SumProduct

NEWSLETTER #52 - March 2017

www.sumproduct.com | www.sumproduct.com/thought

We're taking a March on Power Query

this month as we look at Power Query in March. The first updates for 2017 have now been released and we add some pointers to boot.

March also sees us out and about in Queensland and New South Wales (well, we have to take in some Southern summer sun before it fades) with regional events in Newcastle, Townsville and Wollongong.

With the usual Power Query Pointers, the A to Z of Excel Functions and possibly the shortest Keyboard Shortcuts session ever, our monthly newsletter hopes to keep you informed.

Until next month.

Liam Bastick, Managing Director, SumProduct

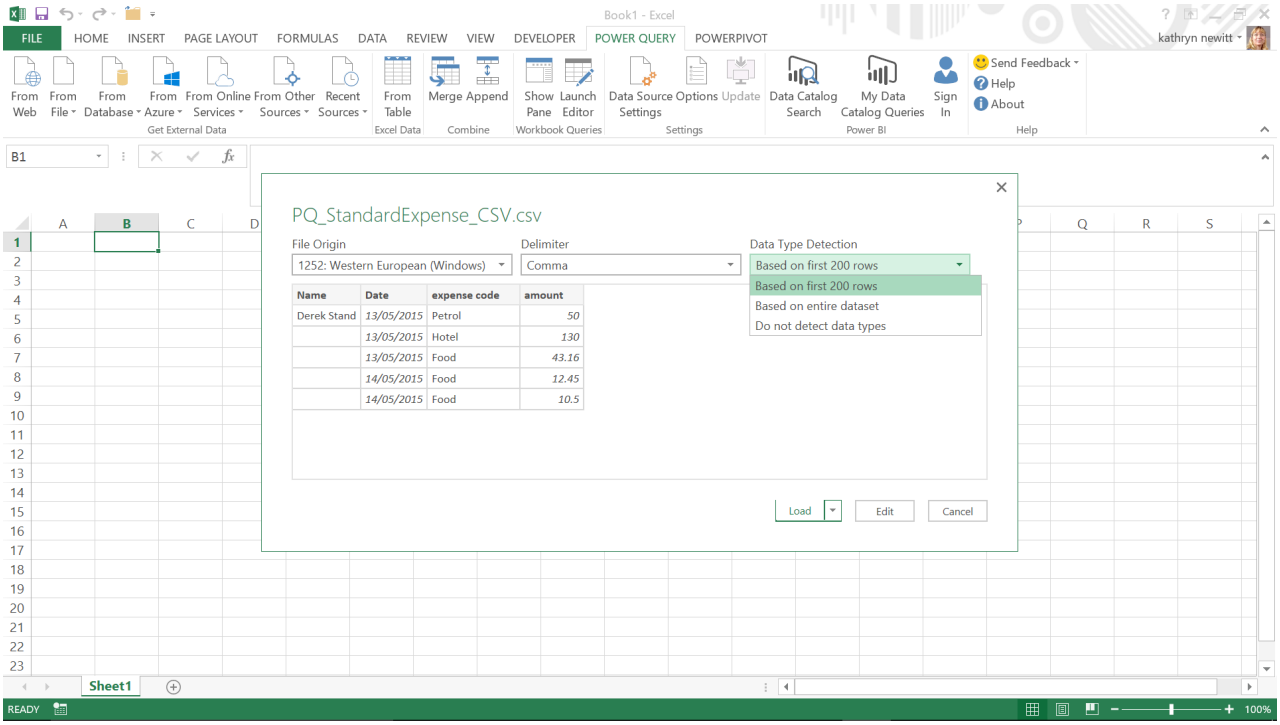
Power Query Pointers

Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Excel 2016) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday.

The best way to get to grips with a new tool like Power Query is to start with a simple task. Excel users may often need to take data from CSV (comma separated values) files and transform it ready for analysis. Power Query has been designed to assist with this, so let's see how easy it can be.

Starting with a new workbook, we open Power Query – the screenshots shown here are from Excel 2013, where Power Query has its own Ribbon, as it does for Excel 2010; for Excel 2016, Power Query is on the Data tab in the 'Get & Transform' group. As shown in the last blog, there are a variety of possible sources to extract from in the Get External Data section of the Ribbon.

For this example, we are using the 'From File' option and browsing to the location of a simple expense CSV file. The image below appears first (this has been added in a recent update, so if you go straight to the Power Query Editor, you are missing the latest update – see the Installing and Updating blog for guidance on this). This intermediate screen has been added to give the user the option to decide whether Power Query should make some assumptions about data types. This will become clearer when the steps are generated so take the default option to allow Power Query to detect data types 'Based on first 200 rows' (more than enough for out short example). We could just choose to load from this point, but instead, choose to Edit or transform the data:



