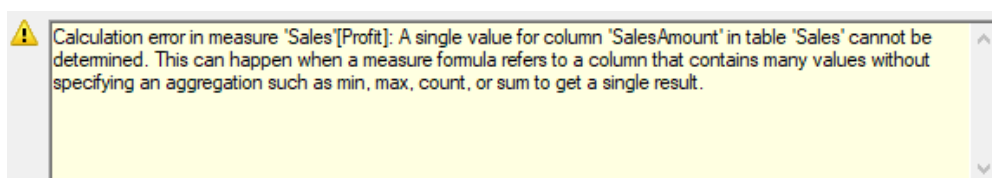
Let's create a simple measure. Select the 'Power Pivot' tab on the Ribbon and then choose the 'Measures' option.



Enter the following inputs into the resulting dialog box:

The 'Measure' dialog box is shown with the following fields: 'Table name' set to 'Sales', 'Measure name' set to 'Profit', and 'Description' empty. The 'Formula' field contains '=Sales[SalesAmount]-Sales[TotalProductCost]'. Under 'Formatting Options', the 'Category' is 'Currency', 'Symbol' is '\$', 'Decimal places' is '2', and 'Use 1000 separator' is checked. 'OK' and 'Cancel' buttons are at the bottom right.

If we click on 'Check formula' Power Pivot will return with this warning message:



This is because we need to include an aggregation function such as **SUM**, **MAX**, **MIN**, **AVERAGE** or **COUNT**. A quick fix would be to include the 'SUM' aggregation in front of the fields, which will make Power Pivot happy:

Measure

Table name: Sales

Measure name: Profit

Description:

Formula:

=Sum(Sales[SalesAmount])-Sum(Sales[TotalProductCost])

✓ No errors in formula.

Formatting Options

Category:

- General
- Number
- Currency
- Date
- TRUE/FALSE

Symbol: \$

Decimal places: 2

Use 1000 separator (,)

Profit will then appear as the sole field in the 'Values' section of the PivotTable.

	A	B
1	Profit	
2	\$1,154,313.35	
3		

Drag fields between areas below:

Filters

Columns

Rows

Σ Values

Profit

Do you see how the value is simply 'Profit' and not 'Sum of Profit' which is an annoying irritation in PivotTables? This is another benefit of creating a measure. We can now bring in other fields to complete our PivotTable – but more on that next time!

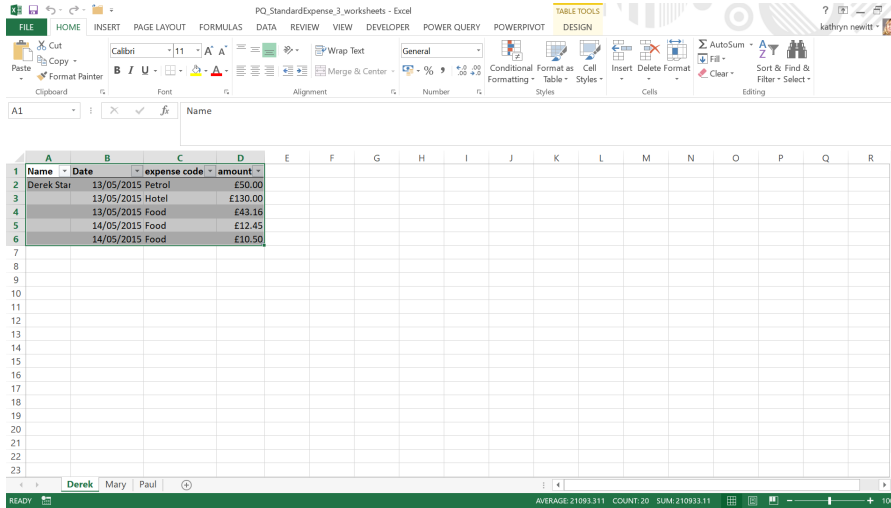
More *Power Pivot Principles* next month.

# Power Query Pointers

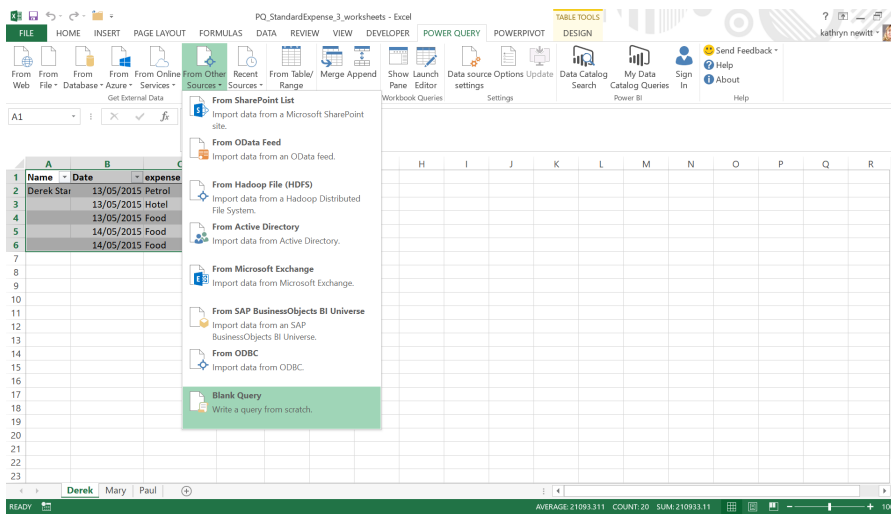
Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from [www.sumproduct.com/blog](http://www.sumproduct.com/blog). If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we take a look at when the 'Structured Column' section on the Transform tab is used.

Previously, we looked at how to aggregate three worksheets into one query. For this month, we'll run through the process again briefly, as the main aim in this newsletter is to explore the 'Structured Column' section on the Transform tab, which is not available for most column types.

The workbook we will use, containing the three worksheets, is shown below:



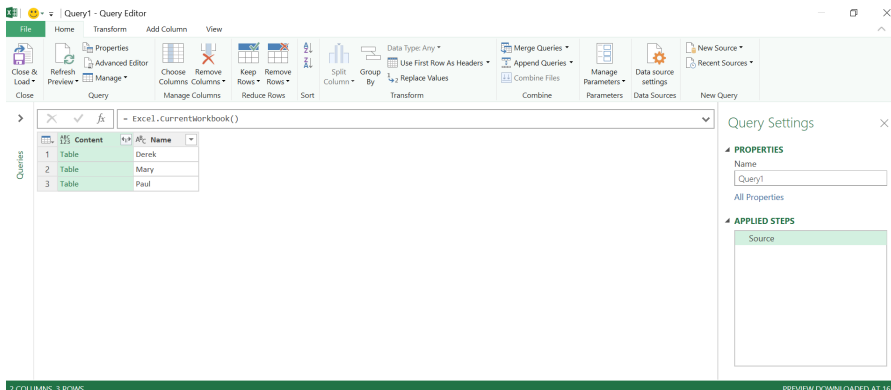
First, we extract the data from these worksheets by creating a new 'Blank Query' from the 'POWER QUERY' tab in the 'From Other Sources' dropdown:



In the Power Query Editor, in the formula bar at the top of the screen, enter the following formula:

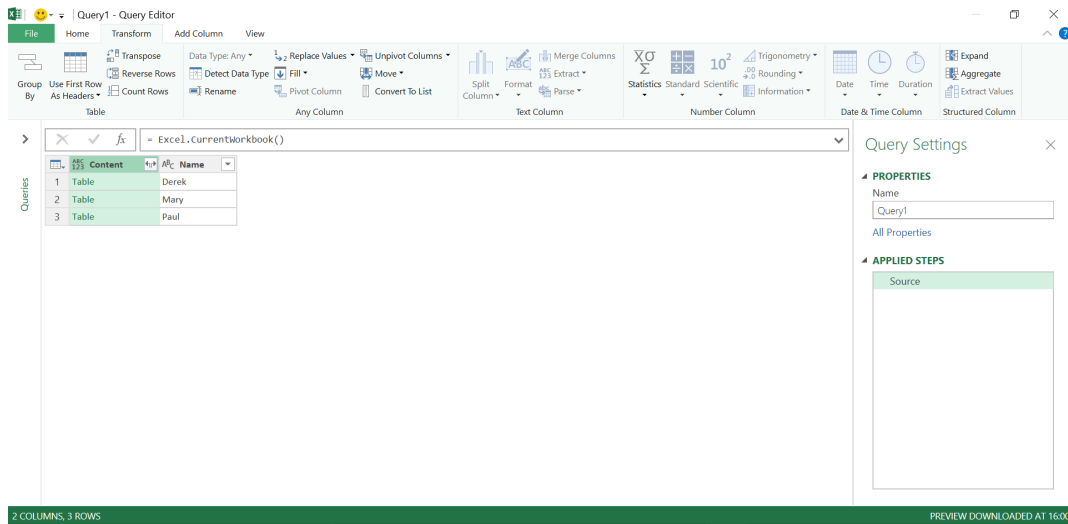
**=Excel.CurrentWorkbook()**

This lists our tables (and would list named ranges and workbook connections) that we have in our workbook.

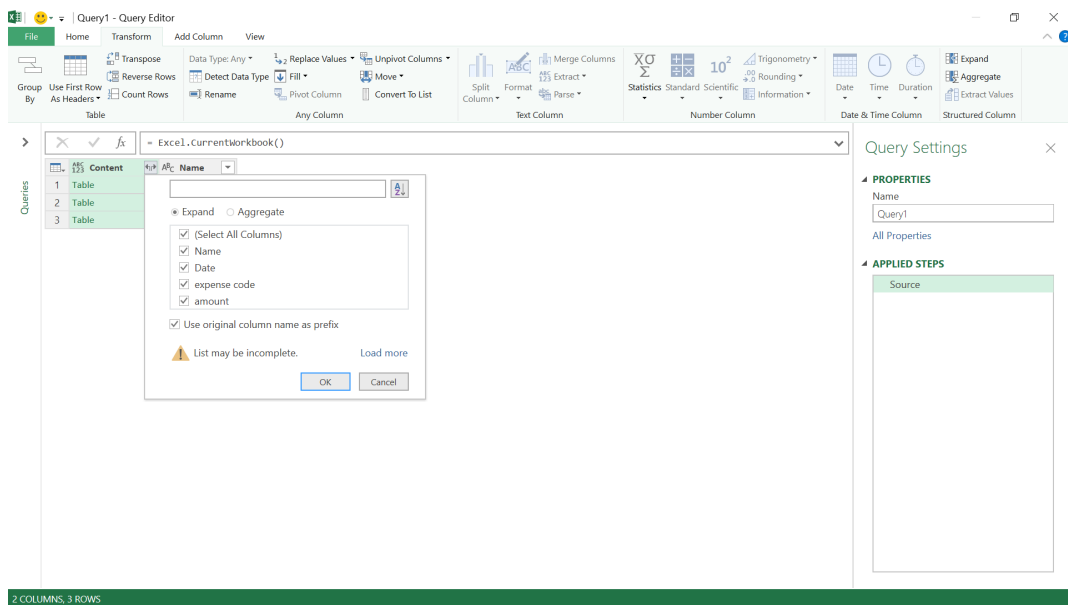




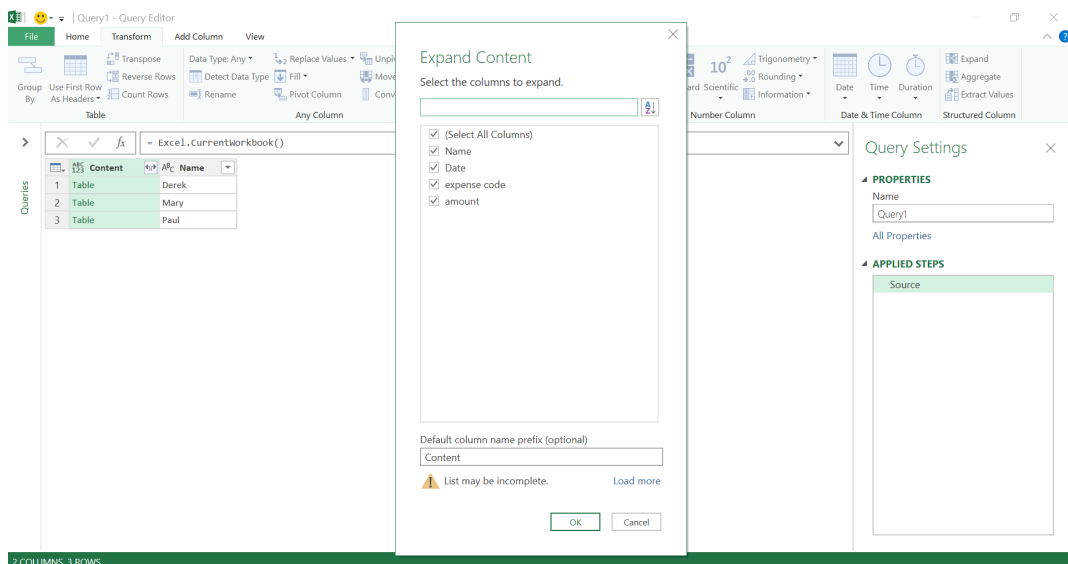
It is at this point that we deviate from our prior aggregating worksheets example to investigate the section on the Transform tab which is normally greyed out: the 'Structured Column' section.



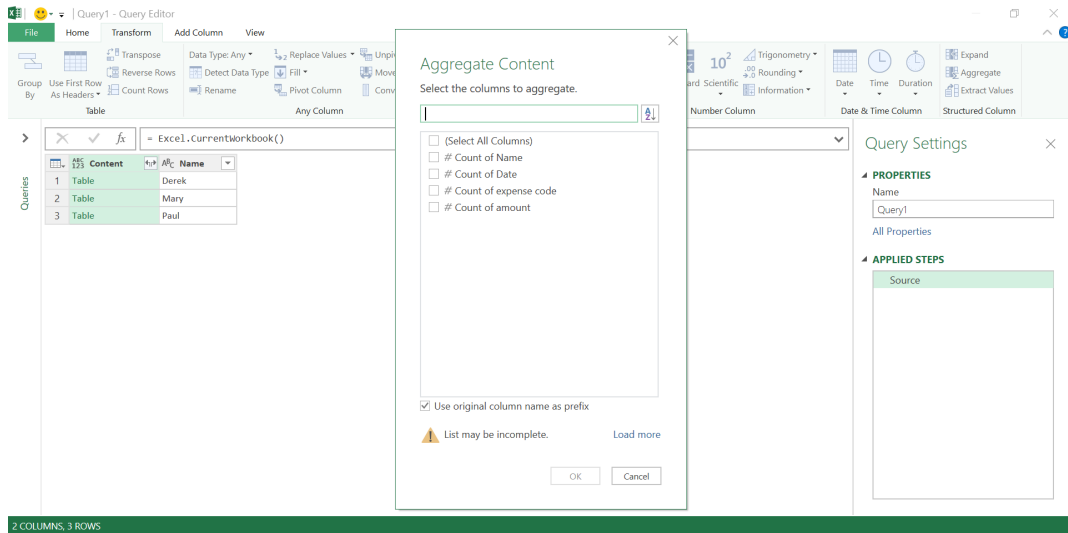
'Expand' looks very similar to the icon next to the **Content** column. To see if this is true, try using each option, firstly, the icon next to **Content**:



We can choose to 'Expand' or 'Aggregate' our data, and decide which columns we wish to bring through. Try using the 'Expand' option from the 'Structured Column' section.

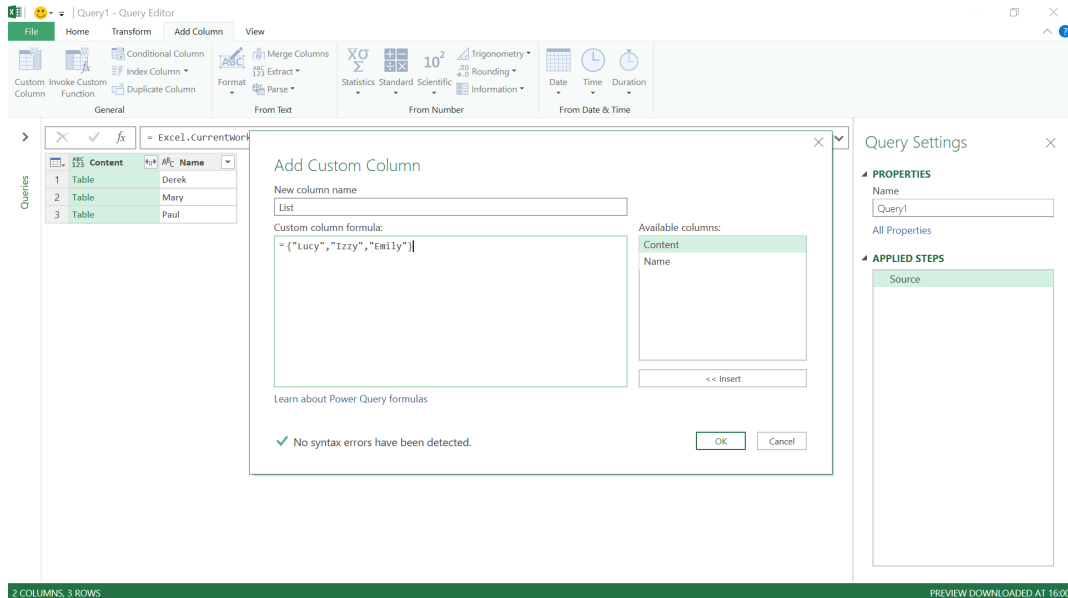


A different size and shape perhaps, but no aggregate option here. We would need to pick the 'Aggregate' option from the 'Structured Column' section.



