IPI Paul

Working with Java and Excel VBA

Name : TrapRange Java
Type : Java Jar and Excel VBA
Designer : Paul I Ighofose
Date Created : 19/11/2020
Date Updated : 23/11/2020
Optional : Microsoft Office 2013+, Java 8
Function : Extract tables from PDF file, format, cleanse and display in Excel Table

Introduction

PDF Finance Statements and Agreements of Balances have always had a residual effect on my colleagues in the NHS Finance Team. I tried various ways of pulling data from PDF invoices to speed up the validation process, but without the availability of Power BI I was left to use Word for the conversion process. After watching a Linkedin youtube video on the use of Power Query, I searched the internet for more information and built the 1st Excel VBA workbook that converted a PDF using Word and then Extracted tables from the resulting MHTML file using Power BI. I later found some R Script code and extended the VBA code to call modified R Scripts and pull in the resulting file tables using an ADODB connection.

Both those methods took around 5 to 10 minutes just to generate the base file from the PDF. I then found Java code that offered PDF table extraction and went with TrapRange from https://github.com/thoqbk/traprange. After making a few modifications to Tho's code and creating a new Excel macro driven file from the current R Script and Power Query one, I was able to witness the base file creation from the Java module in 47 to 52 seconds as opposed to 5 to 10 minutes. Eventually, after a lot of modifications the data was being displayed accurately in the Excel List Object in no more than 1 minute 18 seconds.

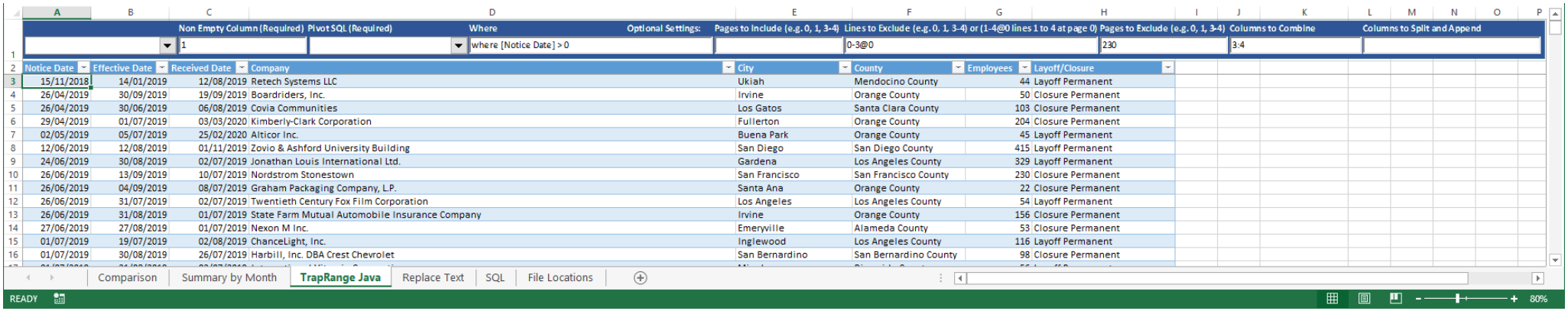
Package Structure

The modified Java module has been compiled to the file TrapRange.jar. And the Excel file is currently named TrapRange Java.xlsm. The VBA code has been written to work with a specific **Directory Structure**. A folder named csv must exist in the parent folder of the subfolder that the PDF file is in (i.e. if the PDF file is in C:\Temp\pdf then the folder C:\Temp\csv must already exist).