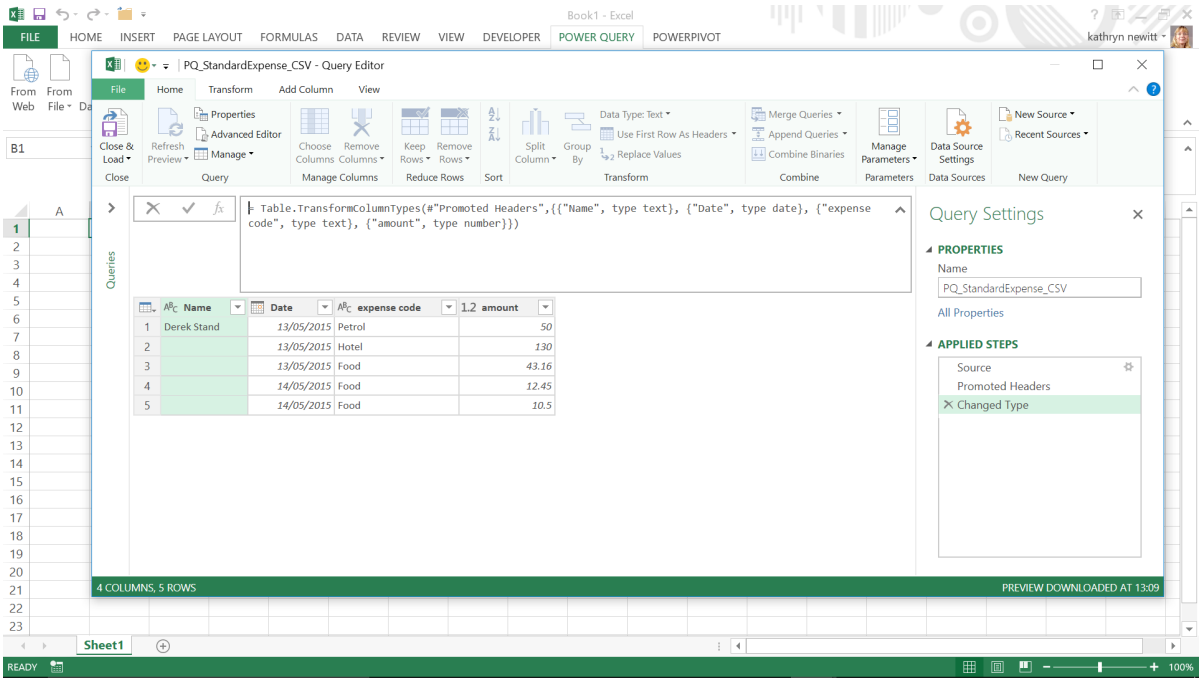
The Power Query Editor screen appears, which shows the table of data as it will be uploaded and the steps that Power Query has taken automatically. The source has been identified, the header column names have been assumed, and as shown in the screen below, the 'Change Type' step sets the detected type for each column. The syntax for this is:

```
=Table.TransformColumnTypes(#"Promoted Headers",{{"Name", type text}, {"Date", type date}, {"expense code", type text}, {"amount", type number}})
```

This is a simple example of M code, which will be explored in more detail in future blogs. It is essentially a list of column names and their assigned data types based on the data that Power Query has analysed:

- Name is type text
- Date is type date
- expense code is type text
- amount is number.

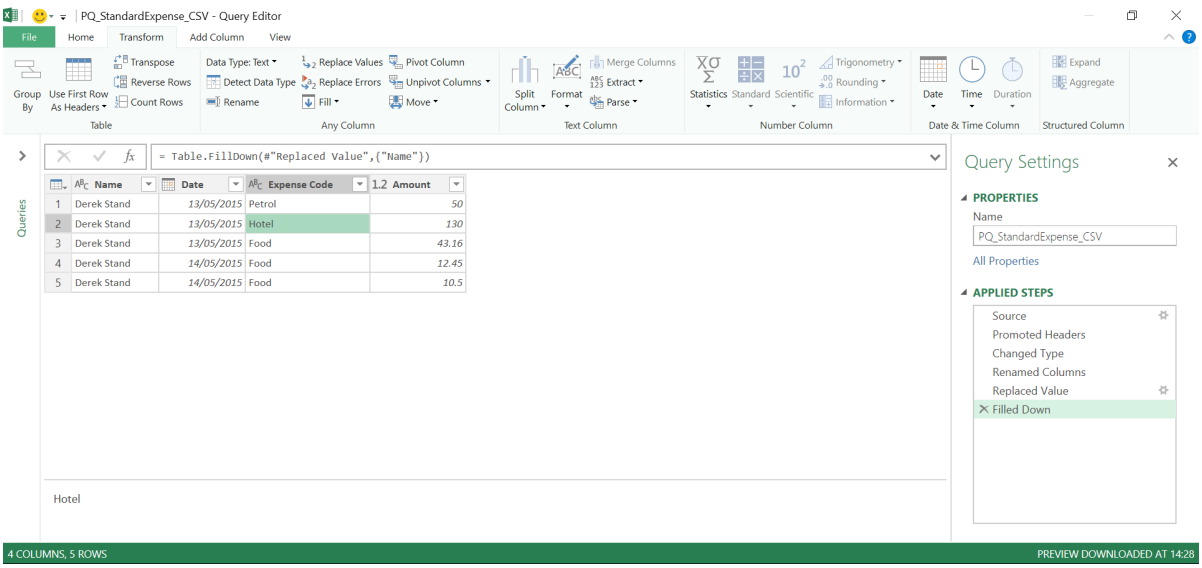
In this case, all the assumptions look good so we can accept Power Query's automatic assignment of data types.