Therefore, this is a quicker way to get to the aggregate options.

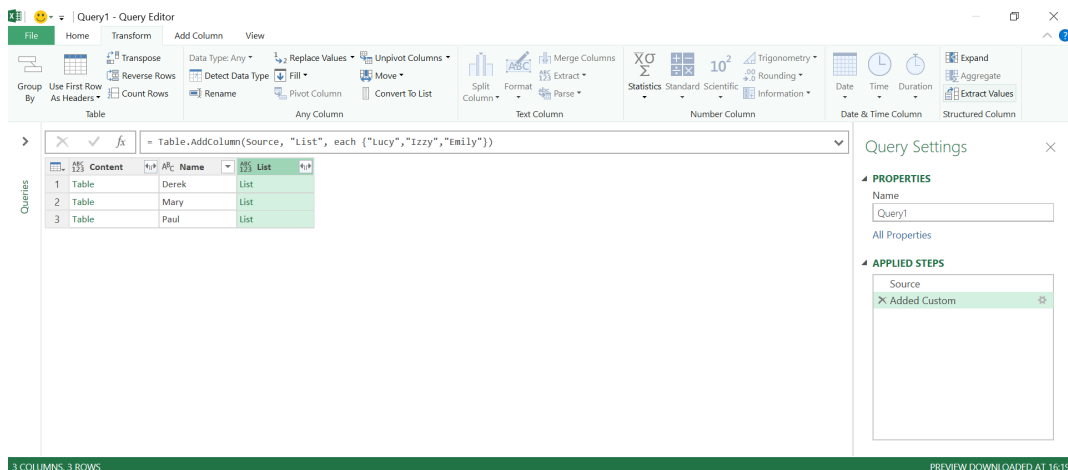
'Extract values' is still greyed out as an option on the 'Structured Column' grouping, so to find out what this does we will create a column that contains a list.



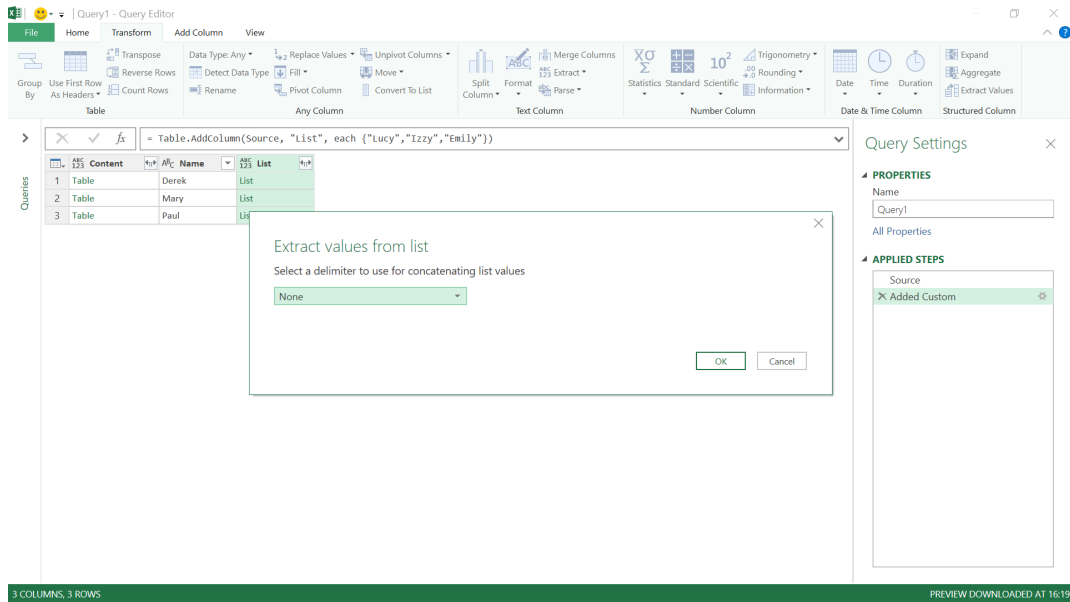
The M code for a list is simple:

```
={"Lucy","Izzy","Emily"}
```

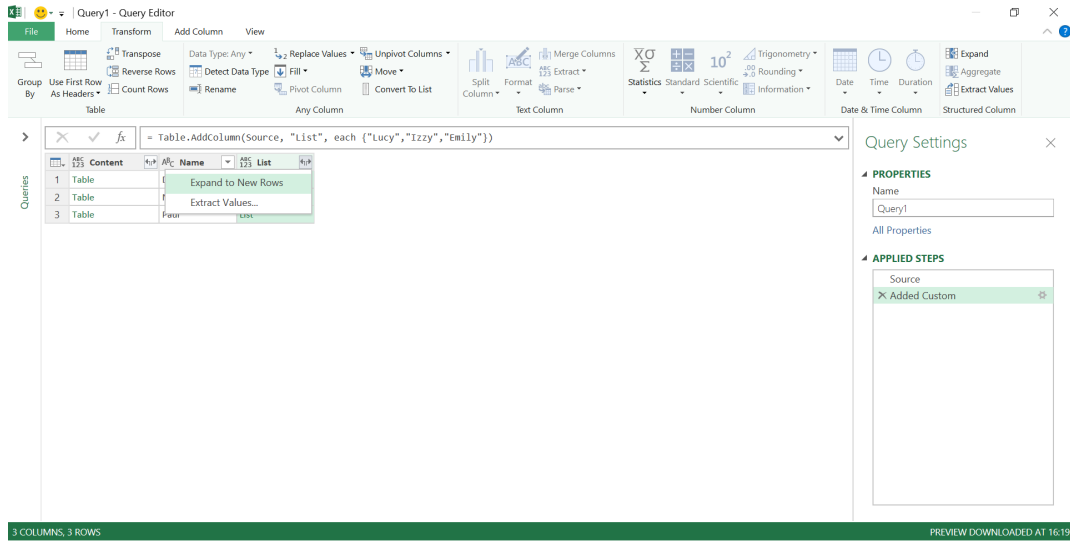
which creates a short list of three names. We click OK to create the column:



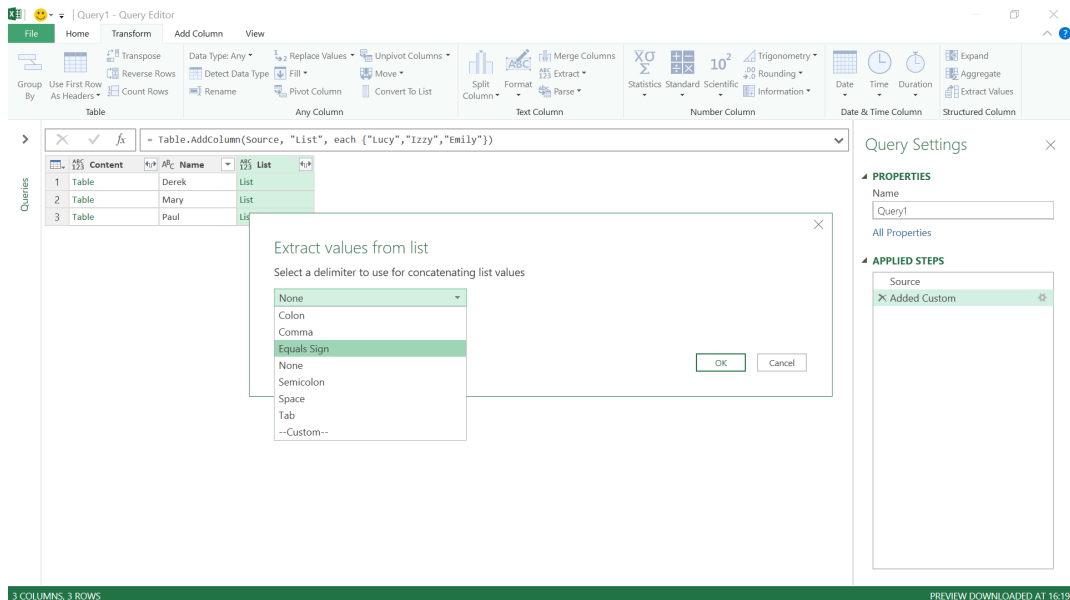
Not only has our **List** column appeared, but 'Extract Values' is no longer greyed out. Therefore, let's try it.



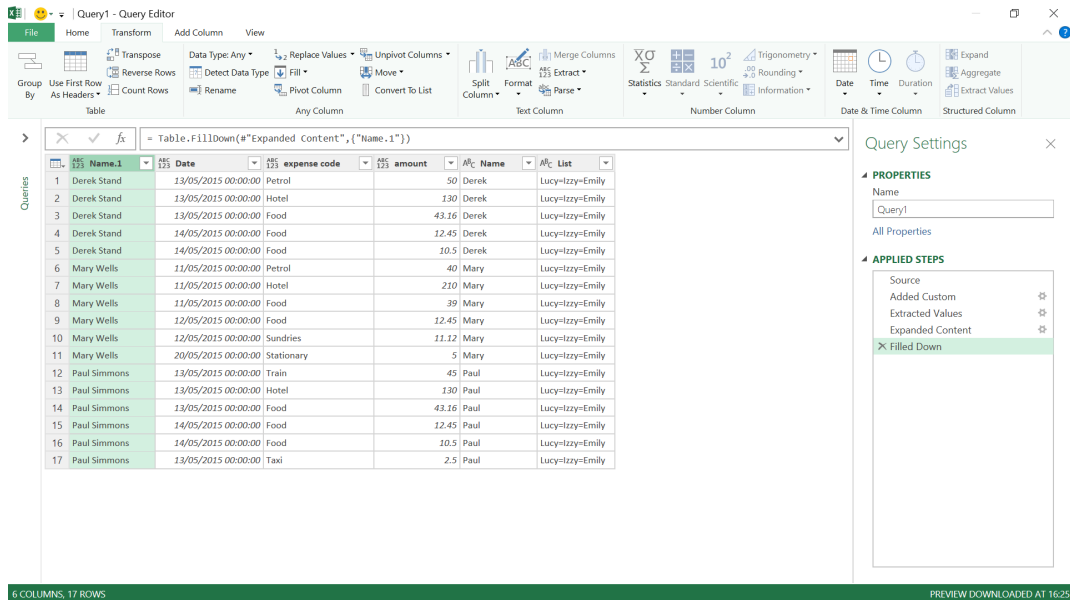
The icon next to List can also be used, and this gives access to both the 'Extract Values' and Expand options:



If we continue to extract our values, we may pick a delimiter to separate the list items:



Clicking OK means that our list has been extracted into a table. Finally, we can expand all the data and tidy it up.



The functions in the 'Structured Column' section can be also be accessed from the icons, but sometimes seeing the words can make it easier to see what is needed and jump right to what we need to do.

More next month!

## Power BI Desktop Update for October

The latest update provides improvements in several areas. In the data preparation area, Microsoft has introduced query diagnostics, which allow you to see which queries are being run behind the scenes. On the reporting side, real-time reporting has been improved, with a new automatic page

refresh feature. And then there's Q&A – which has also been revamped, with a new Q&A visual, tooling and an improved user experience.

Here's the complete list of October updates:

### Reporting

- Automatic page refresh for DirectQuery (Preview)

### Analytics

- The new Q&A visual
- Improved user experience for Q&A:
  - Improved drop-down control
  - Red / Blue underlines
  - Improved visual results
- Natural Language improvements for Q&A:
  - Integration with Office / Bing thesaurus
  - Support for measure tables and better handling of table names and ambiguity
  - Row label and synonyms support inside the Modeling pane
- Q&A Tooling (Preview):
  - Review questions
  - Teach Q&A
  - Review all changes made
- Support for SSAS and Azure AS including RLS

### Visualizations

- PowerApps visual now included by default
- New xViz visuals:
  - Hierarchical Variance Table
  - Linear Gauge
  - Tag Cloud

### Data connectivity

- Sagra Emigo connector Generally Available
- Azure Cost Management connector updated
- New Workplace Analytics connector

### Data preparation

- Query Diagnostics
- Data profiling enhancements

### Template Apps

- Project Web App

### Other

- New file format: .pbids
- Performance improvements for modelling operations.

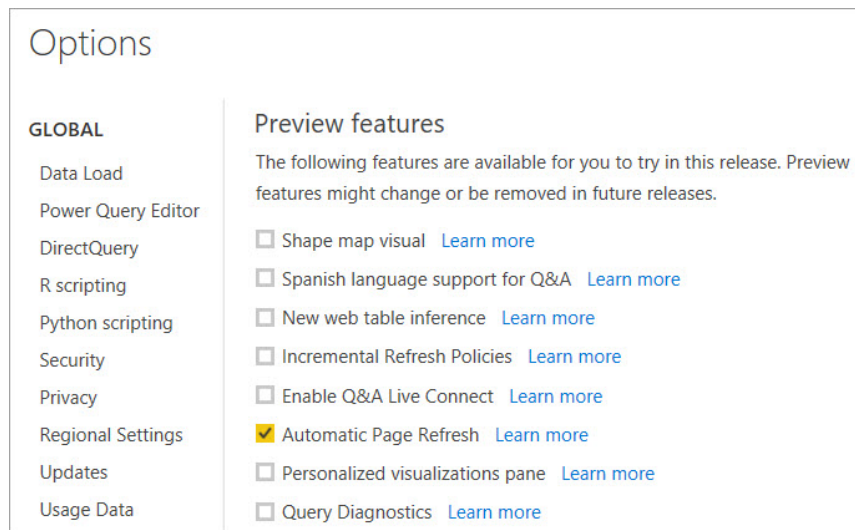
Let's take a look at each in turn.

## Automatic page refresh for DirectQuery (Preview)

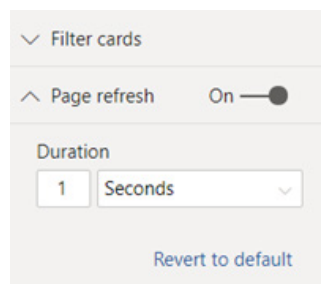
When monitoring critical events with Power BI, it's important that the data is reflected in Power BI as soon as the data source gets updated. For example, in the manufacturing industry, it's critical to know when a machine is malfunctioning. To enable this, Microsoft is releasing automatic page refresh, which allows you to set the refresh interval of visuals in your report when using a DirectQuery source. For very

fast updating sources, this near real-time experience will allow your consumers to monitor time-critical reports and ensure that they are always viewing the latest.

To use automatic page refresh, make sure you have it turned on in the Options dialog:



After that, you will see a 'Page refresh' card in the Formatting pane for the page when connected to a DirectQuery source. In that card, you'll be able to set the duration of your refresh, viz.



In Power BI Desktop, you can set up this refresh as frequently as once a second, assuming your DirectQuery source can handle that. In the Power BI Service, the minimum refresh time is 30 minutes for a regular workspace, so once you publish the report, the duration will be

overwritten if it's less than 30 minutes. If your workspace is part of a Premium capacity, your duration can go as low as the minimum set by the capacity administrator, and will only work if the said administrator has enabled automatic page refresh in the capacity.

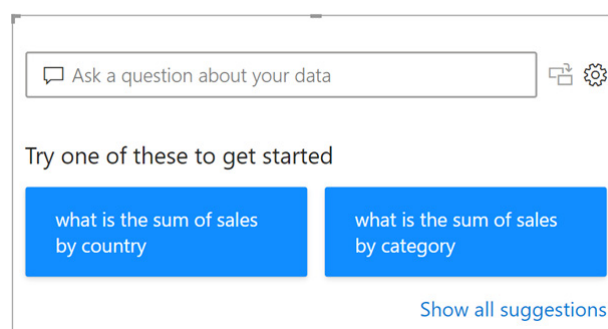
## The new Q&A visual

NLG (natural language generation) is improved in this update, with a new Q&A visual. This is a visual experience that allows you to give the Q&A experience to your end-users. While similar to using a button to trigger Q&A, this visual can improve discoverability for your report consumers,

as the experience isn't hidden behind an extra button click. The new Q&A Visual can interact, cross filter, support bookmarks and be themed just like any other visual. To add the visual to the report, just click on the Q&A icon from the Visualization pane:

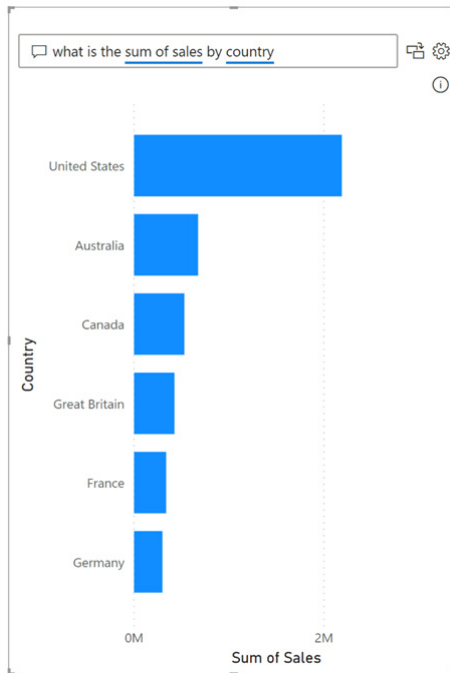


This will add the Q&A visual to the report page with an input box for you to type your questions and a few suggestions to get started with. Clicking any of the suggestions will automatically populate the input box with that question and show you the result.