Operational Guide

- Save both TrapRange.jar and TrapRange Java.xlsm to a location on your computer.
- Ensure that you have the required **Directory Structure** before you start.
- It is always best to open the PDF file and look at its structure so as to be sure what variables to pass in as parameters in the parameter fields provided. You will need to know:
    - What page number (Java parameter, page 1 is 0) the table starts on and how many lines down on that page it is
    - What page number (Java parameter, page number -1) the table ends on and how many more lines of text are below the table on that page

- Be aware that Java indexes start from 0 and VBA indexes start from 1



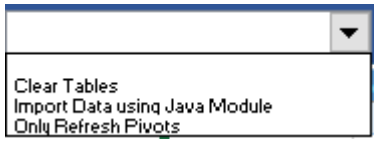
	Non Empty Column (Required)	Pivot SQL (Required)	Where	Optional Settings:	Pages to Include (e.g. 0, 1, 3-4)	Lines to Exclude (e.g. 0, 1, 3-4) or (1-4@0 lines 1 to 4 at page 0)	Pages to Exclude (e.g. 0, 1, 3-4)	Columns to Combine	Columns to Split and Append
1	1		where [Notice Date] > 0			0-3@0	230	3:4	

Notice Date	Effective Date	Received Date	Company	City	County	Employees	Layoff/Closure
15/11/2018	14/01/2019	12/08/2019	Retech Systems LLC	Ukiah	Mendocino County	44	Layoff Permanent
26/04/2019	30/09/2019	19/09/2019	Boardriders, Inc.	Irvine	Orange County	50	Closure Permanent
26/04/2019	30/06/2019	06/08/2019	Covia Communities	Los Gatos	Santa Clara County	103	Closure Permanent
29/04/2019	01/07/2019	03/03/2020	Kimberly-Clark Corporation	Fullerton	Orange County	204	Closure Permanent
02/05/2019	05/07/2019	25/02/2020	Alticor Inc.	Buena Park	Orange County	45	Layoff Permanent
12/06/2019	12/08/2019	01/11/2019	Zovio & Ashford University Building	San Diego	San Diego County	415	Layoff Permanent
24/06/2019	30/08/2019	02/07/2019	Jonathan Louis International Ltd.	Gardena	Los Angeles County	329	Layoff Permanent
26/06/2019	13/09/2019	10/07/2019	Nordstrom Stonestown	San Francisco	San Francisco County	230	Closure Permanent
26/06/2019	04/09/2019	08/07/2019	Graham Packaging Company, L.P.	Santa Ana	Orange County	22	Closure Permanent
26/06/2019	31/07/2019	02/07/2019	Twentieth Century Fox Film Corporation	Los Angeles	Los Angeles County	54	Layoff Permanent
26/06/2019	31/08/2019	01/07/2019	State Farm Mutual Automobile Insurance Company	Irvine	Orange County	156	Closure Permanent
27/06/2019	27/08/2019	01/07/2019	Nexon M Inc.	Emeryville	Alameda County	53	Closure Permanent
01/07/2019	19/07/2019	02/08/2019	ChanceLight, Inc.	Inglewood	Los Angeles County	116	Layoff Permanent
01/07/2019	30/08/2019	26/07/2019	Harbill, Inc. DBA Crest Chevrolet	San Bernardino	San Bernardino County	98	Closure Permanent

Comparison | Summary by Month | TrapRange Java | Replace Text | SQL | File Locations