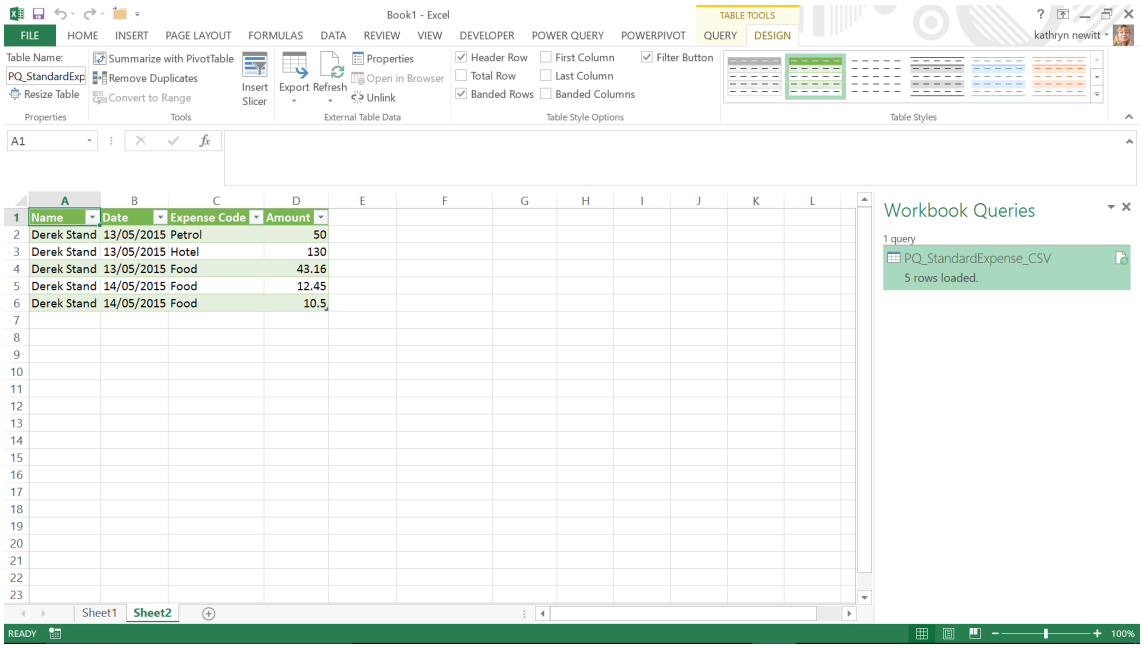
We can then make any changes to the data, such as changing column names or removing columns. Selecting a column and viewing the 'Transform' tab reveals buttons for many common actions like this. If we choose to rename **expense code** and **amount** to **Expense Code** and **Amount**, then Power Query not only creates a step, but knows to combine both renames into one step:

```
= Table.RenameColumns(#"Changed Type",{{"expense code", "Expense Code"}, {"amount", "Amount"}})
```

A more complex step is to make the **Name** data appear on each row. In order to use the 'Fill Down' button, the **Name** cells that are to be populated must be set to null and they are currently blank. Therefore, we need to replace the blanks with null and then fill down:



The steps are generated in the right-hand pane, and we are ready to load to Excel using the 'Close and Load' button on the 'File' tab. The uploaded table is shown below, and the Workbook query window displays the query that generated the data, so that it can be updated and / or refreshed as required. The 'TABLE TOOLS' tab opens automatically ready for use in the Excel workbook:



First Updates for Get & Transform / Power Query in 2017

After a quiet time of late, Microsoft has finally got busy again. Excel 2016's Get & Transform / Excel 2010's or Excel 2013's Power Query for Excel add-in have been updated.

In total, six new data transformation and connectivity features have been announced. For Excel 2016, they are available as part of an Office 365 subscription. For earlier versions, you simply need to download the latest version of the add-in.

These updates include the following new or improved data connectivity and transformation features:

- New OLE DB connector
- Enhanced "Combine Binaries" experience when importing from any folder
- Maximize / Restore buttons in the Navigator and Query Dependencies dialogs
- Support for percentage data type
- Improved "Function Authoring" experience
- Improved performance for OData connector.

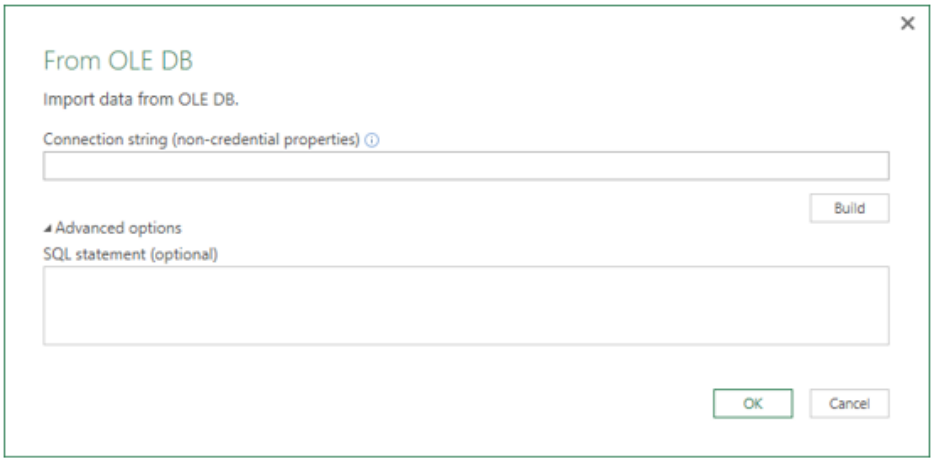
Let's go through each in turn.