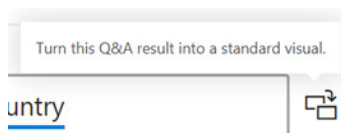




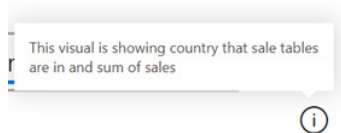
Otherwise, if you have your own question to ask, type it out and see Q&A automatically give you the result.



As a report author, you also have the ability to use this visual as an authoring tool and convert the Q&A visual into the specific visual you're seeing if you'd like to stick with that one visual. Just click the icon next to the input box.



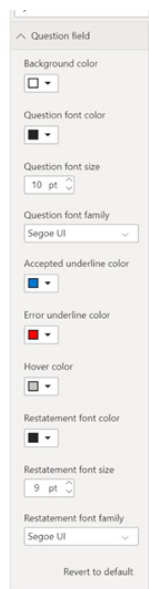
Near the top of the visual result, you'll also see an info icon that has the restatement as a Tooltip. If you ever want to confirm what the Q&A result is showing you, you can hover over this icon to see the restatement.



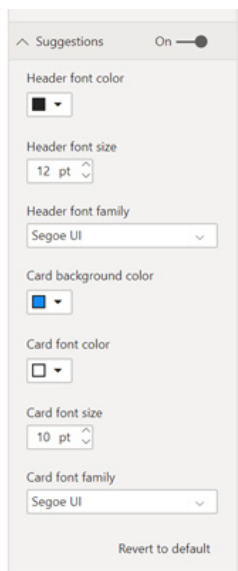
The state the Q&A visual is in when you save your report is what your report consumers will see as well. Therefore, if you have the report with a Q&A in the box, that will be the default view for your end users. You can also use bookmarks to save different Q&A questions for your end users. The Q&A visual will also automatically pick up your applied theme

and cross-highlight automatically works between visuals on the report page and the Q&A visual.

You also have complete formatting control over the question input box through the 'Question field' card in the Formatting pane.



The Suggestions appearance can also be formatted:

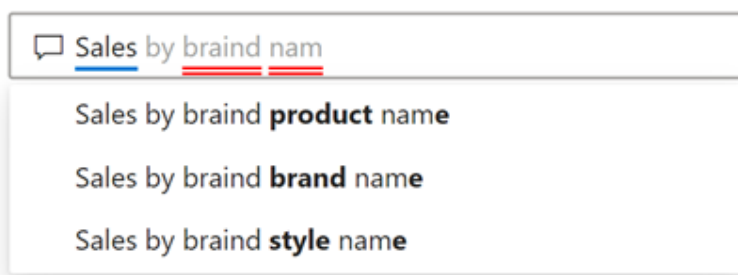


If you're using the Ask a Question button in the Ribbon, or double-click to trigger the Q&A experience, those entry points will now create Q&A visuals.

### Improved user experience for Q&A

This update has also advanced the user experience for the Q&A question input box. You'll now see [blue underlines](#) on terms that the software understands from your data model. [Red underlines](#) are used for terms

that are ambiguous or aren't understood at all. There is also a new dropdown experience when selecting a specific auto-suggestion as you type that is similar to what is common with web search engines:



A lot of work has clearly been undertaken to improve the visuals that are generated from Q&A. This feature should, as a result, be even better at

suggesting the best visual (following Microsoft's researched visualisation best practices) for the specific questions you ask.

### Natural Language improvements for Q&A

#### INTEGRATION WITH OFFICE / BING THESAURUS

The core engine inside Q&A has been modified, allowing it to automatically understand common synonyms by leveraging Office / Bing data to understand terms which are not mentioned in your data model.

#### SUPPORT FOR MEASURE TABLES AND BETTER HANDLING OF TABLE NAMES AND AMBIGUITY

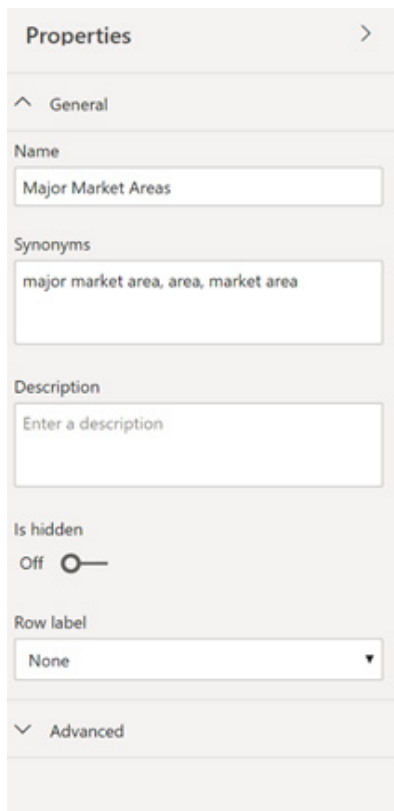
If you have ever used Q&A, you may well be aware that Q&A does not work very well when dealing with disconnected tables (*i.e.* tables that are not connected by a one-to-many specified relationship). This update brings the biggest improvement to Q&A working out of the box automatically

by providing automatic handling of disconnected / measure tables and figuring out which measures are related to which dimensions. You no longer need to make a specific data model just for Q&A.

#### ROW LABEL AND SYNONYMS SUPPORT INSIDE THE MODELING PANE

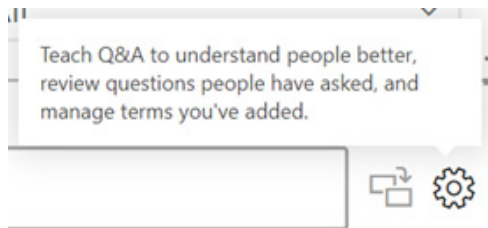
This update also brings support for row labels (formerly known as default labels which were available inside Excel), which allows you to define the row identifier for a table. To define the row label, you can go to the Modeling tab and click a table name to see the option. Also, if you

wish to define synonyms which are to be used by Q&A, you may use the Modeling pane to quickly define all the synonyms for all columns inside your data mode.



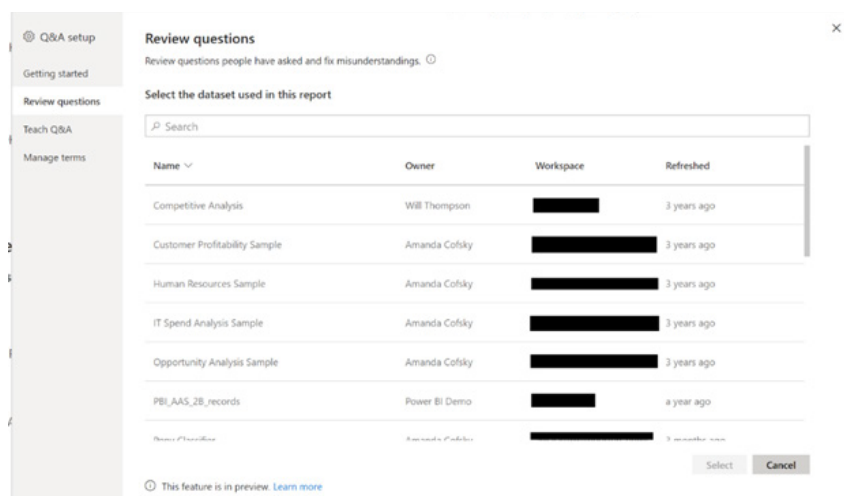
### Q&A Tooling (Preview)

This update sees the introduction of Tooling, which you may use to see how people are using the Q&A visual and improve the results they're getting. Tooling may be accessed by click on the gear icon.



### REVIEW QUESTIONS

The first section of the Tooling allows you to see the questions consumers are asking in your reports. You start out by picking which dataset is used in this report.

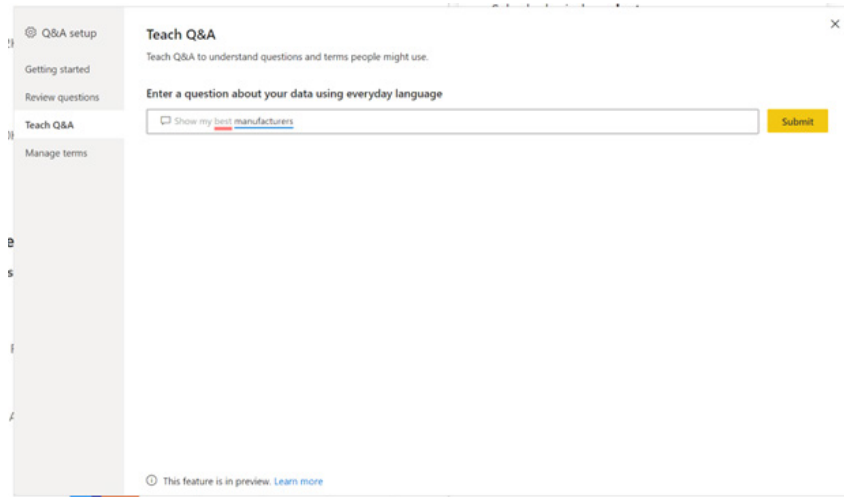


From there, you'll see a list of questions that have been asked in the Power BI Service from that data. You can filter the list to see either all questions or just the ones where the consumer has asked for you to

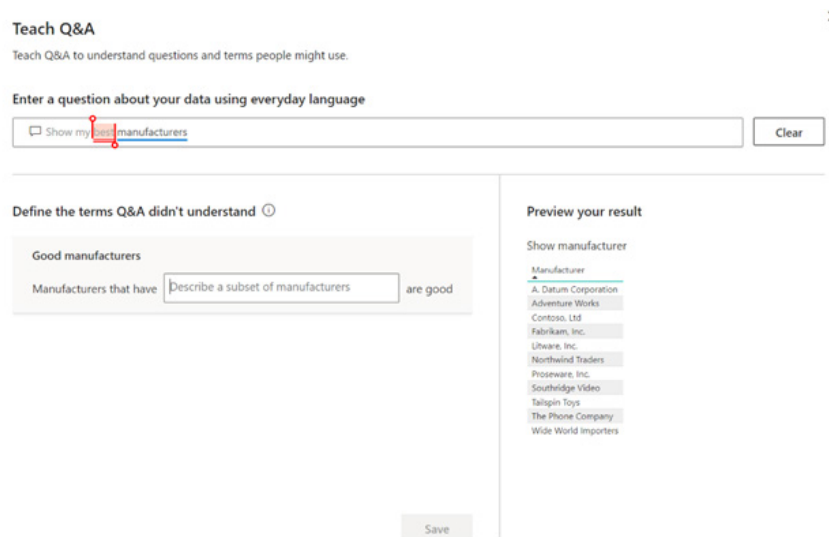
fix the result to give a better answer. Any questions where Q&A didn't understand the terms will have a pencil next to them, which will take you to the next section, **Teach Q&A**, to fix it.

## TEACH Q&A

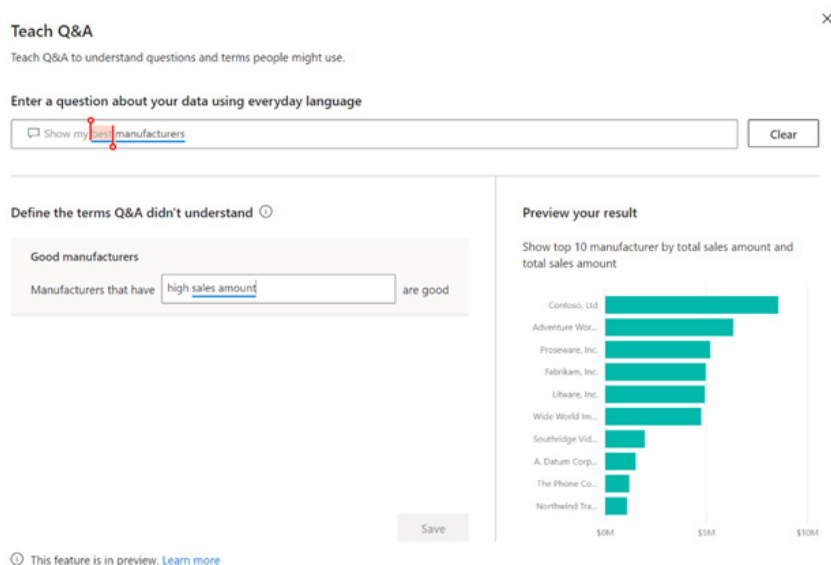
The second section in the Tooling allows you to teach Q&A how to respond to questions that are ambiguous. You can start by typing in a question such as “Show my best manufacturers” and clicking Submit.



The terms that Q&A didn't understand, in this case “best”, are identified, and you are provided with an opportunity to teach Q&A what those terms mean.



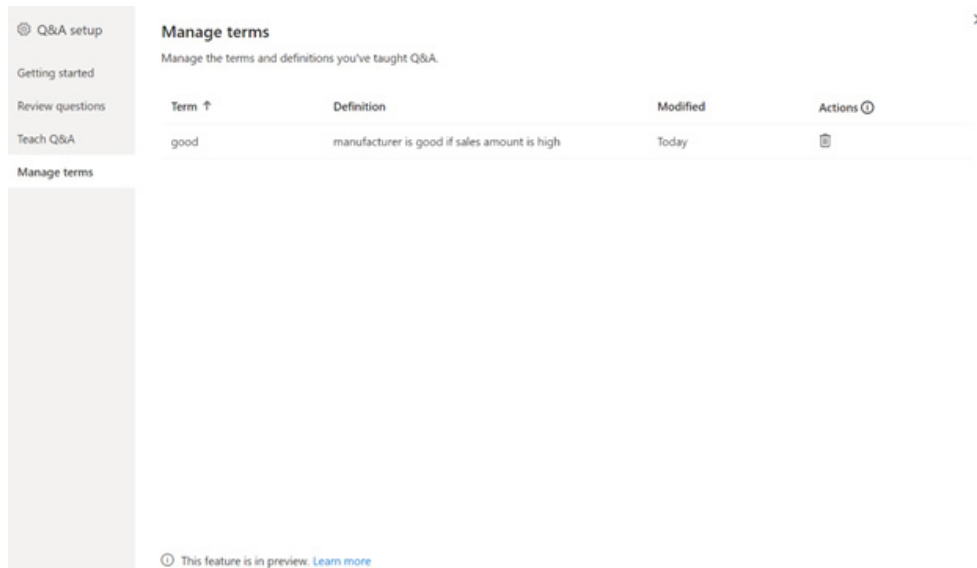
You can then type in your own custom definition, such as good manufacturers have a high sales amount, see the potential result on the right, and save the definition once you are happy.



You can do this for as many terms and questions as needed.

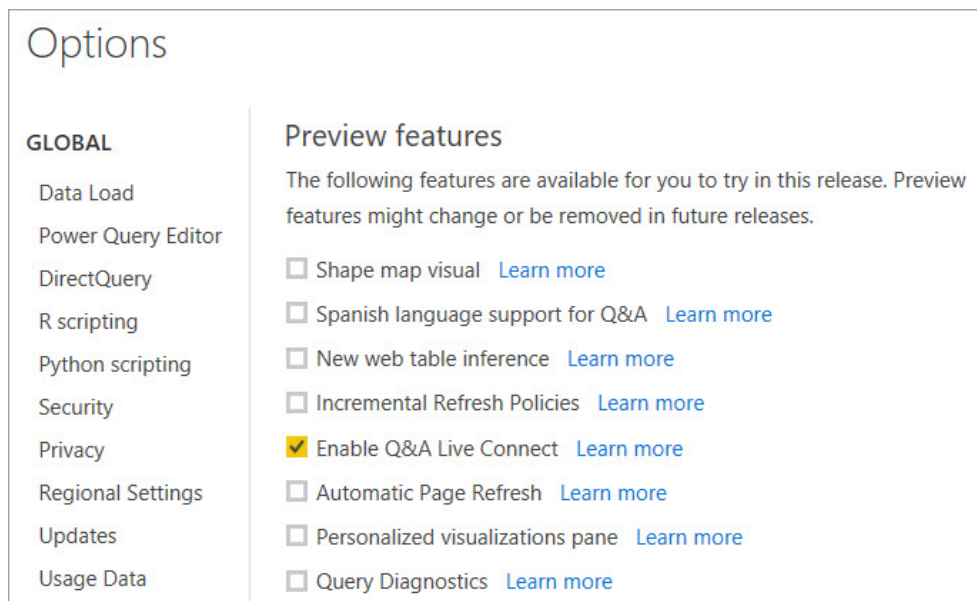
## REVIEW ALL CHANGES MADE

The third and final section allows you to manage the terms you've added to Q&A through the **Teach Q&A** section. You'll see each term, its definition, when it was created, and if you no longer want the term, an option to delete it.