READY 80%

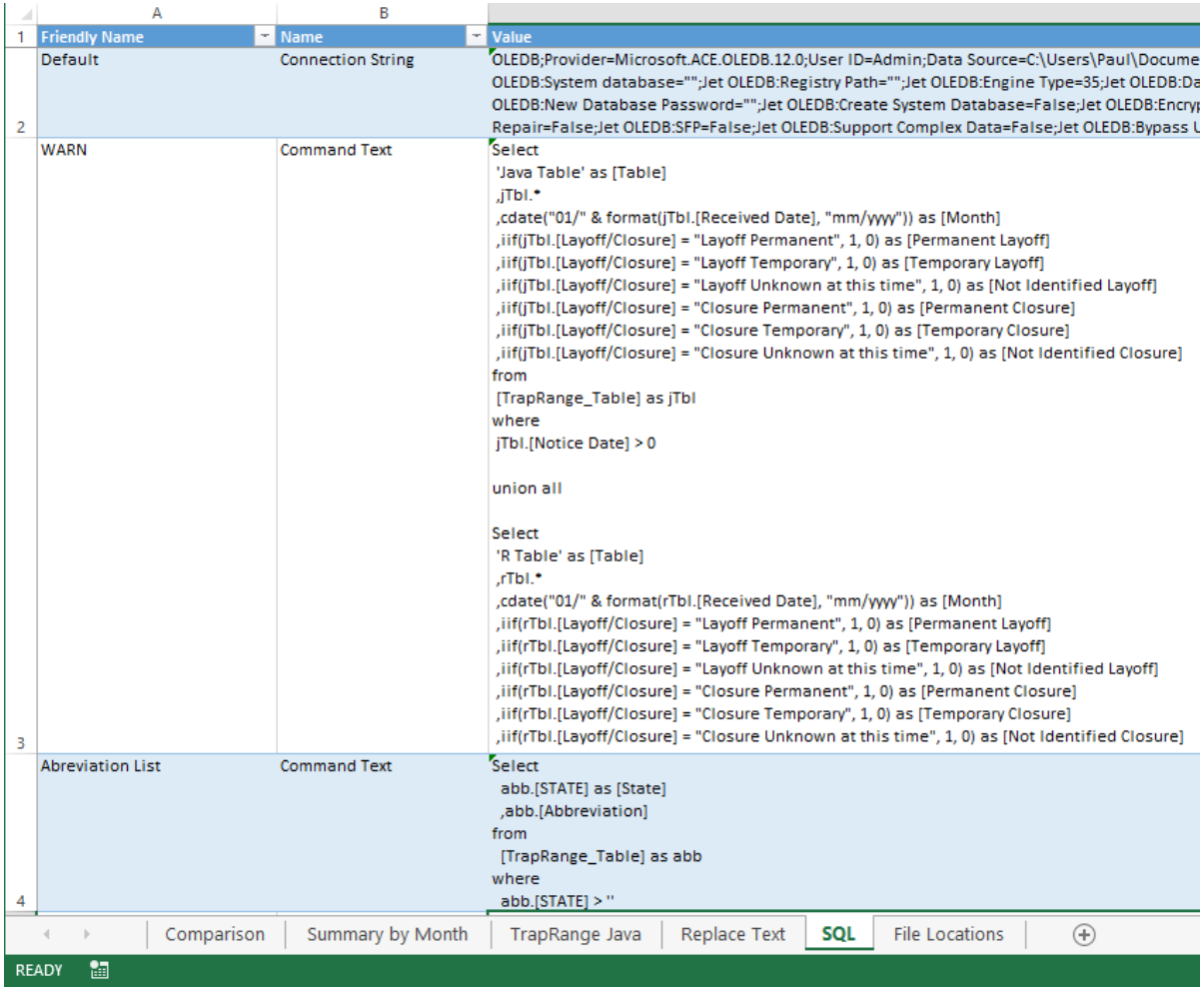
- There are currently six worksheets in the Excel file. This is because the file was built to compare operational requirements, function, accuracy and performance between the three methods (Java and VBA, R Script and VBA, and Power Query and VBA). Hence the 1st two tabs containing Pivot Tables using a Workbook Connection Data Link
- The TrapRange Java worksheet is where all the Java Module parameters need to be entered
    - The 1st dropdown list is for calling the VBA code functions to run



    - The Non Empty Column text box is essential and required. Usually tables have no missing data in the 1st column, but there may be times when the tables have a Pivot Style approach. At this point in time the VBA code is only designed to append column text to the column in the row above them when there is an Empty Non Empty Column

    - The Pivot SQL drop down list is also required as it is used to index the SQL Syntax used in the Linked Pivot tables of the 1st two worksheets

- This list is pulled from the Friendly Name column of the table in the SQL worksheet