New OLE DB connector

In this update, Microsoft has enabled connectivity to OLE DB drivers via this new connector. In addition to the wide range of out-of-the-box sources supported, OLE DB greatly increases the number of sources that users can now import from by using ***Get & Transform*** capabilities in Excel.

The new OLE DB connector can be found under **Data > New Query > From Other Sources > From OLE DB**.

The connector dialog allows users to specify a Connection String and, optionally, an SQL statement to execute. If no SQL statement were specified, users will be taken into the Navigator dialog, where they can browse and select one or multiple tables available via the selected OLE DB driver.



Enhanced "Combine Binaries" experience when importing from any folder

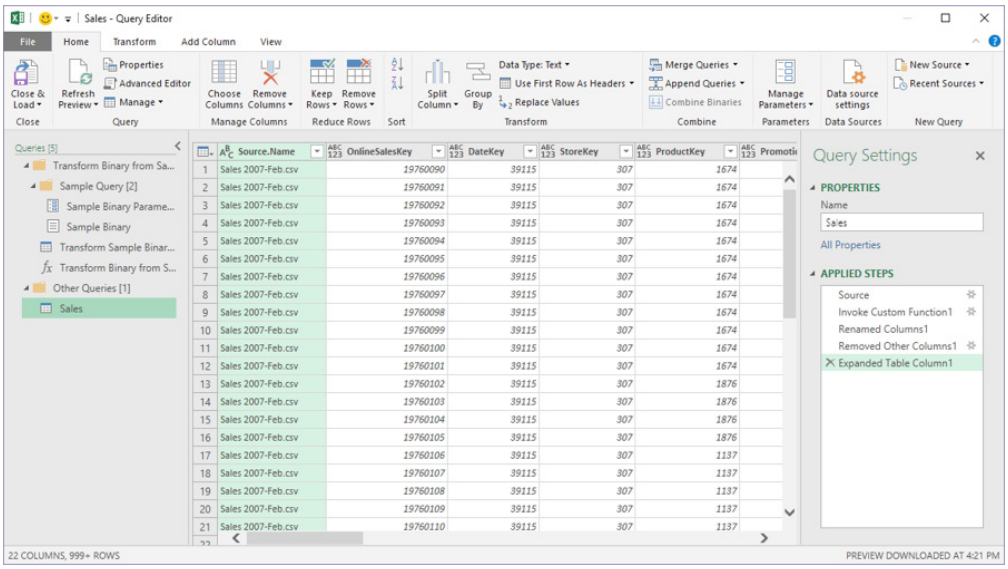
One of the most popular scenarios in Excel consists of leveraging one of the folder-like connectors (such as Folder, SharePoint folder, etc.) to combine multiple files with the same schema into a single logical table.

Before this release, users could combine Text or CSV files only. Combining would not work for any other supported file formats (such as Excel Workbooks, JSON files) and it would not account for transformations required on each file before combining them into a single table (e.g. removing the first row with header values).

With this release, Microsoft has enhanced the 'Combine Binaries' experience when importing from any folder so that:

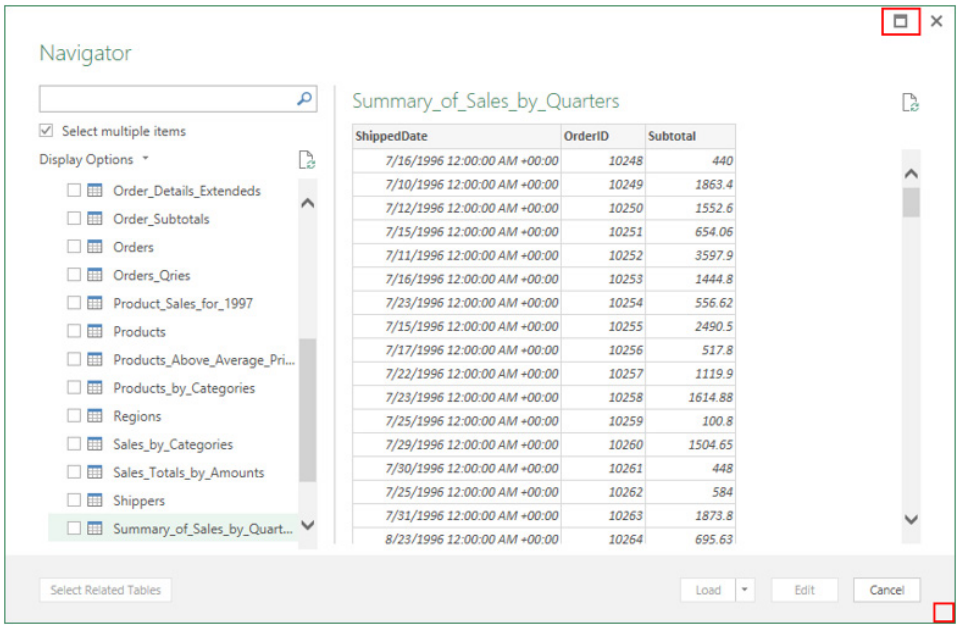
- Excel analyses the input files from the Folder query and detects the right file format to use (*e.g.* Text or Excel Workbook)
- Users can select a specific object from the list (such as a spreadsheet name) to use for data combination
- Excel automatically creates the following entities:
  - o An example query that performs all required transformation steps in a single file
  - o A function query that parameterises the file input to the exemplar query created in the previous step
  - o Excel then applies the created function query on each file from the original Folder query and expands the resulting data extraction as top-level columns.