### Support for SSAS and Azure AS including RLS

Power BI Desktop supports SQL Server Analysis Services and Azure Analysis services, with Q&A allowing you to connect to your data without having to import the data first inside Power BI. In order to use this feature, you must first enable the 'Preview feature' switch in **Options and settings -> Options -> Preview features**.



Presently, there are a few limitations that will be addressed shortly, but Row Level Security (RLS) is available.

### PowerApps visual now included by default

Last month's update saw the PowerApps visual become Generally Available and certified. To build upon this, this update sees visual now included by default. Instead of needing to add the visual from the marketplace, you'll find it automatically included with the list of other built-in visuals within

the Visualization pane. This visual is a key integration with the rest of the Power Platform and enables key scenarios such as writeback and taking immediate data-driven actions on your insights.

### New xViz visuals

There are three new visuals as part of the xViz suite. The xViz suite is a set of custom visuals that have been added to Power BI Desktop. You can use all the visuals within the suite for free with their branding and some data row restrictions or else you can pay one price to unlock all the visuals.



## HIERARCHICAL VARIANCE TABLE

The Hierarchical Variance table lets you compare measures within a hierarchical table layout and see the variance, % variance, and how these contribute to the whole. If you include a date field, you can also see a sparkline within the table.

Category	Trend	Forecast	Budget	Variance	% Variance	% Contribution
- Total		93,493K	82,887K	10,607K	13.0%	*
- Revenues		223,493K	211,099K	12,394K	6.0%	>99.0%
- Urban		221,368K	209,077K	12,290K	6.0%	>99.0%
+ Moderation		98,358K	92,701K	5,657K	6.0%	46.0%
+ Convenience		93,841K	88,745K	5,097K	6.0%	41.5%
+ Extreme		28,910K	27,386K	1,524K	6.0%	12.4%
+ Regular		258K	245K	13K	5.0%	0.1%
- Rural		2,126K	2,022K	103K	5.0%	0.8%
+ Productivity		1,840K	1,751K	89K	5.0%	85.7%
+ Select		286K	271K	15K	5.0%	14.3%
- Expenses		(130,000K)	(128,213K)	(1,787K)	(1.0%)	(16.9%)
- Rural		(2,319K)	(2,432K)	114K	5.0%	6.4%
+ Productivity		(911K)	(849K)	(62K)	(7.0%)	(54.3%)
+ Select		(1,408K)	(1,583K)	176K	11.0%	>99.0%
- Urban		(127,681K)	(125,780K)	(1,901K)	(2.0%)	(>99.0%)
+ Regular		(249K)	(182K)	(67K)	(37.0%)	(3.5%)
+ Extreme		(22,299K)	(22,413K)	115K	1.0%	6.0%
+ Moderation		(34,772K)	(35,184K)	411K	1.0%	21.6%
+ Convenience		(70,362K)	(68,002K)	(2,360K)	(3.0%)	(>99.0%)

You can use the 'Number formatting' card to apply custom formatting to the measures.

Number Formatting (PRO)

Decimal Separator  
.

Thousand Separator  
,

Prefix

Suffix

Value decimal places  
0

Scaling Display  
Auto

Customize Scaling Label  
On

Thousands  
K

Millions

The same card also has a section called 'Semantic Formatting' that lets you change the colour and format based on whether the value is negative or positive.

Semantic Formatting  
On

Negative Value Format  
(X)

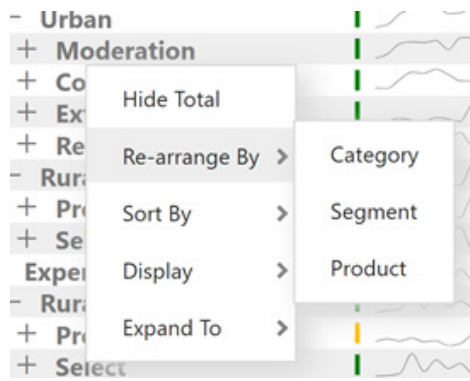
Color for Negative Values  
Red

Positive Value Format  
X

Color For Positive Values  
Green

[Revert to default](#)

Right-clicking on the row headers will allow you to do things like rearrange the order of the hierarchy, hide or show totals, change the sort order, expand to a specific level of the hierarchy or show and hide different column headers.



Other formatting options include:

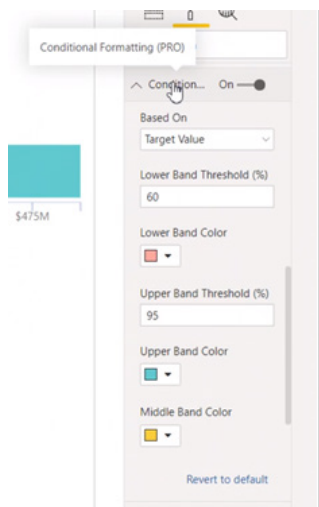
- Text and background colours for alternating rows
- Renaming headers
- Hiding empty rows
- Adding a search bar (only for the Power BI Service).

### LINEAR GAUGE

The second visual is the Linear Gauge (similar to an horizontal Bullet Chart), which allows you to compare a value to a target goal.



In the 'Conditional formatting' card of the Formatting pane, you can set the band thresholds and colours.



You can also customise the marker shape, size and offset from the bottom, for both the actual and target in the 'Gauge Options' card.

Actual Value Marker

Triangle

Line

Circle

Square

Triangle

Diamond

20

Target Value Marker

Line

Marker Size

8

Marker Vertical Offset

Auto

Other formatting options include:

- Setting minimum and maximum values
- Axis formatting, including text size, colours and label density
- Data label formatting.

## TAG CLOUD

The third of these visuals is the Tag Cloud. This Word Cloud visual allows you to get instant insights of the more prevalent terms in your data, such as product descriptions, tweets and feedback comments.



Some key features and options include:

- Split phrases into individual words
- Remove special characters
- Exclude words
- Format the word size, colour, angles and number of orientations.

You can also control the colour and size of the words based on different metrics.

### ***Sagra Emigo connector Generally Available***

Emigo supports sales organisations and field teams management. It automates daily activities of field professionals. Inventory data, orders, answers from surveys or all sorts of other digital forms are available for

managers and other teams within organisation, through the Biqsens Power BI app, as well as through Emigo BI Connector for Power BI Desktop.

### ***Azure Cost Management connector updated***

Azure Cost Management enhances exploration and analysis of your Azure cost data. With it, you can make powerful and flexible reports to understand your consumption patterns.

What's new:

- Supports OAuth 2.0 for authentication
- Supports both Microsoft Customer Agreement and Enterprise Agreements (EA)
- Includes enhancements to RI recommendations and RI usage data and the addition of amortized cost
- Performance improvements.

### ***New Workplace Analytics connector***

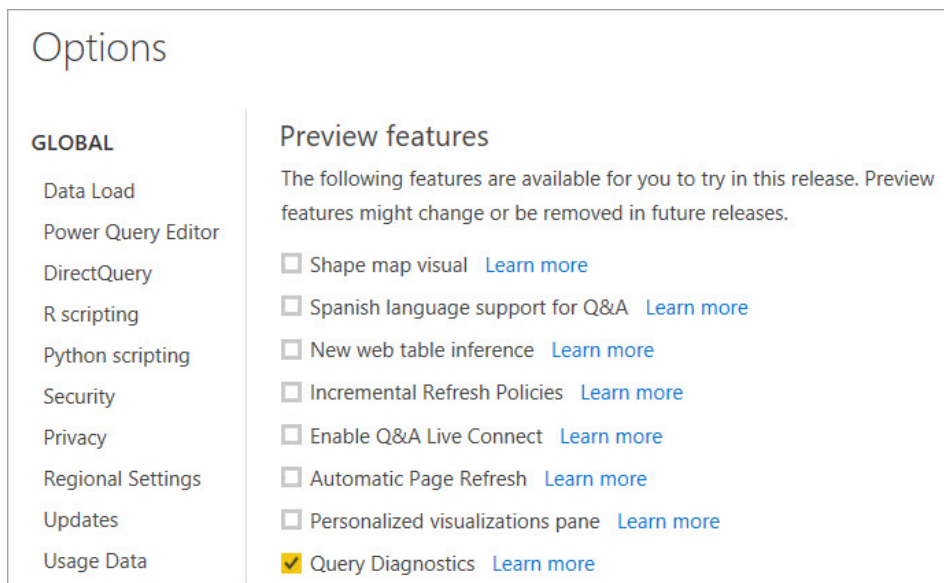
Microsoft Workplace Analytics uses data from everyday work in Office 365 to identify collaboration patterns that impact productivity, workforce effectiveness and employee engagement. This addition may help your customers speed transformation using new insights about the way their people and teams work.

### ***Query Diagnostics***

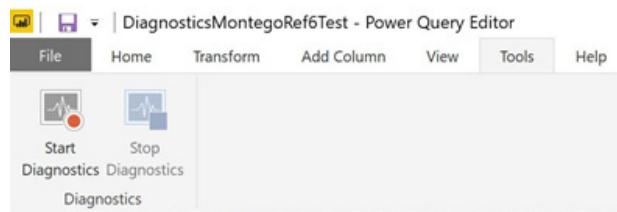
Query Diagnostics is a new feature that will allow you to determine what the Mashup Engine is doing during authoring time. While Microsoft will be expanding on this feature in the future, including allowing you to use it during full refreshes, at this time it allows you to understand what sort

of queries you are emitting, what slowdowns you might run into during authoring refresh, and what kind of background events are happening.

To enable it, enable 'Query Diagnostics' under 'Preview Features' in Options.



To start diagnostics, in the Power Query Editor click on the Tools pane in the Ribbon and select 'Start Diagnostics'. To finish and aggregate the traces, press 'Stop Diagnostics'. Whenever you start diagnostics, Power Query will begin tracing any evaluations that you cause.



The evaluation that most users think of is when you press refresh, or when you retrieve data for the first time, but there are many actions that can cause evaluations depending on the connector. For example, with the SQL connector, when you retrieve a list of values to filter, that would

kick off an evaluation as well—but it doesn't associate with a user query, and that's represented in the diagnostics. Other system-generated queries might include Navigator or the 'Get Data' experience.

With this first version of the feature, Microsoft has presented two different diagnostic views, a summarized and a detailed view:

- The **basic view** is aimed to give you an immediate insight into where time is being spent in your query
- The **detailed view** is much deeper, line by line, and will generally only be needed for serious debugging by power users.

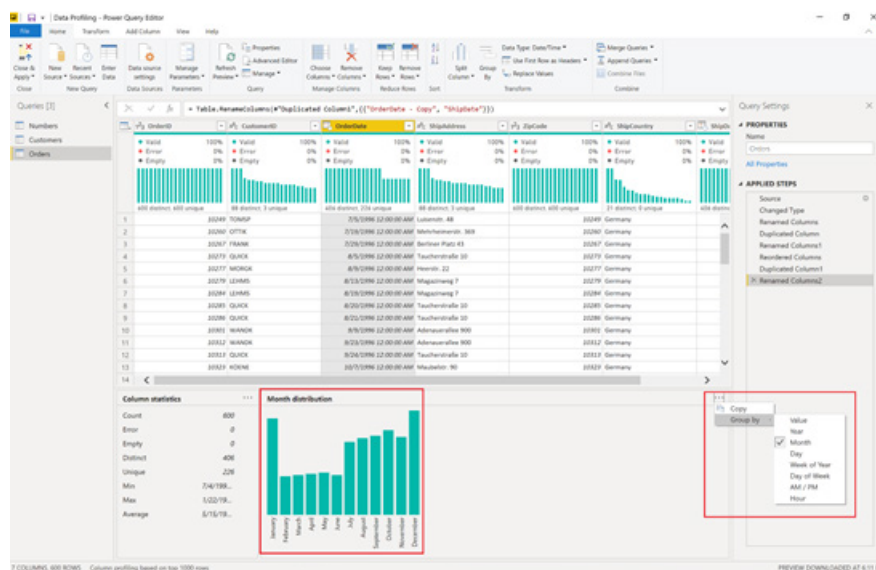
Some capabilities, like the 'Data Source Query' column, are currently available only on certain connectors. Microsoft will be working to extend the breadth of this coverage in the future.

- Additional Info
- Category
- Σ Content Length
- Data Source Kind
- Data Source Query
- ▶ 📅 End Time
- Σ Exclusive Duration
- Σ Exclusive Duration (%)
- Σ Group Id
- Id
- Is User Query
- Operation
- Path
- Query
- Resource
- Σ Row Count
- ▶ 📅 Start Time
- Step

## Data profiling enhancements

This update also sees a couple of significant enhancements introduced to the Data Profiling capabilities within the Power Query Editor, including:

- **Multiple Grouping options for the 'Column Profile' pane value distribution visual**, specific by column type, in addition to the existing 'By Value' criteria:
  - o **Text**: by Text Length (number of characters)
  - o **Number**: by Sign (positive / negative) and Parity (even / odd)
  - o **Date / DateTime**: by Year, Month, Day, Week of Year, Day of Week, AM/PM Time and Hour within a day
  - o and more for other data types (Logical True/False), etc.

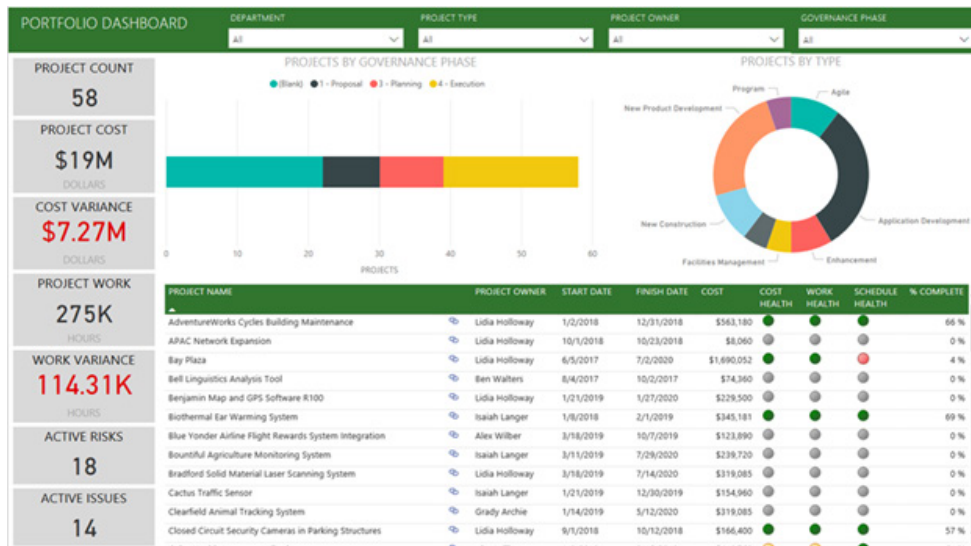


- **Filter options**: not only can you now leverage several type-specific grouping criteria within the 'Column Profiles distribution' pane but also Power BI Desktop has now enabled filtering from within the callouts for each of the values in the distribution chart when the grouping criteria is applied. For example, you can easily exclude all values that fall in a given Month right from the 'Data Profiles' pane for a Date / DateTime column.

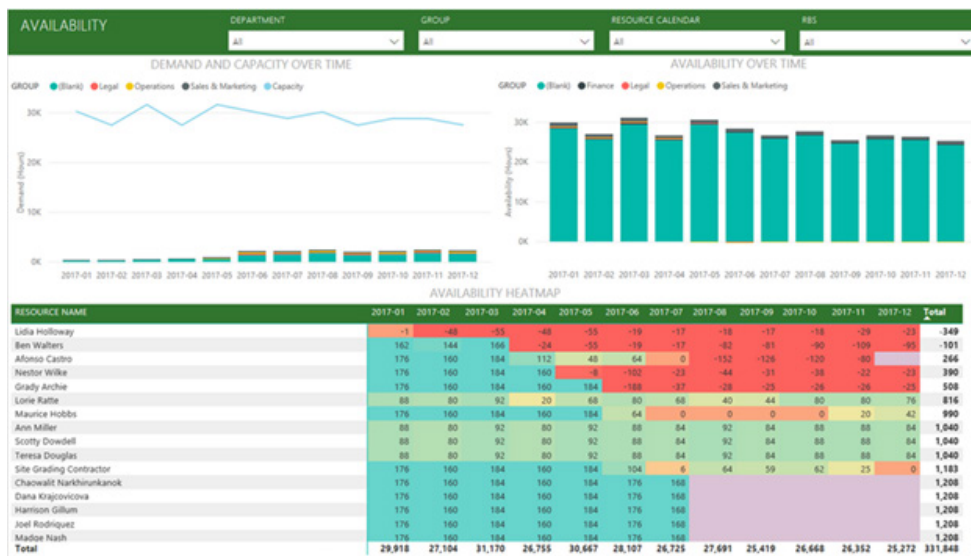
## Project Web App

This application is aimed at helping project, portfolio and resource managers who work in Project Web Apps to visualise their work. The report model (below) was built using the Project Web App data connector.

It can be connected to your own Project Web App data using your own credentials. This template app is comprised of 13 report pages to support such needs.



As another example, here's a Resource Management report page:



The reports enable users to do a deep-dive analysis on project management, resource management and portfolio management views. Overall, this report provides a good starting point for managers using Project Web Apps (PWA) to take their data to the next level.