	A	B	Value
1	Friendly Name	Name	
2	Default	Connection String	OLEDB;Provider=Microsoft.ACE.OLEDB.12.0;User ID=Admin;Data Source=C:\Users\Paul\Docume OLEDB:System database="";Jet OLEDB:Registry Path="";Jet OLEDB:Engine Type=35;Jet OLEDB:Da OLEDB:New Database Password="";Jet OLEDB:Create System Database=False;Jet OLEDB:Encryp Repair=False;Jet OLEDB:SFP=False;Jet OLEDB:Support Complex Data=False;Jet OLEDB:Bypass L
3	WARN	Command Text	Select 'Java Table' as [Table] ,jTbl.* ,cdate("01/" & format(jTbl.[Received Date], "mm/yyyy")) as [Month] ,iif(jTbl.[Layoff/Closure] = "Layoff Permanent", 1, 0) as [Permanent Layoff] ,iif(jTbl.[Layoff/Closure] = "Layoff Temporary", 1, 0) as [Temporary Layoff] ,iif(jTbl.[Layoff/Closure] = "Layoff Unknown at this time", 1, 0) as [Not Identified Layoff] ,iif(jTbl.[Layoff/Closure] = "Closure Permanent", 1, 0) as [Permanent Closure] ,iif(jTbl.[Layoff/Closure] = "Closure Temporary", 1, 0) as [Temporary Closure] ,iif(jTbl.[Layoff/Closure] = "Closure Unknown at this time", 1, 0) as [Not Identified Closure] from [TrapRange_Table] as jTbl where jTbl.[Notice Date] > 0 union all Select 'R Table' as [Table] ,rTbl.* ,cdate("01/" & format(rTbl.[Received Date], "mm/yyyy")) as [Month] ,iif(rTbl.[Layoff/Closure] = "Layoff Permanent", 1, 0) as [Permanent Layoff] ,iif(rTbl.[Layoff/Closure] = "Layoff Temporary", 1, 0) as [Temporary Layoff] ,iif(rTbl.[Layoff/Closure] = "Layoff Unknown at this time", 1, 0) as [Not Identified Layoff] ,iif(rTbl.[Layoff/Closure] = "Closure Permanent", 1, 0) as [Permanent Closure] ,iif(rTbl.[Layoff/Closure] = "Closure Temporary", 1, 0) as [Temporary Closure] ,iif(rTbl.[Layoff/Closure] = "Closure Unknown at this time", 1, 0) as [Not Identified Closure]
4	Abreviation List	Command Text	Select abb.[STATE] as [State] ,abb.[Abbreviation] from [TrapRange_Table] as abb where abb.[STATE] > ''

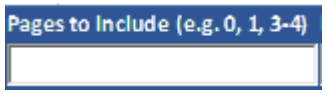
- As data connections can work with multiple files of the same type I have used variable type names in the From Clauses (e.g. From [RTable@].[R_Table] as rTbl). Then with VBA done a find and Replace on the SQL Syntax before passing it to the ADODB Connection

	A	B	C
	Friendly Name	Lookup	File Path
	Default	Connection@	C:\Users\Paul\Documents\Source Files\xlsm\TrapRange Java.xlsm
	WARN	RTable@	C:\Users\Paul\Documents\Source Files\xlsm\WARN Report for 7-1-2019 to 6-30-2020.xlsm
	WARN	PQuery@	C:\Users\Paul\Documents\Source Files\xlsm\WARN Report for 7-1-2019 to 6-30-2020.xlsm

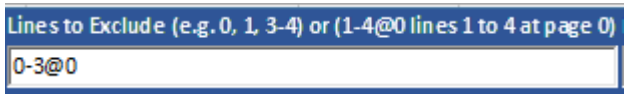
The Default connection uses the function Cell("filename", ref)

- The Where text box ensures that the Table in the TrapRange Java worksheet does not retrieve blank rows from the resulting csv file of the VBA and Java code. You can also specify any other filtering of that file

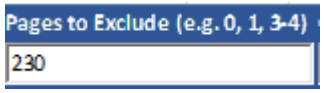
- The Pages to Include text box follows Java convention where indexes start with 0 as opposed to 1. This parameter will determine which pages are returned in the base csv file before any VBA code cleanses the data in that file. Pages are separated by commas and ranges of page numbers by dashes