With this new approach, users can easily combine all binaries within a folder if they have a homogeneous file type and column structure. Users can also easily apply additional transformations by modifying the "exemplar query" without having to worry about any additional function invocation steps, as they are automatically generated for them.



Maximize / Restore buttons in the Navigator and Query Dependencies dialogs

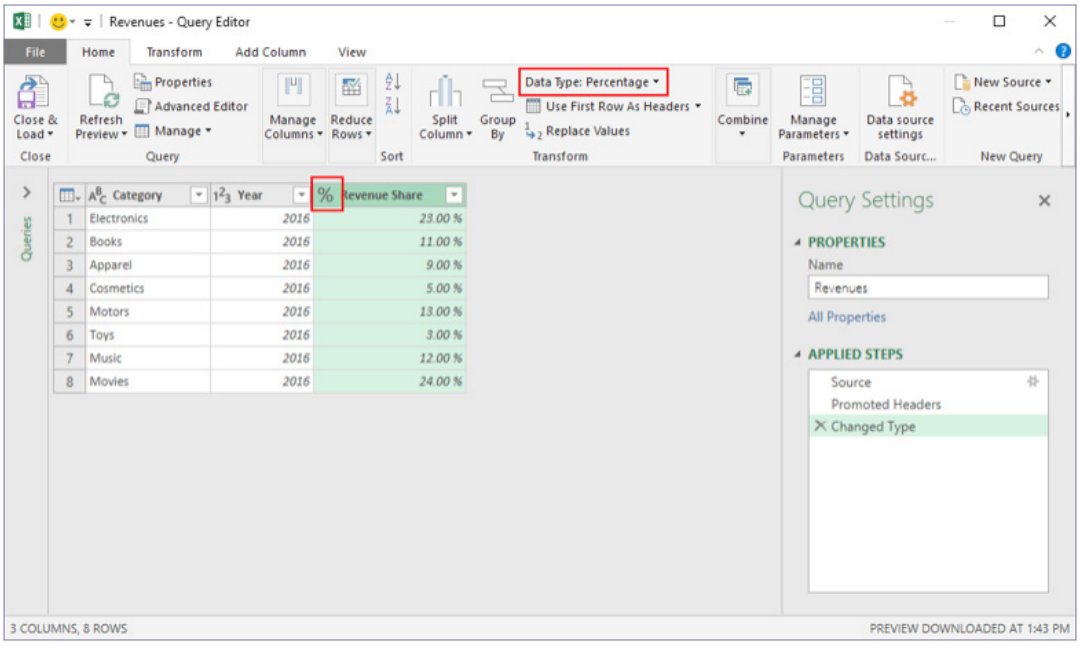
The 'Navigator and Query Dependencies' dialog (activated from 'Query Editor') support window resizing by dragging the bottom-right edges of the dialog. In this release, the updates have made it possible to maximise / restore these dialogs by exposing 'Maximize' and 'Restore' icons in the top-right corner of the dialog boxes.



Support for percentage data type

With this update, Microsoft has also added support for percentage data types, so they can easily be used in arithmetical operations for Get & Transform scenarios. An input value such as "5%" will be automatically recognized as a percentage value and converted to a two-digit precision decimal number (*i.e.* 0.05), which can then be used in arithmetical operations within a spreadsheet, the Query Editor and / or the Data Model.

Besides automatic type recognition from non-structured sources (such as Text, CSV or HTML), users can also convert any value to percentage using the Change Type options in the Query Editor. You can do this on the 'Query Editor Home' tab, on the 'Transform' tab, by clicking **Data Type > Percentage**, or right-clicking a column and then selecting **Change Type > Percentage**.



Improved "Function Authoring" experience

It's been made easier to update function definitions without the need to maintain the underlying M code (yay).

To illustrate this, create a function based upon another query using the 'Create Function' command. This may be performed by right-clicking the 'Queries' pane inside 'Query Editor'. Once this is done, a link will be created between the original query and the newly generated function. This way, when the user modifies the original query steps, the linked function will be automatically updated as well.

When using 'Query Parameters', creating a function out of a query will allow users to use 'Function Inputs' to replace parameter values in the generalised function query.

Improved performance for OData connector

The latest update has added support for pushing 'Expand Record' operations to be performed in the underlying OData service. This will result in improved performance when expanding records from an OData feed.

We will continue to bring the latest updates as and when we receive them.

The A to Z of Excel Functions: BAHTTEXT

Every now and then we unveil a pearl of a function, one that will rock your world and make you wonder how you ever lived without it. Today's function is probably not going to be on that list for those outside of Thailand.

This function converts a number to Thai text and adds a suffix of "Baht". You can change the Baht format to a different style in the Excel desktop application by using Regional and Language Options (Windows Start menu, Control Panel).

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

BAHTTEXT(number)

The **BAHTTEXT** function has the following argument only:

- **number:** this is required and represents a number you want to convert to text, or a reference to a cell containing a number, or a formula that evaluates to a number.