This template includes six report pages offering analysis on portfolios of projects including views of a Portfolio Dashboard, Portfolio Timeline, Portfolio Costs, Portfolio Milestones and Portfolio Issues and Risks.

It also includes five report pages with analysis on resourcing across projects including views of a Resource Overview, Resource Availability,

Resource Assignments, specific Resource Details and a predictive analysis Resource Demand Forecast.

For individual projects, there are two report pages with detailed project information for single projects including a dive on Project Status and Project Issues and Risks.

You can install the app from the Power BI Service and set its parameters to your PWA site URL, and the language your site is in. Once you install it, the report is then yours.

### New file format: .pbids

This latest update witnesses the release of a new file format: **PBIDS**, to streamline the 'Get Data' experience for report creators in your organisation. It's recommended that administrators create these files for commonly used connections to help the 'Getting Started' experience for new report authors.

When an author opens a **.PBIDS** file, Power BI Desktop will launch and prompt for authentication to connect to the data source specified in the file. Then, the navigation dialog will pop-up and the user will need to select the tables to load into the model. They may also need to select the

database(s) if one was not specified in the file. From there the user can start building visualisations or revisit 'Recent sources' to load a new set of tables into the model.

For the administrator to create this file, they will need to specify in the file the required inputs for a single connection and they can specify the mode of the connection. The 'mode' may either be DirectQuery or Import. If 'mode' is missing / null, the user that opens the file in Desktop will be prompted to select DirectQuery or Import.



Here is an example of a PBIDS file for a SQL Server connection:

```
{
  "version": "0.1",
  "connections": [
    {
      "details": {
        "protocol": "tds",
        "address": {
          "server": "<<server-name>>",
          "database": "<<db-name (Optional)>>"
        }
      },
      "options": {},
      "mode": "DirectQuery"
    }
  ]
}
```

Here's another example for a SharePoint list. Please note that the URL must point to the SharePoint site itself and not to a list within the site. The user will receive a navigator which allows them to select one or more lists from that site which will each become a table in the model.

```
{
  "version": "0.1",
  "connections": [
    {
      "details": {
        "protocol": "sharepoint-list",
        "address": {
          "url": "<<Sharepoint Site URL>>"
        }
      }
    }
  ]
}
```

### **Performance improvements for modelling operations**

Finally, Microsoft has made a performance improvement in the Analysis Services engine to speed up modelling operations (such as adding measures or calculated columns, creating relationships, etc.) when the model satisfies the following conditions:

- Contains many calculations
- Uses any of the **UseRelationship** or **CrossFilter** functions
- Uses any of the **Today**, **Now**, **Random**, etc. volatile functions.

While the actual improvement will be highly dependent on the model, Microsoft reports that they measured a 20x performance improvement for some customers when doing things like opening the file and adding a measure.

More next month, we're sure...

## **Power BI Report Server Updates**

It's not every month that this comes out, but this release is packed with many reporting features to make up for the regularity of updates. In this release, there's the new updated pane design, the new filtering experience, and more conditional formatting options.

Here's the complete list of the latest updates:

## Reporting

- Updated pane design
- Visual level filters for slicers
- Sorting for the 'Performance Analyzer' pane
- Visual header ToolTips
- Table and matrix total label customisation
- Sync slicer support for hierarchy slicer
- Consistent font sizes across visuals
- Icon sets for table and matrix
- Percent support for conditional formatting by rules
- New Filter pane is now Generally Available
- Data colors support when using play axis on scatter charts
- Performance improvements when using relative date and dropdown slicers
- Grouping visuals
- Color and text classes in themes
- New default themes

## Analytics

- Custom format strings
- Conditional formatting updates for formatting options:
  - o Visual background and title colors
  - o Card colors
  - o Gauge fill and colors
  - o Alt text
  - o Border color
- Conditional formatting warnings
- Drill through discoverability improvement
- New DAX expressions: **REMOVEFILTERS** and **CONVERT**
- New DAX comparison operator: **==**

## Data preparation

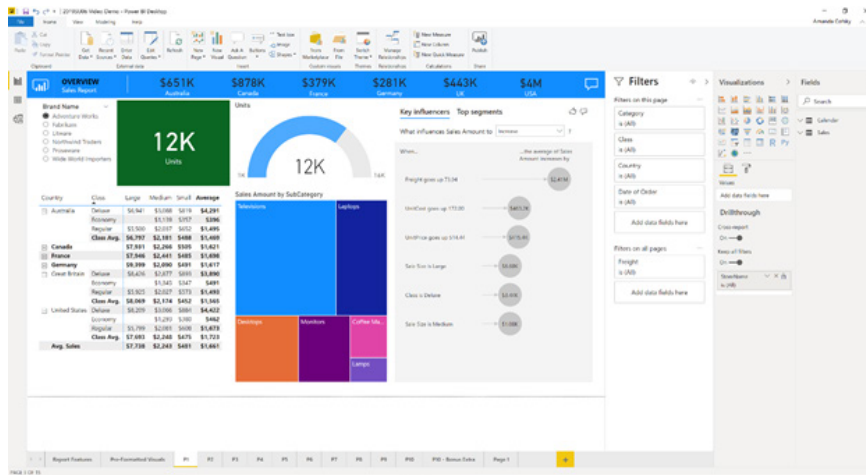
- Improvements to **M** Intellisense
- New Transformation: Split column by positions
- Copy to clipboard from data profiling.

Let's go through each in turn.

## Updated pane design

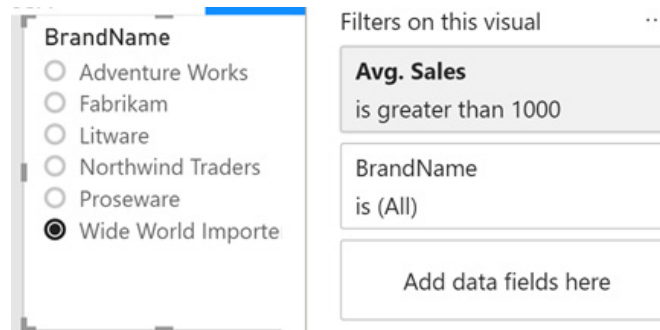
The interface had been refreshed. This updates all panes, the footer, and view switcher to be a lighter color, plus there is updated spacing and new

icons. This new look is intended to provide a more 'modern' experience and is the first step in refreshing the entire interface.



## Visual level filters for slicers

You can now add a visual level filter to slicers. This has been a popular idea on [ideas.powerbi.com](https://ideas.powerbi.com). It works just like any other visual level filter. It will just filter the slicer itself and no other visuals. This can be useful for filtering out blanks or if you want to use measure filters.

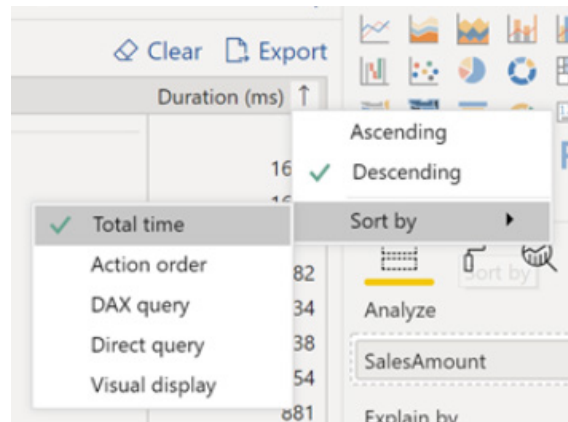


The filter that the slicer applies to other visuals doesn't change: it's still just the selection made of the value in the slicer. In the example above, Power BI doesn't also wrap the 'Avg. Sales > 1000' filter into the filter applied to other visuals.

## Sorting for the 'Performance Analyzer' pane

There's a small update to the new 'Performance Analyzer' pane this release as well. You can now change the sort direction and pick to sort between:

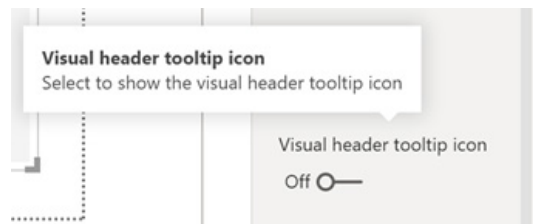
- Total time spent
- Order the actions occurred
- Time spent for the DAX query
- Time spent querying the source (for DirectQuery)
- Time spent displaying the visual.



## Visual header ToolTips

Microsoft has noted that users add helpful information for their end users to reports, such as hints on how to interact with the report or descriptions of the fields used in the visual. Setting that up though used to take a fair amount of work, so Microsoft has now added a way to add a ToolTip directly to the visual's header.

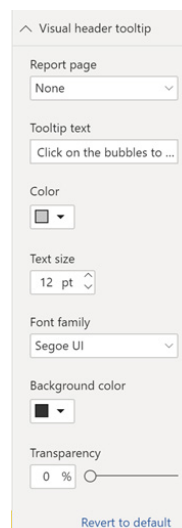
This option is off by default, but you can turn it on through the 'Visual header tooltip icon' toggle in the 'Visual header' card of the Formatting pane. This is the one toggle within the card that affects whether the icon shows in edit mode as well as reading mode.



Once the toggle is on, you'll see a question mark icon in the visual's header,



and you'll see a new Formatting card in the pane called 'Visual header tooltip'. In this card, you can add the text to show in the ToolTip and the formatting of the ToolTip. Alternatively, you can pick a report page ToolTip instead if you require more customisation.

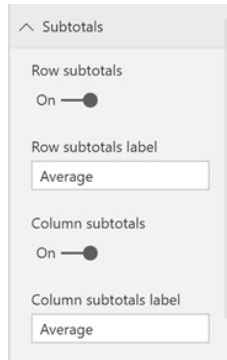


Once you've added ToolTip text or selected a report page ToolTip, you'll see it on hover over the question mark icon.

### Table and matrix total label customisation

One of the most popular requests in the forums recently has been for allowing the total labels for table and matrix to be customisable. Therefore,

you now have the option to change the label under the subtotals card for both columns and rows.



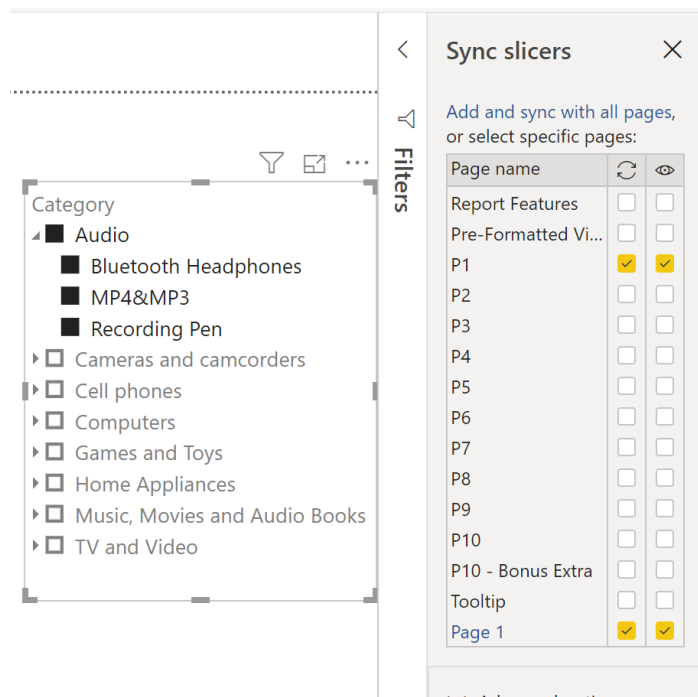
You can also customise the label for each subtotal field when stepped layout is off.



### Sync slicer support for hierarchy slicer

The hierarchy slicer, one of the most used marketplace visuals, allows users to display a hierarchy within their slicer for optimised navigation. This is a very powerful visual, but until recently, it had a limitation where it couldn't work with Microsoft's sync slicer feature. With this update, the sync slicer feature has been improved to support hierarchy slicers

now. This support could also extend to other marketplace visuals, if those visuals add support for the right capabilities. Some things to be aware of are that you can't sync two hierarchy slicers that have different columns from your model, and if you remove columns from one slicer, it will not remove it from the other synced slicers.



### Consistent font sizes across visuals

If you've been building reports for a while, you may have noticed that when you set the font size to a specific size, such as '8', the size might look different, depending on where you've been setting it. The reason for this is that different places were either using pixels or points to

represent the font size, and eight (8) pixels is a different size compared to eight (8) points. This has now been updated so that all the formatting options within the visuals will always use points. Yay!

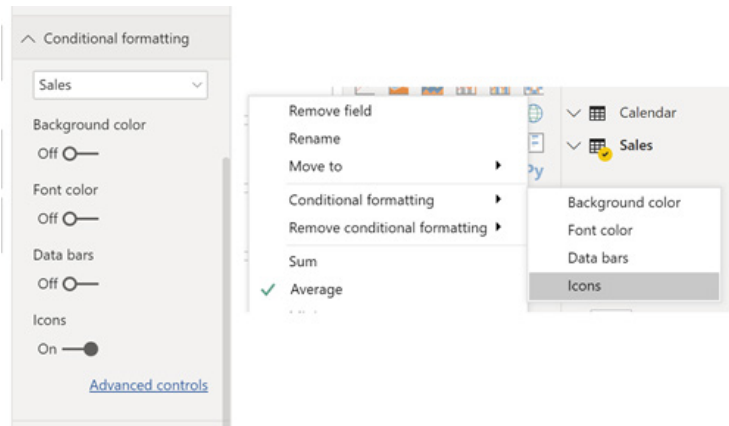
### Icon sets for table and matrix

The top request on Microsoft's Ideas site is for KPI icons for reports. This update sees the release of these icon sets for the table and matrix visual.

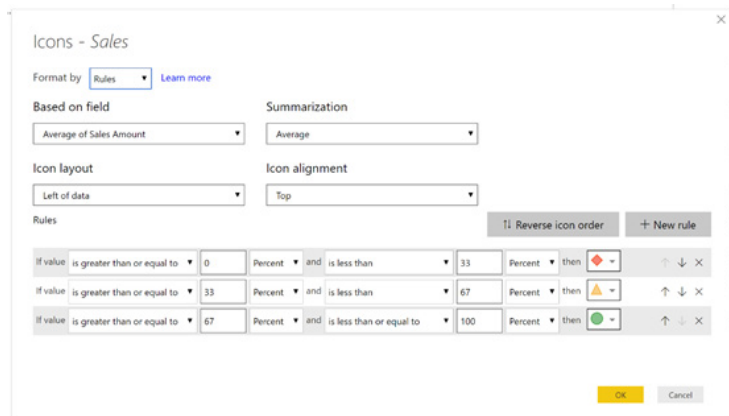
This feature will let you set up rules for showing different sets of icons to show in your table and matrix, very similar to the Icon Sets feature in Excel.

Country	Large	Medium	Small	Total
<b>Australia</b>	<b>\$8,098</b>	<b>\$2,090</b>	<b>\$358</b>	<b>\$1,052</b>
Deluxe	● \$8,369	◆ \$2,479	◆ \$476	<b>\$2,646</b>
Economy		◆ \$1,640	◆ \$251	<b>\$315</b>
Regular	● \$7,135	◆ \$2,004	◆ \$445	<b>\$1,098</b>
<b>Canada</b>	<b>\$8,454</b>	<b>\$2,098</b>	<b>\$366</b>	<b>\$1,015</b>
<b>France</b>	<b>\$8,539</b>	<b>\$2,099</b>	<b>\$352</b>	<b>\$1,049</b>
<b>Germany</b>	<b>\$8,430</b>	<b>\$2,045</b>	<b>\$358</b>	<b>\$1,084</b>
<b>Great Britain</b>	<b>\$8,453</b>	<b>\$2,124</b>	<b>\$350</b>	<b>\$1,075</b>
Deluxe	● \$8,859	◆ \$2,505	◆ \$460	<b>\$2,724</b>
Economy		◆ \$1,765	◆ \$237	<b>\$329</b>
Regular	● \$7,012	◆ \$2,039	◆ \$443	<b>\$1,114</b>
<b>United States</b>	<b>\$8,299</b>	<b>\$2,077</b>	<b>\$361</b>	<b>\$1,050</b>
Deluxe	● \$8,662	◆ \$2,506	◆ \$490	<b>\$2,720</b>
Economy		◆ \$1,519	◆ \$251	<b>\$316</b>
Regular	● \$6,977	◆ \$1,998	◆ \$454	<b>\$1,109</b>
<b>Total</b>	<b>\$8,323</b>	<b>\$2,085</b>	<b>\$360</b>	<b>\$1,049</b>

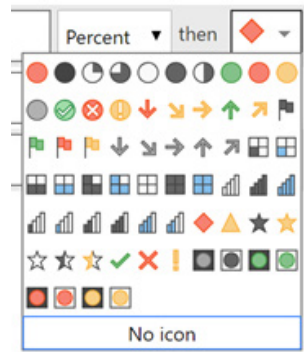
You can turn on icons for your table or matrix just like any other conditional formatting feature, either through the contextual menu for the field you want to apply it to or through the 'Conditional formatting' card in the Formatting pane.



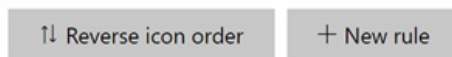
When you turn on Icons, the default rules will apply a red diamond to the lower 33% of your data, a yellow triangle to the middle 33%, and a green circle to the upper 33%.