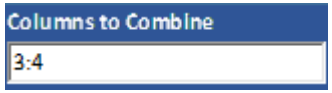
- The Lines to Exclude text box takes 4 types of entries: Single lines separated by commas; Line ranges separated by dashes; Single Lines at specific page numbers with the line number on the left of the @ symbol and the page number on the right; and/or Line ranges separated by a dash and on the left of the @ symbol with the page number on the right. All follow Java index convention, 0=1



- The Pages to Exclude text box again follows Java convention and is like the Pages to Include entry style

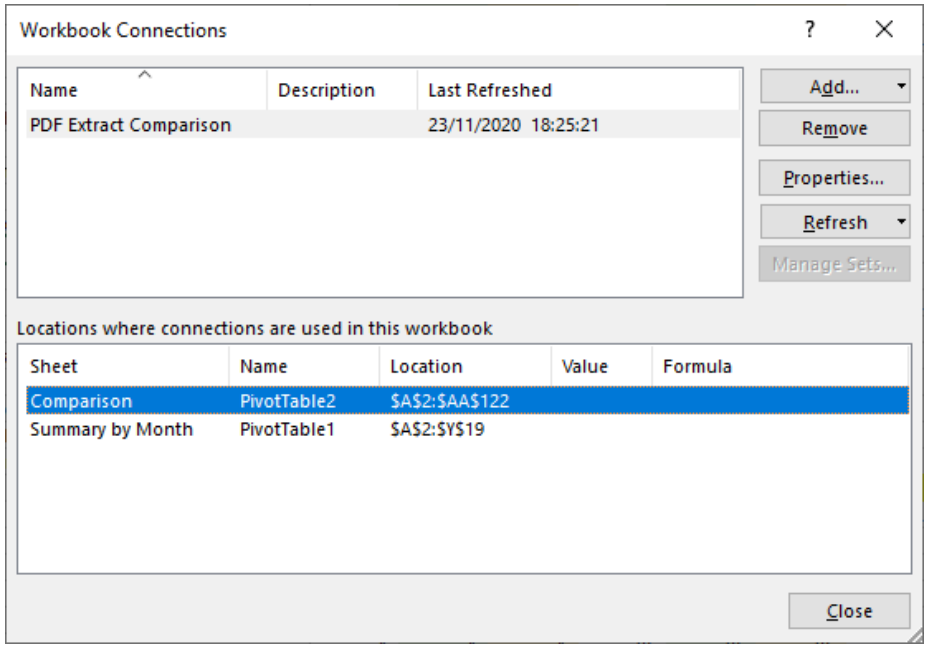


- The Columns to Combine text box is a VBA Parameter and follows VBA Indexing convention with 1 as the first instance. When pulling in the data without an entry in this field you may note that a column header has been split over two columns and that the second column is empty save the partial heading. You can re-join the header and eliminate the empty column by entering the two column numbers separated with a colon