	A	B	C	D
1	Value	Result	Formula	Comments
2	1234	หนึ่งพันสองร้อยสามสิบสี่บาทถ้วน	=BAHTTEXT(A2)	Converts the number to text (one thousand, two hundred and thirty-four Baht in Thai text)
3	Simpson	#VALUE!	=BAHTTEXT(A3)	Worth a shot
4				

I have to be honest, I am not quite sure why Microsoft has singled out Thai for this special treatment – especially as there is no equivalent to do this in English.

For the record (and probably of more relevance than the above for many), back in **Newsletter 33** (August 2015), we did actually enlighten readers how VBA code could be used to convert numbers to text by using the user defined function ***SpellNumber***. To create this function, open the Visual Basic Editor in Excel (**ALT + F11**) and paste in the following code:

```
Option Explicit
'Main Function
Function SpellNumber(ByVal MyNumber)
    Dim Dollars, Cents, Temp
    Dim DecimalPlace, Count
    ReDim Place(9) As String
    Place(2) = " Thousand "
    Place(3) = " Million "
    Place(4) = " Billion "
    Place(5) = " Trillion "
    ' String representation of amount.
    MyNumber = Trim(Str(MyNumber))
    ' Position of decimal place 0 if none.
    DecimalPlace = InStr(MyNumber, ".")
    ' Convert cents and set MyNumber to dollar amount.
    If DecimalPlace > 0 Then
        Cents = GetTens(Left(Mid(MyNumber, DecimalPlace + 1) & _
                "00", 2))
        MyNumber = Trim(Left(MyNumber, DecimalPlace - 1))
    End If
    Count = 1
    Do While MyNumber <> ""
        Temp = GetHundreds(Right(MyNumber, 3))
        If Temp <> "" Then Dollars = Temp & Place(Count) & Dollars
        If Len(MyNumber) > 3 Then
            MyNumber = Left(MyNumber, Len(MyNumber) - 3)
        Else
            MyNumber = ""
        End If
        Count = Count + 1
    Loop
    Select Case Dollars
        Case ""
            Dollars = "No Dollars"
        Case "One"
            Dollars = "One Dollar"
         Case Else
            Dollars = Dollars & " Dollars"
    End Select
    Select Case Cents
        Case ""
            Cents = " and No Cents"
        Case "One"
            Cents = " and One Cent"
              Case Else
            Cents = " and " & Cents & " Cents"
    End Select
    SpellNumber = Dollars & Cents
End Function

' Converts a number from 100-999 into text
Function GetHundreds(ByVal MyNumber)
    Dim Result As String
    If Val(MyNumber) = 0 Then Exit Function
    MyNumber = Right("000" & MyNumber, 3)
    ' Convert the hundreds place.
    If Mid(MyNumber, 1, 1) <> "0" Then
```

```
        Result = GetDigit(Mid(MyNumber, 1, 1)) & " Hundred "
    End If
    ' Convert the tens and ones place.
    If Mid(MyNumber, 2, 1) <> "0" Then
        Result = Result & GetTens(Mid(MyNumber, 2))
    Else
        Result = Result & GetDigit(Mid(MyNumber, 3))
    End If
    GetHundreds = Result
End Function

' Converts a number from 10 to 99 into text.
Function GetTens(TensText)
    Dim Result As String
    Result = ""           ' Null out the temporary function value.
    If Val(Left(TensText, 1)) = 1 Then   ' If value between 10-19...
        Select Case Val(TensText)
            Case 10: Result = "Ten"
            Case 11: Result = "Eleven"
            Case 12: Result = "Twelve"
            Case 13: Result = "Thirteen"
            Case 14: Result = "Fourteen"
            Case 15: Result = "Fifteen"
            Case 16: Result = "Sixteen"
            Case 17: Result = "Seventeen"
            Case 18: Result = "Eighteen"
            Case 19: Result = "Nineteen"
            Case Else
        End Select
    Else                                 ' If value between 20-99...
        Select Case Val(Left(TensText, 1))
            Case 2: Result = "Twenty "
            Case 3: Result = "Thirty "
            Case 4: Result = "Forty "
            Case 5: Result = "Fifty "
            Case 6: Result = "Sixty "
            Case 7: Result = "Seventy "
            Case 8: Result = "Eighty "
            Case 9: Result = "Ninety "
            Case Else
        End Select
        Result = Result & GetDigit _
            (Right(TensText, 1))  ' Retrieve ones place.
    End If
    GetTens = Result
End Function

' Converts a number from 1 to 9 into text.
Function GetDigit(Digit)
    Select Case Val(Digit)
        Case 1: GetDigit = "One"
        Case 2: GetDigit = "Two"
        Case 3: GetDigit = "Three"
        Case 4: GetDigit = "Four"
        Case 5: GetDigit = "Five"
        Case 6: GetDigit = "Six"
        Case 7: GetDigit = "Seven"
        Case 8: GetDigit = "Eight"
        Case 9: GetDigit = "Nine"
        Case Else: GetDigit = ""
    End Select
End Function
```

The A to Z of Excel Functions: BASE

This is a more useful function. This function converts a **number** into a text representation with the given radix (base). Clear as mud if you speak gobbledygook.