For each rule you can set the range based on an absolute number or a percentage. You can also customise the icon for each rule with any of the icons Microsoft supports in the icon picker.



In addition to adjusting the current rules, you can add new rules to the list using the 'New rule' button and swap the order of the icons in all the rules with one click using the 'Reverse icon order' button.



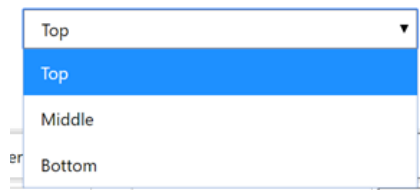
In this dialog, you can also change the field used to generate the icons, of the numbers showing in the cell or if the icons should replace the numbers all together. using the Icon layout dropdown, if the icon shows to the left or right

#### Icon layout



You can also control the icon alignment, switching between top, middle and bottom alignment.

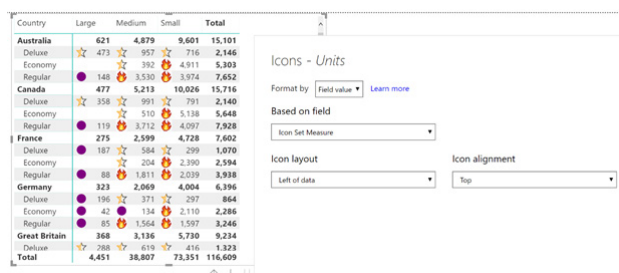
#### Icon alignment



You can also format based on fields in your model, where the fields use the name of any of the existing icons, defines svgs inline, or references an image or gif online. For example, you can create a measure like:

```
Icon Set Measure =
SWITCH (
    TRUE(),
    SUM(Sales[Units]) < 200, "data:image/svg+xml;utf8, <svg xmlns='https://www.w3.org/2000/svg'
viewBox='0 0 100 100'> <circle cx='50' cy='50' r='40' stroke='purple' stroke-width='4' fill='purple'
/> </svg>",
    SUM(Sales[Units]) < 1000, "StarMediumLight",
    "https://i.gifer.com/Omjx.gif"
)
```

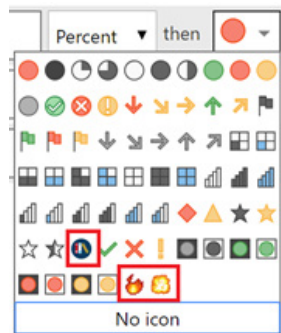
and apply it to get a completely custom set of icons.





You can also change icons or add new ones through a theme file. For example, the below theme will add two new icons and replace the half-filled yellow star with a different icon:

```
{
  "name": "CustomIcons",
  "icons": {
    "fire": {
      "url": "https://i.gifer.com/Omjx.gif",
      "description": "Fire!"
    },
    "explosion": {
      "url": "https://i.gifer.com/3iCN.gif",
      "description": "Boom!"
    },
    "StarMediumLight": {
      "url": "https://www.uscsd.k12.pa.us/cms/lib/PA01000033/Centricity/Domain/1/library-
icon.png",
      "description": {"resourceKey": "Icon_ColoredArrowUpRed"}
    }
  }
}
```



Each icon you are adding or modifying will get its own group in the icons section of the theme. The structure of each icon is:

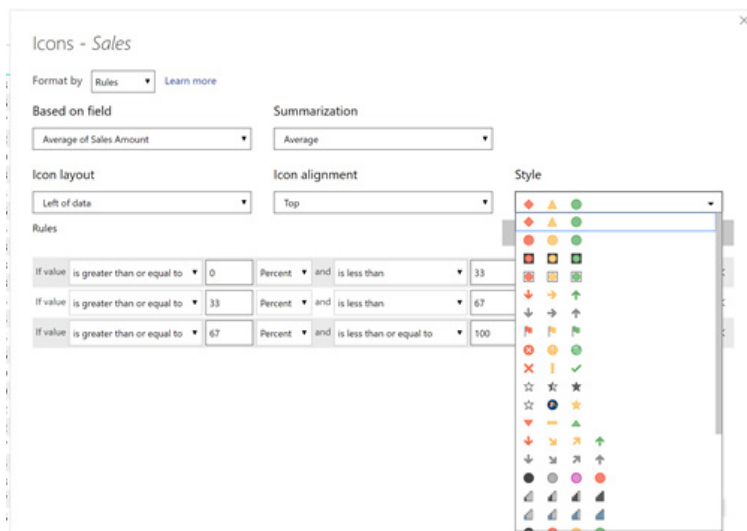
```
"IconID": {
  "url": " https://i.gifer.com/3iCN.gif ",
  "description": "Boom!"
}
```

The IconID section is either a unique name you are giving a custom icon or the id of any of Microsoft’s built-in icons you want to modify. The URL is where you’d include either a link to web image you want to reference or embed your SVG or base64 encoded image. Lastly, the description is what will be used for the tooltip of the icon and read out by a screen reader.

```
"data:image/svg+xml;utf8, <svg xmlns='https://
www.w3.org/2000/svg' viewBox='0 0 100 100'>
<circle cx='50' cy='50' r='40' stroke='purple'
stroke-width='4' fill='purple' /> </svg>"
```

If you are using SVGs either through measures or the theme file, make sure to include the viewBox section of the code, where the viewBox is the original canvas size that was used when the SVG icon was made. This will ensure the SVG will continue to scale as the font size of your values changes.

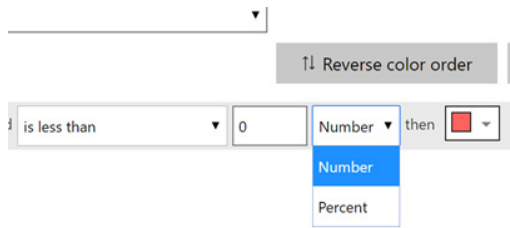
Microsoft has also added an icon set picker to the conditional formatting dialog. This will give you a fast way to select a set of icons and rules to start with and then customise from there, which should save you some time manually selecting icons for each individual rule.



### Percent support for conditional formatting by rules

In addition to adding a new type of conditional formatting, this update extends in general the Rules form of conditional formatting to support Percent, which allows you to apply formatting on a dynamic range of data, instead of having to specify an absolute number that can become

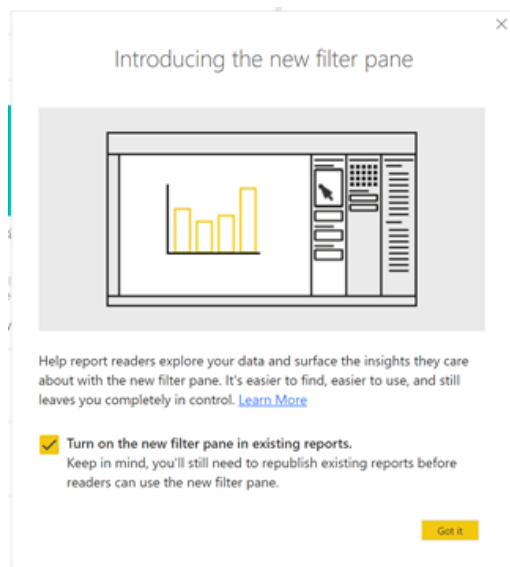
absolute as data refreshes. It has also added the 'Reverse color order' button from the Icon dialog for colours as well, so it's easier to quickly swap the direction of your rules.



### New Filter pane is now Generally Available

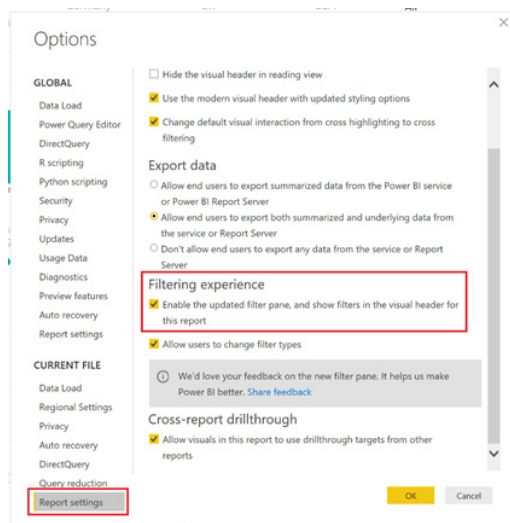
The new Filter pane is now Generally Available in Power BI Report Server. Any new reports should automatically have the new pane on by default and Microsoft is starting the process to transition current reports to the new pane as well. Apparently, this will be performed in a way that will minimise the impact on your current reports. With that in mind, when

you open Power BI Report Server for the first time, the software will show you a Filter pane auto-update dialog. Make sure to take advantage of this opportunity, as eventually all existing reports will be updated to the new Filter pane.



Through this dialog, you can opt into using the new Filter pane experience. Once you do this, anytime you open an old report it will show the new Filter pane automatically. If you save the report after opening it, the new pane will now persist, and once you publish the report again, it will

show for you end users. If you're not ready to upgrade a specific report, you can turn off the new pane through the 'Report level' settings in the Options dialog.



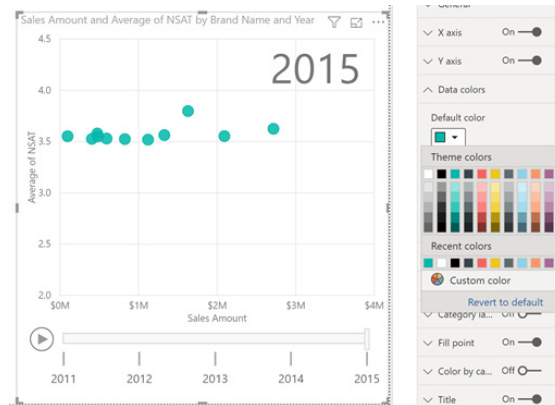
If you're not ready to switch to the new experience, you can close out the auto-update dialog or choose to opt out at this time. You'll still see the

old pane for your reports, but when you're ready, you can opt in through the dialog the next time you open a report.

### Data colors support when using play axis on scatter charts

As a smaller formatting improvement this release, you can now set the data color for the scatter chart when using the play axis. Up until now, you could only use data colors when using a scatter chart without a play

axis, so this should add more flexibility when designing your reports if you're using the play axis.



### Performance improvements when using relative date and dropdown slicers

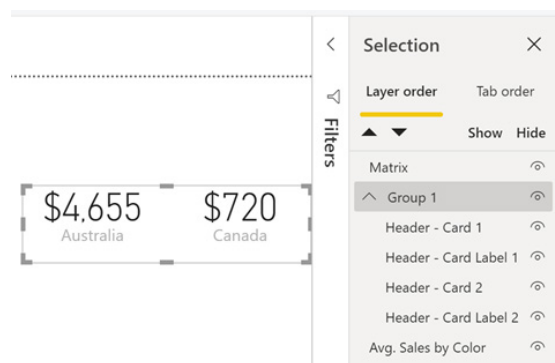
If you are using the relative date slicer or the dropdown slicer, you should see some performance improvements for page loads and filtering interactions this release. The relative date slicer will no longer run any

DAX queries when loading, and the dropdown slicer will only run queries when you expand the dropdown. The more slicers you had on your page, the bigger performance improvement you should see.

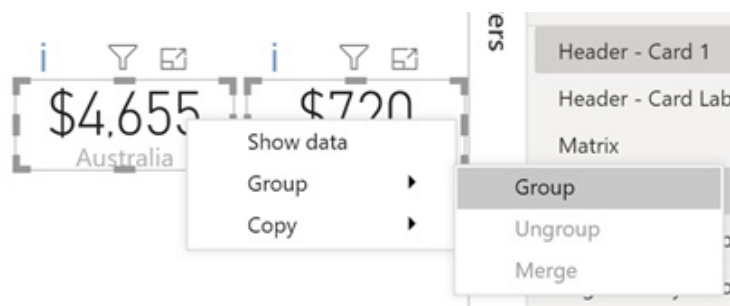
### Grouping visuals

This update includes the release of one of the most welcomed demos from the Microsoft Business Applications Summit: grouping. Grouping allows you to group visuals, shapes, textboxes, images and buttons together on your report page just like you can do in PowerPoint. Once

objects are grouped together, you'll be able to move and resize them all together. This should make it a lot easier to work with reports with lots of objects layered on each page.



To group visuals, you can **CTRL + Click** them and right-click to select the Group option.



Right-clicking a group will give you an option to ungroup them later.

Generally, by default, the behaviour mimics PowerPoint, such as:

- Clicking on the empty space within the group will not select anything
- Clicking on a visual within a group will select the entire group on first click and the individual visual on second click
- Selecting a group and another object on the page and then selecting the Group option will create a nested group
- Use the selection pane to hide everything in the group at once (and bookmark it if you).

There are a few ways the feature has been extended as well, including:

- Use the Selection pane to drag and drop objects in and out of groups
- Selecting two groups and opening the context menu will give you an option to merge the groups into one group instead of nesting them
- Using the Formatting pane, you can add a background colour to a group. If a group has a background, it now has “substance” so you can click on the background of the group to select it.



Groups, since they aren't actual content on the report page, don't affect the tab order of your report page. Grouping is also only supported on reports using our new modern header, so if you have an older report

still using the previous header, upgrade it in the Options dialog to take advantage of this feature.

### Color and text classes in themes

When writing your own custom theme JSON file, it can be quite difficult to build a theme that changes the entire look of your report. For example, if you wanted to create a dark theme that set the page to a dark color and use white text on top, you'd have to change numerous settings through the visual styles section of the JSON, which can be very time consuming and error prone.

To help with this, Microsoft has expanded the color classes within their theme file to allow you to format many of the colours in your theme without needing to touch visual styles. If you're already writing theme files, you're likely already using a few existing color classes: foreground, background, and **tableAccent**, which update various settings within your report with one line. With this update, we're adding new color classes to that list, which as a result, means you can colour all visual elements in a report just by setting six colours.

To summarise:

- **foreground**
  - o Labels background color (when outside data points)
  - o Filter pane and Filter cards font and icon colors
  - o Trend line color
  - o Textbox default color
  - o Table and matrix values and totals font colors
  - o Data bars axis color
  - o Card data labels
  - o Gauge callout value color
  - o KPI goal color
  - o KPI text color
  - o Slicer item color (when in Focus mode)
  - o Slicer dropdown item font color
  - o Slicer numeric input font color
  - o Slicer header font color
  - o Scatter chart ratio line color
  - o Line chart forecast line color
  - o Map leader line color
- **foregroundNeutralSecondary**
  - o Label colors
  - o Legend label color
  - o Axis label color
  - o Table and matrix header font color
  - o Gauge target and target leader line color
  - o KPI trend axis color
  - o Slicer slider color
  - o Slicer item font color
  - o Slicer outline color
  - o Line chart hover color
  - o Multi-row card title color
  - o Ribbon chart stroke color
  - o Shape map border color
  - o Button text font color
  - o Button icon line color
  - o Button outline color
- **foregroundNeutralTertiary**
  - o Legend dimmed color
  - o Card category label color
  - o Multi-row card category labels color
  - o Multi-row card bar color
  - o Funnel chart conversion rate stroke color
- **backgroundLight**
  - o Filter card background color for applied filters
  - o Axis gridline color
  - o Table and matrix grid color
  - o Slicer header background color (when in Focus mode)
  - o Multi-row card outline color
  - o Shape fill color
  - o Gauge arc background color
- **backgroundNeutral**
  - o Table and matrix grid outline color
  - o Shape map default color
  - o Ribbon chart ribbon fill color (when match series option is turned off)
- **background**
  - o Labels background color (when inside data points)
  - o Filter pane and available filter card background color
  - o Slicer dropdown items background color
  - o Donut chart stroke color
  - o Treemap stroke color
  - o Combo chart background color
  - o Button fill color
- **TableAccent**
  - o Overrides table and matrix grid outline color when present.

In addition to color classes, this update has added text classes to make it easier and quicker to set text styles. There are four primary text classes that adjust a total of 14 total text classes. You can set the font family, font size and font color for each text class. The four main text classes are: "Title", "Label", "Callout" and "Header".

The Title and Label classes have several secondary classes that are automatically derived from the primary class settings but can be formatted individually. For example, if you set the label class, which for

example is used for the values in a table to 10pt, a "Small Label", which is for example used the search box text in slicers, would be automatically set to 9pt. The goal of the primary / secondary text classes is to make it quicker and easier for you to set the text classes, which still maintaining a visual hierarchy to the text.

The below table shows the primary classes with example settings and the secondary classes with the settings that make them unique compared to their associated primary class.

Primary class	Secondary classes	Class name in JSON	Settings	Associated visual objects
<b>Callout</b>	N/A	callout	DIN #252423 45pt	Card data labels KPI indicators
<b>Header</b>	N/A	header	Segoe UI Semibold #252423 12pt	Key influencers headers
<b>Title</b>		title	DIN #252423 12pt	Category axis title Value axis title Multi-row card title * Slicer header
	Large title	largeTitle	14pt	Visual title
<b>Label</b>		label	Segoe UI #252423 10pt	Table and matrix column headers Matrix row headers Table and matrix grid Table and matrix values
	Semibold	semiboldLabel	Segoe UI Semibold	Key influencers profile text
	Large	largeLabel	12pt	Multi-row card data labels
	Small	smallLabel	9pt	Reference line labels * Slicer date range labels Slicer numeric input text style Slicer search box Key influencers influencer text
	Light	lightLabel	#605E5C	Legend text Button text Category Axis labels Funnel chart data labels Funnel chart conversion rate labels Gauge target Scatter chart category label Slicer items
	Bold	boldLabel	Segoe UI Bold	Matrix subtotals Matrix grand totals Table totals
	Large and Light	largeLightLabel	#605E5C 12pt	Card category labels Gauge labels Multi-row card category labels
	Small and Light	smallLightLabel	#605E5C 9pt	Data labels Value axis labels

\* The font color for these settings is based on the data colours, not the text class settings.