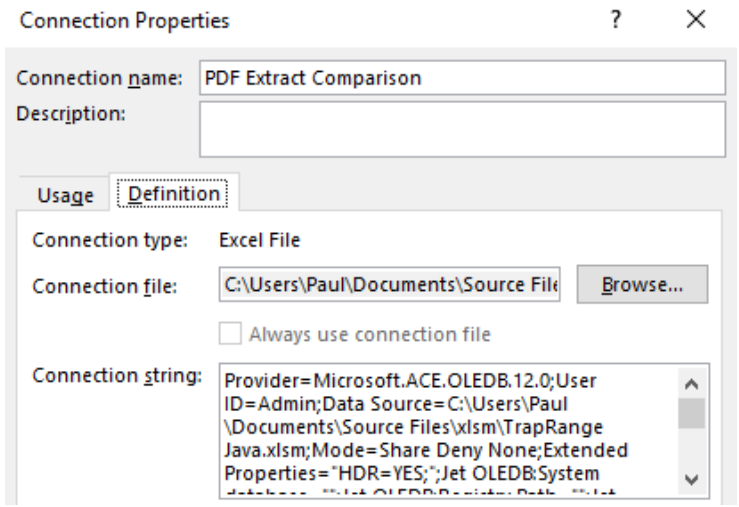


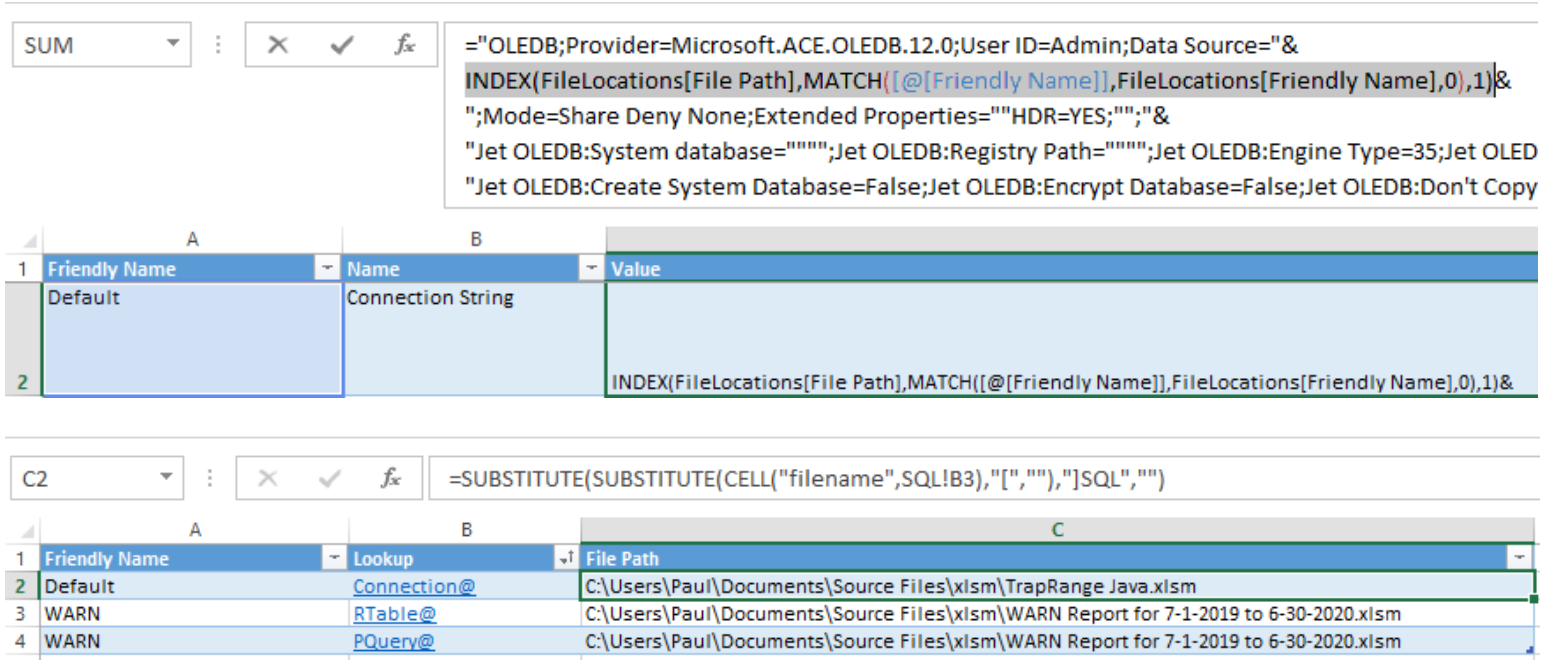
- The Columns to Split and Append text box again is a VBA Parameter and follows a similar to Cell Reference Parameter. R1C1-2:R2C3-4 stands for Row 1 Columns 1 to 2 as the recipient and Row 2 Column 3 to 4 as the data being moved. A new line after the last row entry in Column 1 is to receive all rows from Row 2 in Columns 3 to 4 and the headings and data in Columns 3 to 4 deleted after the transfer