What does it mean in English? **BASE** converts a **number** in decimal (base 10) to a different base (which is known as the **radix**). Let me provide an example. Take the number 37. In base 3, this can be represented as:

$$(1 \times 33) + (1 \times 32) + (0 \times 31) + (1 \times 30)$$

$$= (1 \times 27) + (1 \times 9) + (0 \times 3) + (1 \times 1)$$

$$= 27 + 9 + 0 + 1$$

So in base 3 this is the text representation **1101**. If we needed this representation to be of minimum length (**min_length**) 8, then

$$27 = (0 \times 37) + (0 \times 36) + (0 \times 35) + (0 \times 34) + (1 \times 33) + (1 \times 32) + (0 \times 31) + (1 \times 30)$$

$$= 00001101.$$

Makes sense now?

The **BASE** function has only been around since Excel 2013 so it will not work in earlier versions of Excel. It employs the following syntax to operate:

BASE(number, radix, [min_length])

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

- **number:** this is required and represents the number that you want to convert. This must be an integer greater than or equal to 0 and less than 2^{53}
- **radix:** this is also required and represents the base radix that you want to convert the number into. This must be an integer greater than or equal to 2 and less than or equal to 36 (*why 36??*)
- **min_length:** this is optional. This represents minimum length of the returned string and must be an integer greater than or equal to zero if specified.

It should be further noted that:

- If **number**, **radix**, or **min_length** are outside the minimum or maximum constraints, **BASE** returns the *#NUM!* error value
- If **number** is a non-numeric value, **BASE** returns the *#VALUE!* error value
- Any non-integer number entered as an argument is truncated to an integer
- If the **min_length** argument is included, leading zeros are added to the result if the result would otherwise be shorter than the minimum length specified. For example, **BASE(16,2)** returns 10000, but **BASE(16,2,8)** returns 00010000
- The maximum value of the **min_length** argument is 255.

Please see my example below:

	A	B	C	D	E
1	Number	Radix	Min Length	Text Representation	Formula
2	37	3		1101	=BASE(A2,B2)
3	37	3		1101	=BASE(A3,B3,C3)
4	37	3	8	00001101	=BASE(A4,B4,C4)
5	3	37		#NUM!	=BASE(A5,B5,C5)
6	4.2	5.1	3.7	004	=BASE(A6,B6,C6)
7	dog	cat		#VALUE!	=BASE(A7,B7,C7)
8					

Errata

We hate it when we have to come clean. The Quatily Cotnrol department is aptly named. We have two errata to 'fess up to. Well done to our eagle-eyed readers: last month's newsletter was actually for **February** 2017 and Melbourne director Liam Bastick has actually been appointed an Excel MVP by Microsoft for the **sixth** time not the fourth. We look forward to your comments for this month's newsletter…

Remember, every time you admit to a mistake the errorists win.

Australian Road Trip

For our readers Down Under, aside from our usual training activities (please check out our training calendar on the back page), SumProduct is off on the road. In conjunction with CPA Australia, we're presenting two Excel Masterclasses, ***Laying Out a Financial Model*** and ***Strategic Budgeting and Forecasting***, this month in the following locations:

- **Newcastle:** Wednesday 15 March
- **Townsville:** Monday 20 March
- **Wollongong:** Thursday 23 March

To find out more and / or register, please visit the official website here.

And Then There Was One More: Meet Jacqueline Day

A criticism sometimes aimed at us is more is known about CIA personnel than SumProduct's so we thought we might try and put that right going forward. And what better way than to introduce our latest addition, Jacqueline (Jacqui) Day.

Picture provided by CIA

Jacqui joined us in late February. With a degree in Commerce, Jacqui is also a Member of the Institute of Chartered Accountants (CAANZ) and is about to complete her MBA. We're all feeling nervous about our own qualifications now…

Jacqui has worked as an accountant in a large accounting firm as well as in commerce. Jacqui commenced her accounting career with William Buck Chartered Accountants predominantly working with Small and Medium Enterprises (SMEs) in their Business Advisory Department. Consequently, Jacqui is highly specialised in various compliance tasks, including preparation and lodgement of business and individual tax returns, the preparation of financial statements for a variety of businesses and also lodgement of Business Activity Statements.

More recently, Jacqui was Chief Financial Officer to a well-known cosmetics and tanning products producer and distributor, making the successful leap from professional services to a commercial accounting role.

All this means Jacqui has a meticulous eye for detail and an excellent understanding of the needs and demands of a busy commercial enterprise. Originally based in South Australia, Jacqui will be working in Adelaide until July when she will join our Sydney office. Jacqui has always had a passion for financial analysis and business modelling and will be introducing herself to many of our clients, past and present, throughout Australia in the coming months.

Welcome aboard Jacqui – we are sure she will be sharing some of her own thoughts through this newsletter and our supporting blogs / news articles very shortly.

Microsoft Data Insights Summit

Microsoft has just announced the second running of its Data Insights Summit. Last year's event was extremely was a great success with over 1,200 people attending.

The Data Insights Summit is a user conference for business analysts, whether novice or highly experienced, designed to help them identify deeper insights, make better sense of their data and take action to transform their business. Attendees can meet directly with the Microsoft Power BI, SQL Server BI, Excel, PowerApps, Flow and Stream teams to help answer their most complex data questions as well as provide product feedback.