For example, here's a small theme that sets some of these properties:

```
{
  "name": "NewThemeOptions",
  "foreground": "#252423",
  "foregroundNeutralSecondary": "#605E5C",
  "foregroundNeutralTertiary": "#B3B0AD",
  "background": "#FFFFFF",
```

```

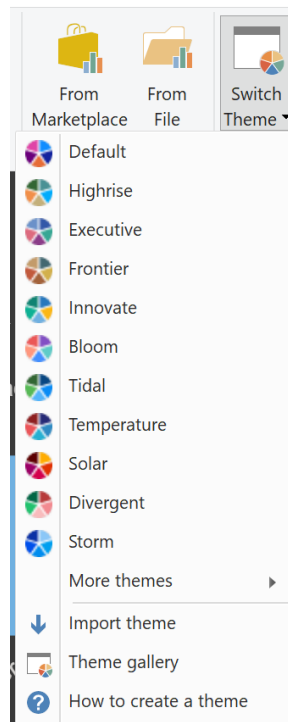
"backgroundLight": "#F3F2F1",
"backgroundNeutral": "#C8C6C4",
"tableAccent": "#118DFF",
"textClasses": {
  "callout": {
    "fontSize": 45,
    "fontFace": "DIN",
    "color": "#252423"
  },
  "title": {
    "fontSize": 12,
    "fontFace": "DIN",
    "color": "#252423"
  },
  "header": {
    "fontSize": 12,
    "fontFace": "Segoe UI Semibold",
    "color": "#252423"
  },
  "label": {
    "fontSize": 10,
    "fontFace": "Segoe UI",
    "color": "#252423"
  }
}
}
}

```

You can still use visual styles to set any specific formatting you want, and that will override any color or text classes that have been set. Microsoft expects with these new changes for most users to stick to the color and text classes the majority of the time, and only use the visual styles for non-color / text options (such as turning titles on / off by default).

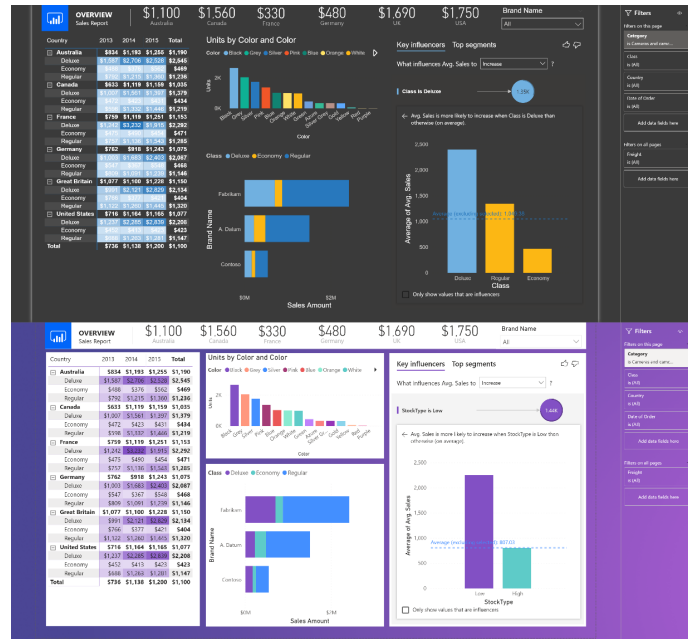
### ***New default themes***

To go along with the new theme JSON options, Microsoft is updating the themes available for reports and changing the default theme for new reports.

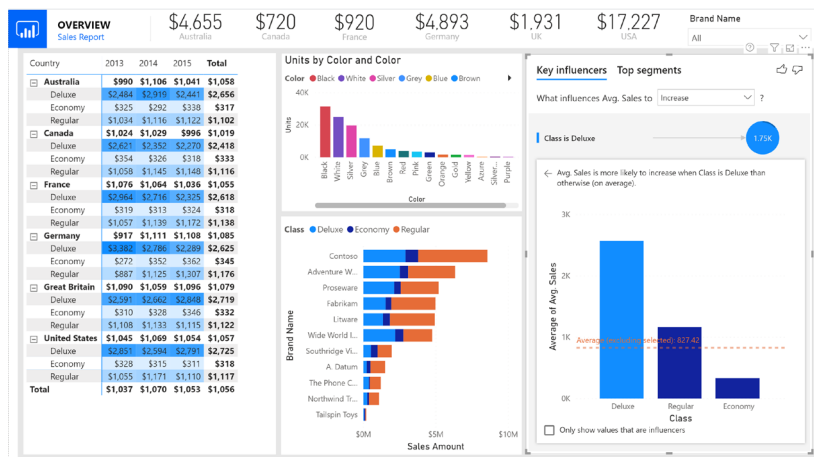




Their Design and Research team spent time developing a new set of themes that have more variety (such as Innovate, a dark theme) and show off more theme-able feature (such as Bloom, with its background image):



The new default theme is meant to both align better with Microsoft's design language and follow best design practices for visuals.



Some visual updates with the new default theme are:

- Larger, darker, more readable text
- Smaller bubble sizes for scatter and map visuals
- Updated layout for pie and donut charts to improve readability
- Expand / collapse on by default for matrices
- Backgrounds on for visuals by default
- Wider line strokes for line and combo charts.

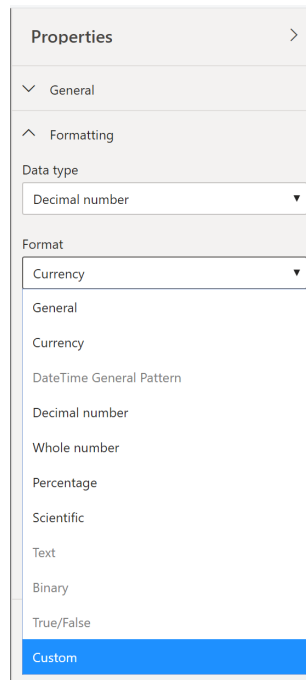
You can still find the previous themes under the 'More themes...' submenu any time you want to reach them. One thing to remember around custom themes is that they always build on top of the default theme. Therefore, if you created your own custom theme that just set the data colours and nothing else, if you imported that theme into a report with the new default applied, it would still have expand / collapse on by

default, backgrounds on for visuals, etc. It may look different compared to when you imported it to a report with the old default theme. If you want your theme to look exactly the same way it did previously, you can first set the theme to 'Classic' in the theme dropdown, so the default theme has the old settings and then import your custom theme.

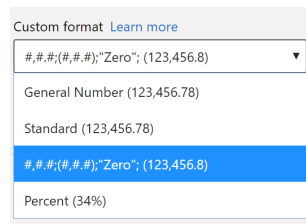
## Custom format strings

This update includes the release of one of the most welcomed demos from the Microsoft Business Applications Summit: grouping. Grouping allows you to group visuals, shapes, textboxes, images and buttons together on your report page just like you can do in PowerPoint. Once

objects are grouped together, you'll be able to move and resize them all together. This should make it a lot easier to work with reports with lots of objects layered on each page.



From there, you'll be able to select from a list of commonly used format strings.



You may use this input box to enter in your own format string:

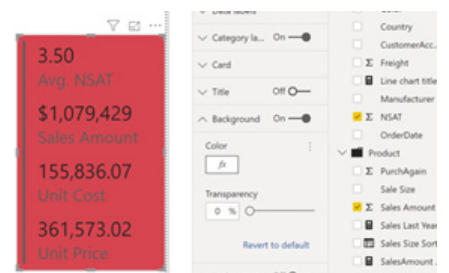


Once you enter in your custom format string, it will immediately be reflected in the Data view and Report view.

## Conditional formatting updates for formatting options

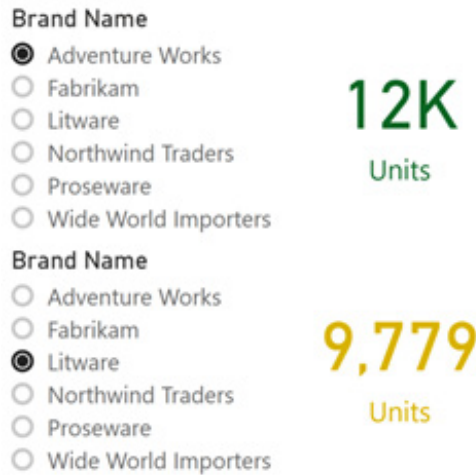
### VISUAL BACKGROUND AND TITLE COLORS

Microsoft has added support for conditionally formatting the visual's background and both the visual title's font and background colours.



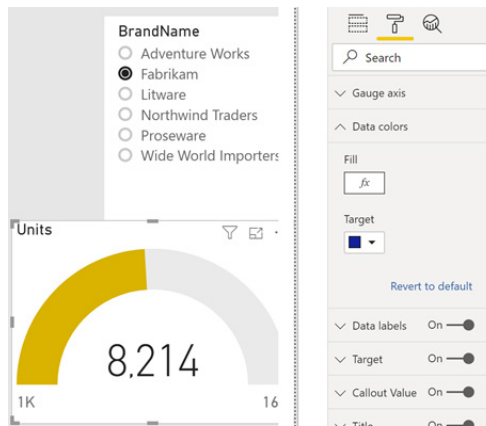
## CARD COLORS

For the Card visual you can now also conditionally format the data label and category label colours.

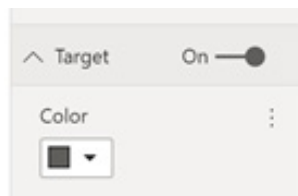


## GAUGE FILL AND COLORS

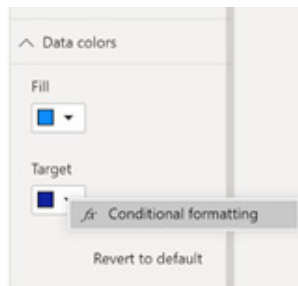
You can conditionally format the fill color of the gauge and other colours used for the gauge visual.



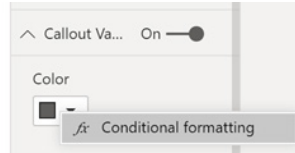
- Target text color



- Target fill color



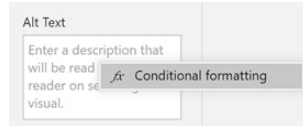
- Callout value color



### ALT TEXT

Alt text is the first feature that now has conditional formatting. Alt text, which is read off by screen readers whenever the visuals are in focus, can now be dynamically changed based on the current filter state of

the visual. You can for example call out in the alternative text both the highest and the lowest categories and have confidence it will always reflect the current data in the visual.



### BORDER COLOR

Border color is the next feature that has this added capability. Just as you can with backgrounds, you can dynamically adjust the border color based on current filter state. This could be a good option to use if you want to

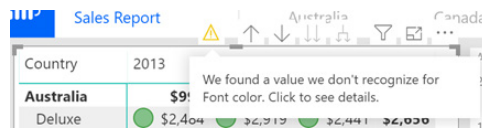
colour a visual based on a specific KPI but you felt like the background colour was too intense an option.



### Conditional formatting warnings

Over the last several months, these updates have expanded the number of places you can apply conditional formatting to include settings, such as titles and font colors, to name just a few. As mentioned before, Microsoft's goal is to eventually support conditional formatting for

most, if not all, formatting properties. Given how prevalent conditional formatting will become with all these changes, they also wanted to make it easy to see errors in your conditional formatting configuration.



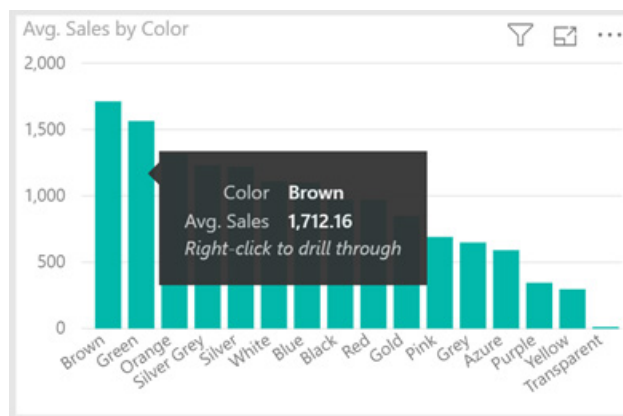
Now, there is a warning in the Visual header and the Formatting pane if one of the following cases is true:

- The measure's data type is not what the formatting property expects (e.g. the measure returns a number, but the property is a string)
- The measure used to conditionally format was deleted
- The measure returns an unrecognized value (e.g. Rad instead of Red).

These warnings will only show up when editing a report. End users will not see these warnings when viewing the report.

### Drill through discoverability improvement

It's true that end users don't always know when they can drill through on a visual to another report page. To help with this, this update adds text to the visual's ToolTip to let them know it's enabled.



## New DAX expressions: REMOVEFILTERS and CONVERT

There are two new DAX expressions this month: **REMOVEFILTERS** is the same as **ALL** when used inside **CALCULATE** to remove filters (so why have it then?), and **CONVERT** changes the expression to another data type.

## New DAX comparison operator: ==

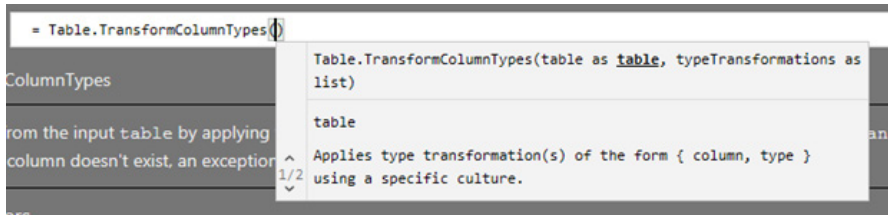
Another DAX update for the month is the addition of a new comparison operator, **==**. This operator is similar to the existing **=** operator, except that it performs strict comparison to **BLANK** values. In DAX, **0 = BLANK()** returns true so a filter like **[Column] = 0** or **[column] = BLANK()** will return

true for both 0 and **BLANK()**. This new **==** operator has been introduced to make it easier to differentiate between 0 and **BLANK()**. **[Column] == 0** will return true only when the column value is 0 and **[Column] == BLANK()** is equivalent to **ISBLANK([Column])**.

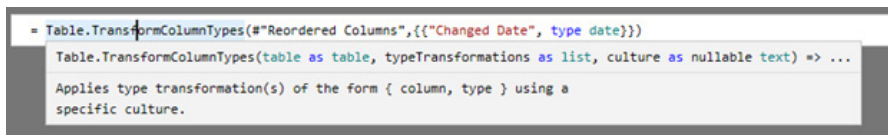
## Improvements to M Intellisense

Based on recent feedback, additional improvements have been made to the **M** Intellisense capabilities that we released a few months ago:

- **Parameter hint support in formula bar:** This previously only worked in the Advanced Editor and Add Custom Column dialogs. This release, sees support for parameter hints added to the Formula bar

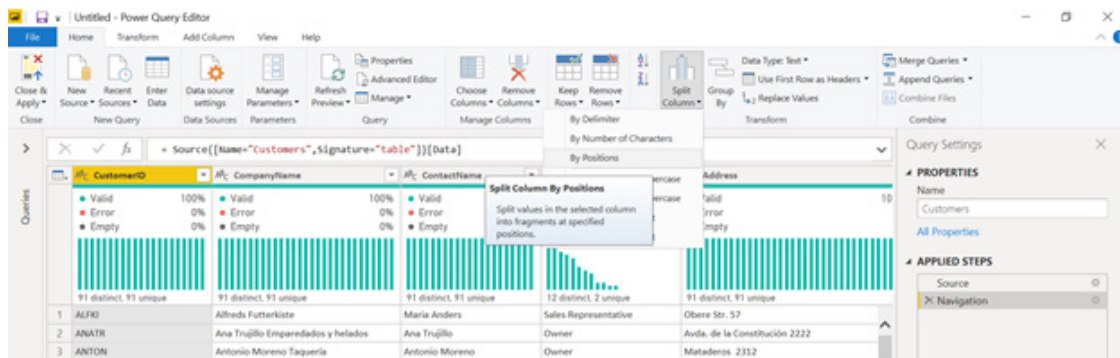


- **Hover support:** Hovering your mouse cursor over identifiers (such as functions and query step names) will display its definition. This works on all **M** Intellisense surfaces (Advanced Editor, Custom Column and Formula bar).