- The Comparison worksheet Pivot table and the Summary by Month Pivot table both use the same Excel Workbook Connection and updating that connection updates both Pivots.
- The Excel Workbook Connection is updated by VBA code using the SQL Syntax from the SQL worksheet as indicated above

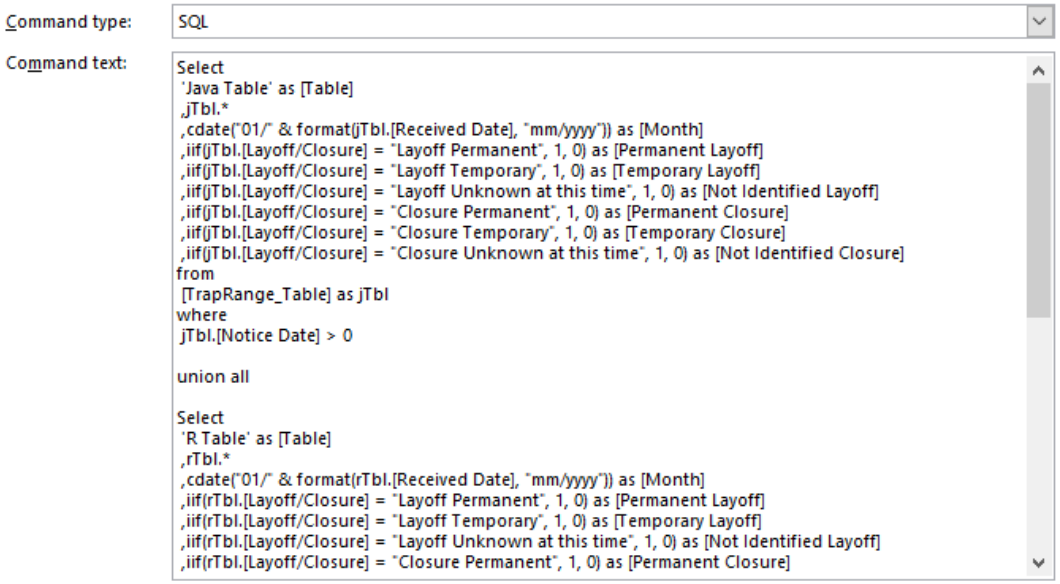


- The Connection String has been set as a Default to use the formula Cell("filename", Ref) in the SQL Table with the Friendly Name as Default