The event will take place June 12 – 13, 2017 at the Washington State Convention Center in Seattle, WA, USA. We're hoping we can get a visa…

For more information, check out Microsoft's website here.

Upcoming SumProduct Training Courses

Location	Course	Date	Duration
Toronto	Financial Modelling	6th - 7th Mar 2017	2 days
Toronto	Power Pivot, Power Query and Power BI	8th - 10th Mar 2017	3 days
New York	Financial Modelling	13th - 14th Mar 2017	2 days
New York	Power Pivot, Power Query and Power BI	15th - 17th Mar 2017	3 days
Sydney	Power Pivot, Power Query and Power BI	20th - 22nd Mar 2017	3 days
Melbourne	Financial Modelling	27th - 28th Mar 2017	2 days
Sydney	Visual Basic for Applications	4th Apr 2017	1 day
Sydney	Excel Tips & Tricks	10th Apr 2017	1 day
Sydney	Financial Modelling	20th - 21st Apr 2017	2 days
Melbourne	Excel Tips & Tricks	29th May 2017	1 day
Melbourne	Financial Modelling	30th - 31st May 2017	2 days
Melbourne	Introduction to Financial Forecasting	1st Jun 2017	1 day
Perth	Excel Tips & Tricks	7th Jun 2017	1 day
Perth	Financial Modelling	8th - 9th Jun 2017	2 days
Sydney	Excel Tips & Tricks	13rd Jun 2017	1 day
Sydney	Financial Modelling	14th Jun 2017	1 day
Melbourne	Power Pivot, Power Query and Power BI	19th - 21st Jun 2017	3 days
Brisbane	Excel Tips & Tricks	26th Jun 2017	1 day
Brisbane	Financial Modelling	27th - 28th Jun 2017	2 days
Melbourne	Excel Tips & Tricks	3rd Jul 2017	1 day
Melbourne	Financial Modelling	4th - 5th Jul 2017	2 days
Sydney	Visual Basic for Applications	10th - 10th Jul 2017	1 day
Sydney	Power Pivot, Power Query and Power BI	17th - 19th Jul 2017	3 days
Adelaide	Power Pivot, Power Query and Power BI	24th - 26th Jul 2017	3 days
Sydney	Excel Tips & Tricks	31st Jul 2017	1 day
Sydney	Financial Modelling	1st - 2nd Aug 2017	2 days

Perth	Power Pivot, Power Query and Power BI	7th - 9th Aug 2017	3 days
Adelaide	Excel Tips & Tricks	14th - 16th Aug 2017	3 days
Adelaide	Financial Modelling	15th Aug 2017	1 day
Melbourne	Mergers and Acquisitions	21st - 24th Aug 2017	4 days
Brisbane	Power Pivot, Power Query and Power BI	28th - 30th Aug 2017	3 days
Melbourne	Excel Tips & Tricks	4th Sep 2017	1 day
Melbourne	Financial Modelling	5th - 6th Sep 2017	2 days
Melbourne	Power Pivot, Power Query and Power BI	12nd - 14th Sep 2017	3 days
Sydney	Excel Tips & Tricks	25th Sep 2017	1 day
Sydney	Financial Modelling	26th - 27th Sep 2017	2 days
Sydney	Power Pivot, Power Query and Power BI	16th - 18th Oct 2017	3 days
Melbourne	Excel Tips & Tricks	23rd Oct 2017	1 day
Melbourne	Financial Modelling	24th - 25th Oct 2017	2 days
Sydney	Visual Basic for Applications	6th Nov 2017	1 day
Sydney	Excel Tips & Tricks	13rd Nov 2017	1 day
Sydney	Financial Modelling	14th - 15th Nov 2017	2 days
Melbourne	Power Pivot, Power Query and Power BI	4th - 6th Dec 2017	3 days
Melbourne	Excel Tips & Tricks	11st Dec 2017	1 day
Melbourne	Financial Modelling	12nd - 13rd Dec 2017	2 days

Referral Programme

Have you heard about the SumProduct Referral programme? If you successfully introduce us to a client we will pay you 10% of the net profits of the engagement won upon receipt of client payment. Sounds like a pretty good deal to us!

Terms and conditions apply. Referrer must be over 18 years of age, able to issue an invoice and not be an employee, director, agent or other related party of the client engaged. For further information, drop us a line at contact@sumproduct.com. We hope to hear from you!

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month we decided we'd continue going through the function keys – this time, with the **F7** function key (and there's not much to say this month):

Keystroke	What it does
F7	Check Spelling
CTRL + F7	Move Window

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

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 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.

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 for a copy of the brochure or download it directly from http://www.sumproduct.com/training.

Sydney Address: SumProduct Pty Ltd, Suite 52, Level 10, 88 Pitt Street, Sydney, NSW 2000
New York Address: SumProduct Pty Ltd, 48 Wall Street, New York, NY, USA 10005
London Address: SumProduct Pty Ltd, Office 7, 3537 Ludgate Hill, London, EC4M 7JN, UK
Melbourne Address: SumProduct Pty Ltd, Level 9, 440 Collins Street, Melbourne, VIC 3000
Registered Address: SumProduct Pty Ltd, Level 6, 468 St Kilda Road, Melbourne, VIC 3004

contact@sumproduct.com
www.sumproduct.com
+61 3 9020 2071