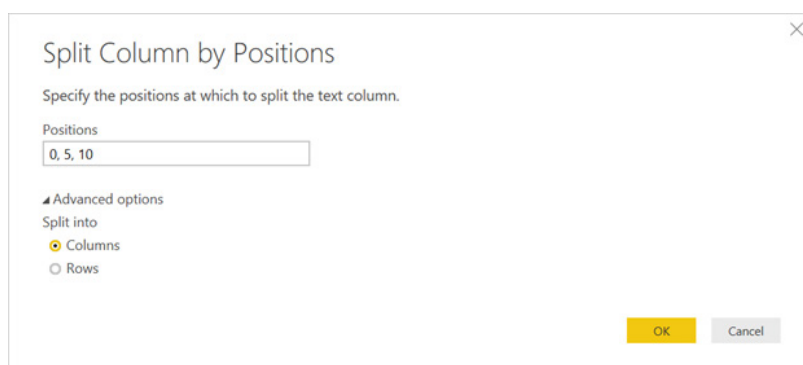


## New Transformation: Split column by positions

A new data transformation was added in this update allowing you to split text columns at specific positions within a given text value. This new transformation can be found under the 'Split Column' menu in the Home tab of the Power Query Editor Ribbon.

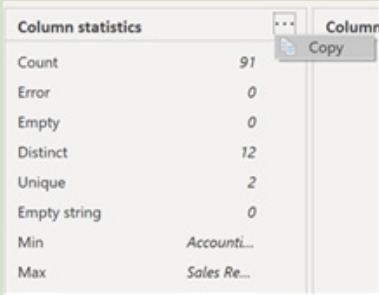


You can specify a comma-separated list of positions to split at. Power Query also tries to detect and make recommendations for those positions based on data in preview rows within the Power Query Editor. You can also customise whether to output text parts as new columns or rows.



## Copy to clipboard from data profiling

With this release, it's now possible to export data profiling information by copying it to the clipboard from the Power Query Editor. The new Copy option is available from all Data Profiling surfaces as described below:

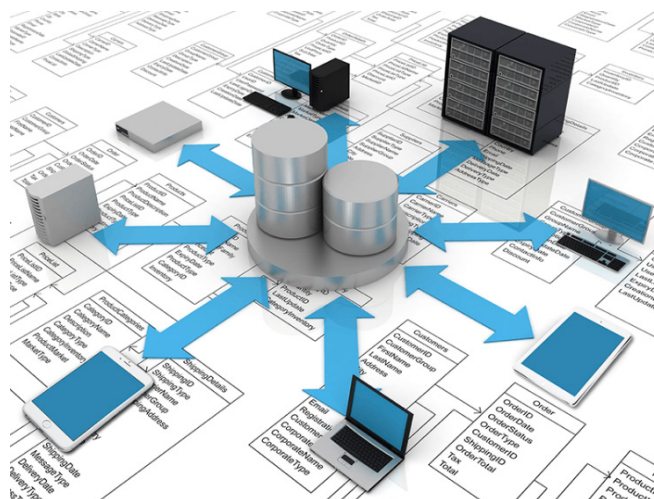
Data Profiling Surface	Screenshot	Data exported to clipboard																								
Column Quality Summary		<table border="1"> <tr> <td>Valid</td> <td>91</td> <td>100%</td> </tr> <tr> <td>Error</td> <td>0</td> <td>0%</td> </tr> <tr> <td>Empty</td> <td>0</td> <td>0%</td> </tr> </table>	Valid	91	100%	Error	0	0%	Empty	0	0%															
Valid	91	100%																								
Error	0	0%																								
Empty	0	0%																								
Column Distribution Charts		<table border="1"> <tr> <td>Sales Representative</td> <td>17</td> </tr> <tr> <td>Owner</td> <td>17</td> </tr> <tr> <td>Order Administrator</td> <td>2</td> </tr> <tr> <td>Marketing Manager</td> <td>12</td> </tr> <tr> <td>Accounting Manager</td> <td>10</td> </tr> <tr> <td>Sales Agent</td> <td>5</td> </tr> <tr> <td>Sales Associate</td> <td>7</td> </tr> <tr> <td>Sales Manager</td> <td>11</td> </tr> <tr> <td>Marketing Assistant</td> <td>6</td> </tr> <tr> <td>Assistant Sales Agent</td> <td>2</td> </tr> <tr> <td>Assistant Sales Representative</td> <td>1</td> </tr> <tr> <td>Owner/Marketing Assistant</td> <td>1</td> </tr> </table>	Sales Representative	17	Owner	17	Order Administrator	2	Marketing Manager	12	Accounting Manager	10	Sales Agent	5	Sales Associate	7	Sales Manager	11	Marketing Assistant	6	Assistant Sales Agent	2	Assistant Sales Representative	1	Owner/Marketing Assistant	1
Sales Representative	17																									
Owner	17																									
Order Administrator	2																									
Marketing Manager	12																									
Accounting Manager	10																									
Sales Agent	5																									
Sales Associate	7																									
Sales Manager	11																									
Marketing Assistant	6																									
Assistant Sales Agent	2																									
Assistant Sales Representative	1																									
Owner/Marketing Assistant	1																									
Column Profile Details Pane – Column Statistics		<table border="1"> <tr> <td>Count</td> <td>91</td> </tr> <tr> <td>Error</td> <td>0</td> </tr> <tr> <td>Empty</td> <td>0</td> </tr> <tr> <td>Distinct</td> <td>12</td> </tr> <tr> <td>Unique</td> <td>2</td> </tr> <tr> <td>Empty string</td> <td>0</td> </tr> <tr> <td>Min</td> <td>Accounting Manager</td> </tr> <tr> <td>Max</td> <td>Sales Re...</td> </tr> </table>	Count	91	Error	0	Empty	0	Distinct	12	Unique	2	Empty string	0	Min	Accounting Manager	Max	Sales Re...								
Count	91																									
Error	0																									
Empty	0																									
Distinct	12																									
Unique	2																									
Empty string	0																									
Min	Accounting Manager																									
Max	Sales Re...																									
Column Profile Details Pane – Column Statistics		<table border="1"> <tr> <td>Sales Representative</td> <td>17</td> </tr> <tr> <td>Owner</td> <td>17</td> </tr> <tr> <td>Order Administrator</td> <td>2</td> </tr> <tr> <td>Marketing Manager</td> <td>12</td> </tr> <tr> <td>Accounting Manager</td> <td>10</td> </tr> <tr> <td>Sales Agent</td> <td>5</td> </tr> <tr> <td>Sales Associate</td> <td>7</td> </tr> <tr> <td>Sales Manager</td> <td>11</td> </tr> <tr> <td>Marketing Assistant</td> <td>6</td> </tr> <tr> <td>Assistant Sales Agent</td> <td>2</td> </tr> <tr> <td>Assistant Sales Representative</td> <td>1</td> </tr> <tr> <td>Owner/Marketing Assistant</td> <td>1</td> </tr> </table>	Sales Representative	17	Owner	17	Order Administrator	2	Marketing Manager	12	Accounting Manager	10	Sales Agent	5	Sales Associate	7	Sales Manager	11	Marketing Assistant	6	Assistant Sales Agent	2	Assistant Sales Representative	1	Owner/Marketing Assistant	1
Sales Representative	17																									
Owner	17																									
Order Administrator	2																									
Marketing Manager	12																									
Accounting Manager	10																									
Sales Agent	5																									
Sales Associate	7																									
Sales Manager	11																									
Marketing Assistant	6																									
Assistant Sales Agent	2																									
Assistant Sales Representative	1																									
Owner/Marketing Assistant	1																									

More soon, we're sure!



# The A to Z of Excel Functions: DGET

This function extracts a single value from a column of a list or database that matches conditions that you specify.



The **DGET** function employs the following syntax to operate:

**DGET(database, field, criteria)**

The **DGET** function has the following arguments:

- **database:** this is required and represents the range of cells that makes up the list or **database**. A **database** is a list of related data in which rows of related information are records, and columns of data are fields. The first row of the list contains labels for each column
- **field:** this is also required. This indicates which column is used in the function. Enter the column label enclosed between double quotation marks, such as "Age" or "Yield," or a number (without quotation marks) that represents the position of the column within the list: 1 for the first column, 2 for the second column, and so on
- **criteria:** this is required too. This is the range of cells that contains the conditions that you specify. You can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label in which you specify a condition for the column.

It should be further noted that:

- if no record matches the **criteria**, **DGET** returns the **#VALUE!** error value
- if more than one record matches the **criteria**, **DGET** returns the **#NUM!** error value
- you can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label for specifying the condition. For example, if the range **G1:G2** contains the column label **Income** in **G1** and the amount **\$10,000** in **G2**, you could define the range as **MatchIncome** and use that name as the **criteria** argument in the **database** functions
- although the **criteria** range can be located anywhere on the worksheet, do not place the **criteria** range below the list. If you add more information to the list, the new information is added to the first row below the list. If the row below the list is not blank, Excel cannot add the new information
- make sure that the **criteria** range does not overlap the list
- to perform an operation on an entire column in a **database**, enter a blank line below the column labels in the **criteria** range.

Please see our example below:

	A	B	C	D	E
1	Tree	Height	Height		
2	Apple	>3	<=5		
3	Pear	>4			
4					
5	Tree	Height	Age	Yield	Profit
6	Apple	6	20	14	105
7	Apple	4	12	10	96
8	Cherry	4.2	14	9	105
9	Apple	5.7	15	10	75
10	Pear	3	8	8	16.8
11	Apple	2.7	9	6	45
12					
13	Formula	Description	Result		
14	=DGET(A5:E11,"Yield",A1:A3)	Returns the <b>#NUM!</b> error value because more than one record meets the criteria (i.e. any apple or pear tree).	<b>#NUM!</b>		
15	=DGET(A5:E11,"Yield",A1:C3)	Returns 10 (the yield of the apple tree in row 7) because that is the only record that meets the conditions in cells A1:C3.	10		
16					

## The A to Z of Excel Functions: DISC

The discount rate is the interest rate charged on loans and also refers to the interest rate used in discounted cash flow analysis to determine the present value of future cash flows. This function returns the discount rate for a security.

Discount Rate	6%	7%	8%	9%	10%	11%	12%	13%	14%
Year									
1	\$0.94	\$0.93	\$0.93	\$0.92	\$0.91	\$0.90	\$0.89	\$0.88	\$0.88
2	\$0.89	\$0.87	\$0.86	\$0.84	\$0.83	\$0.81	\$0.80	\$0.78	\$0.77
3	\$0.84	\$0.82	\$0.79	\$0.77	\$0.75	\$0.73	\$0.71	\$0.69	\$0.67
4	\$0.79	\$0.76	\$0.74	\$0.71	\$0.68	\$0.66	\$0.64	\$0.61	\$0.59
5	\$0.75	\$0.71	\$0.68	\$0.65	\$0.62	\$0.59	\$0.57	\$0.54	\$0.52
6	\$0.70	\$0.67	\$0.63	\$0.60	\$0.56	\$0.53	\$0.51	\$0.48	\$0.46
7	\$0.67	\$0.62	\$0.58	\$0.55	\$0.51	\$0.48	\$0.45	\$0.43	\$0.40
8	\$0.63	\$0.58	\$0.54	\$0.50	\$0.47	\$0.43	\$0.40	\$0.38	\$0.35
9	\$0.59	\$0.54	\$0.50	\$0.46	\$0.42	\$0.14	\$0.36	\$0.33	\$0.31

The **DISC** function employs the following syntax to operate:

**DISC(settlement, maturity, pr, redemption, [basis])**

The **DISC** function has the following arguments:

- **settlement**: this is required and represents the security's settlement date. The security settlement date is the date after the issue date when the security is traded to the buyer
- **maturity**: this is also required. This is the security's maturity date. The maturity date is the date when the security expires
- **pr**: again required. This denotes the security's price per \$100 face value
- **redemption**: required. This is the security's redemption value per \$100 face value
- **basis**: this argument is optional. This is the type of day count basis to use.

• 0 or omitted	US (NASD) 30 / 360
1	Actual / actual
2	Actual / 360
3	Actual / 365
4	European 30 / 360

It should be further noted that:

- Microsoft Excel stores dates as sequential serial numbers so they can be used in calculations. By default, January 1, 1900 is serial number 1, and January 1, 2008 is serial number 39448 because it is 39,448 days after January 1, 1900
- the settlement date is the date a buyer purchases a coupon, such as a bond. The maturity date is the date when a coupon expires. For example, suppose a 30-year bond is issued on January 1, 2008, and is purchased by a buyer six months later. The issue date would be January 1, 2008, the settlement date would be July 1, 2008, and the maturity date would be January 1, 2038, 30 years after the January 1, 2008, issue date
- **settlement**, **maturity** and **basis** are truncated to integers
- if **settlement** or **maturity** is not a valid serial date number, **DISC** returns the #VALUE! error value
- if **pr** ≤ 0 or if **redemption** ≤ 0, **DISC** returns the #NUM! error value
- if **basis** < 0 or if **basis** > 4, **DISC** returns the #NUM! error value
- if **settlement** ≥ **maturity**, **DISC** returns the #NUM! error value
- **DISC** is calculated as follows:

$$DISC = \frac{\text{redemption} - \text{par}}{\text{redemption}} \times \frac{B}{DSM}$$

where:

- **B** = number of days in a year, depending on the year **basis**
- **DSM** = number of days between **settlement** and **maturity**.

ase see the example below:

	A	B	C
1	Data	Description	
2	25/01/2020	Settlement date	
3	15-Jun-21	Maturity date	
4	98.375	Price	
5	100	Redemption value	
6	1	Actual / actual basis	
7			
8			
9	Formula	Description	Result
10	<b>=DISC(A2,A3,A4,A5,A6)</b>	The bond discount rate, for a bond with the above terms.	<b>1.171%</b>
11			

More Excel Functions next month...

## Upcoming SumProduct Training Courses

Location	Course	Date	Duration
Sydney	Excel Tips and Tricks	4 Nov 2019	1 Day
Sydney	Financial Modelling	5 - 6 Nov 2019	2 Days
Sydney	Power Pivot, Power Query and Power BI	11 - 13 Nov 2019	3 Days
Melbourne	Excel Tips and Tricks	6 Nov 2019	1 Day
Melbourne	Financial Modelling	7 - 8 Nov 2019	2 Days
Melbourne	Power Pivot, Power Query and Power BI	9 - 11 Dec 2019	3 Days
Sydney	Power Pivot, Power Query and Power BI	9 - 11 Dec 2019	3 Days
Sydney	Excel Tips and Tricks	16 Dec 2019	1 Day
Sydney	Financial Modelling	17 - 18 Dec 2019	2 Days
Melbourne	Excel Tips and Tricks	13 Jan 2020	1 Day
Melbourne	Financial Modelling	14 - 15 Jan 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	15 - 17 Jan 2020	3 Days
Sydney	Excel Tips and Tricks	27 Jan 2020	1 Day
Sydney	Financial Modelling	28 - 29 Jan 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	17 - 19 Feb 2020	3 Days
Sydney	Excel Tips and Tricks	2 Mar 2020	1 Day
Sydney	Financial Modelling	3 - 4 Mar 2020	2 Days
Melbourne	Power Pivot, Power Query and Power BI	9 - 11 Mar 2020	3 Days

## Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month, we thought we would take control of the function keys:

Keystroke	What it does
CTRL + F1	Show / hide Ribbon
CTRL + F2	Print
CTRL + F3	Open Name Manager
CTRL + F4	Close Window
CTRL + F5	Restore Window
CTRL + F6	Next Window / workbook
CTRL + F7	Move Window
CTRL + F8	Size Window
CTRL + F9	Minimise Window
CTRL + F10	Toggle Maximised / Restored
CTRL + F11	Insert new Macro Sheet
CTRL + F12	Open

There are over 540 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file a [www.sumproduct.com/thought/keyboard-shortcuts](http://www.sumproduct.com/thought/keyboard-shortcuts). Also, check out our new daily **Excel Tip of the Day** feature on the [www.sumproduct.com](http://www.sumproduct.com) homepage.

## Our Services

We have undertaken a vast array of assignments over the years, including:

- **Business planning**
- **Building three-way integrated financial statement projections**
- **Independent expert reviews**
- **Key driver analysis**
- **Model reviews / audits for internal and external purposes**
- **M&A work**
- **Model scoping**
- **Power BI, Power Query & Power Pivot**
- **Project finance**
- **Real options analysis**
- **Refinancing / restructuring**
- **Strategic modelling**
- **Valuations**
- **Working capital management**

If you require modelling assistance of any kind, please do not hesitate to contact us at [contact@sumproduct.com](mailto:contact@sumproduct.com).

## Link to Others

These newsletters are not intended to be closely guarded secrets. Please feel free to forward this newsletter to anyone you think might be interested in converting to "the SumProduct way".

If you have received a forwarded newsletter and would like to receive future editions automatically, please subscribe by completing our newsletter registration process found at the foot of any [www.sumproduct.com](http://www.sumproduct.com) web page.

## Any Questions?

If you have any tips, comments or queries for future newsletters, we'd be delighted to hear from you. Please drop us a line at [newsletter@sumproduct.com](mailto:newsletter@sumproduct.com).

## Training

SumProduct offers a wide range of training courses, aimed at finance professionals and budding Excel experts. Courses include Excel Tricks & Tips, Financial Modelling 101, Introduction to Forecasting and M&A Modelling.

Check out our more popular courses in our training brochure:



Drop us a line at [training@sumproduct.com](mailto:training@sumproduct.com) for a copy of the brochure or download it directly from <http://www.sumproduct.com/training>.