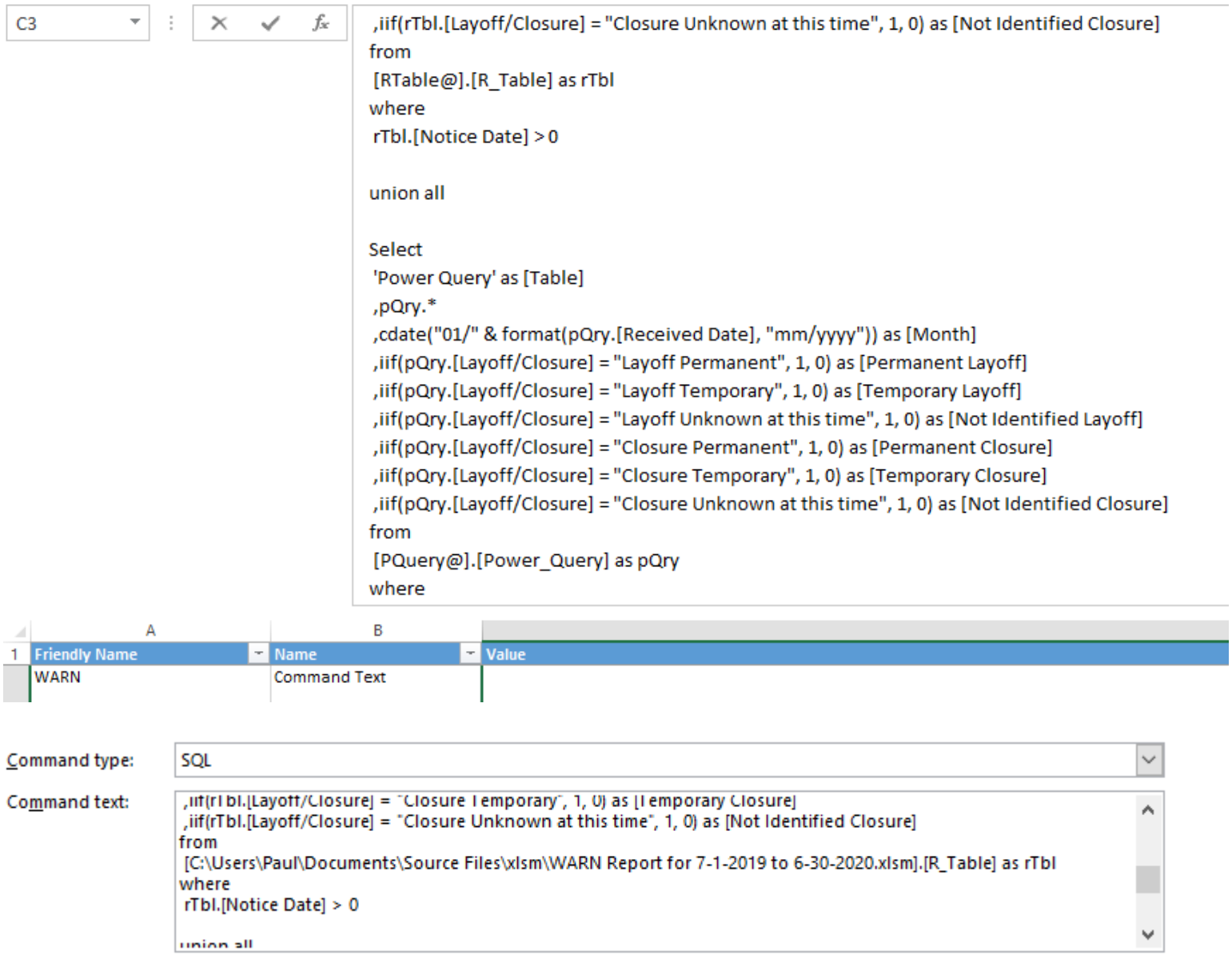




- The Command Text is entered by you using the TrapRange Java table as its original source and is entered in the SQL Syntax table in the SQL worksheet and given a Friendly Name which appears in the drop down list in the TrapRange Java worksheet



- Using the convention Name@ in the File Location part of other Linked Documents/Tables of your SQL Syntax and then in the Lookup column of the table in the File Locations worksheet along with the same Friendly Name for both will ensure that the VBA Code correctly updates the Workbook Connection Command Text replacing the Name@ text with the actual file path



- If the VBA code stops on an error you will either find the extracted csv file still open or you can open it to see where the code encountered an error. There are several reasons for the error to occur:
    - You did not exclude some lines that have a part of another table with a different number of columns or some text that is not in a table, so that changed the dimensions of the data being worked with
    - You did not exclude some pages that have text only in them or tables with a different number of columns
    - There may be some other reason for the error and you can fix it by simply analysing the extracted csv to find the possible cause and eliminate it by excluding that line in the variables you pass to the Java Module or adding a row to the Find and Replace Table should it be some text
    - The PDF is not up to standard and the returned CSV has some text that could not be formatted correctly by the Java module (i.e. the date fields may be joined together or returned as 1 2 / 0 3 / 2 0 2 0 instead of 12/03/2020). Unfortunately this is an issue that the Java module developer will have to overcome. Until then you have some less speedy other options that may work with these particular PDFs.