Kinetic INTERACTIVE

W: www.kinetic-interactive.com
E: info@kinetic-interactive.com
T: +866 586 0999

# SPREADSHEET MAGIC !

## with ColdFusion <cfspreadsheet>

Kevin D. Wright

Kinetic InterActive



www.kinetic-interactive.com info@kinetic-interactive.com + 866 586 0999

# Microsoft Excel

Released in 1985 Excel has grown to become arguably the one of the most critical of software used in organizations around the world today.

Excel is the Swiss army knife for business data analytics. It is a powerful jack of all trades solution for business needs such as quick data storage and manipulation, collaboration with data, automating tasks, and more.

Adobe ColdFusion has always made it easy to manipulate Excel spreadsheets using the <cfspreadsheet> tag.

The true power is revealed when using the strength of both ColdFusion and Excel joined together.

# Contents

This is the companion guide for the demo files available for download.


1 **JAVA POI**
Overview of the Apache Java POI libraries

2 **EXCEL DYNAMIC CHART**
Setup the the template for a dynamic chart

3 **EXCEL FUNCTIONS**
Various 'look-up' functions used in the 'real world' examples

4 **RESOURCES**
Links to different resources mentioned during the presentation.

# Java POI

## What is it?

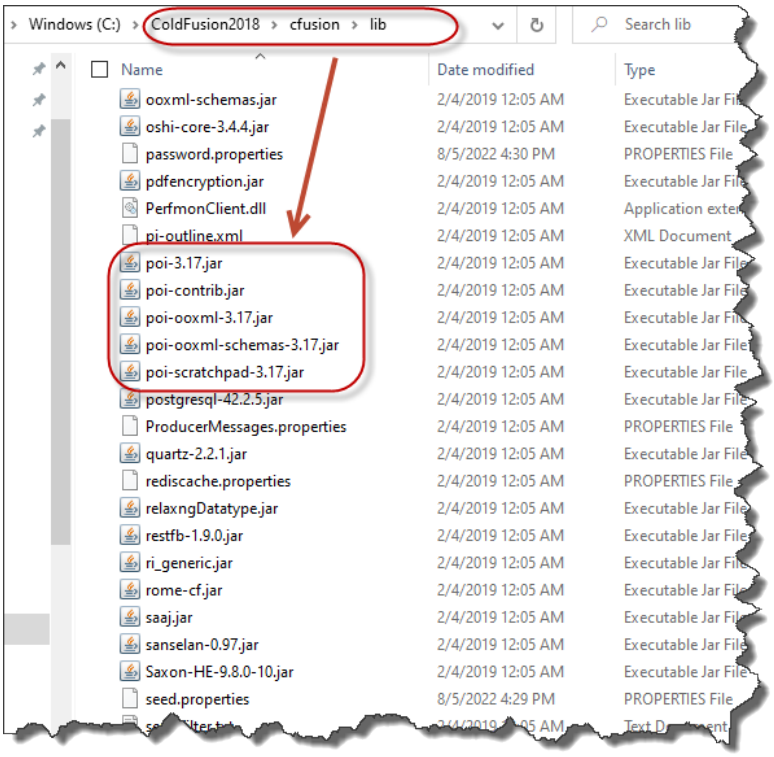
Apache POI (Poor Obfuscation Implementation) is a software project which is design and developed by Apache Software Foundation. It is a collection of pure Java libraries and used to read and write Microsoft office documents such as Excel, Word, PowerPoint, etc.

The objective of Apache POI is to design a cross-platform API which can manipulate various file formats of Microsoft Office and Open Office Documents.

It also supports OpenXML Format which is a new standards based XML file format found in Microsoft Office 2007 and 2008. It includes XLSX, DOCX and PPTX file formats.

We can use POI with any JVM based programming languages, like ColdFusion, that can import Java jar files.

These libraries are what is used by the <cfspreadsheet> tag.
The Java POI jar files are already included in the ColdFusion directory when installation is run.



# Excel Dynamic Chart

## Creating the Template for Demo #5

We will create an excel sheet with dynamic chart using excel named ranges (empty datasource) and then use this excel sheet as a template to create excel sheets with charts by just modifying the reference for named ranges to point the chart data from ColdFusion using Java POI.

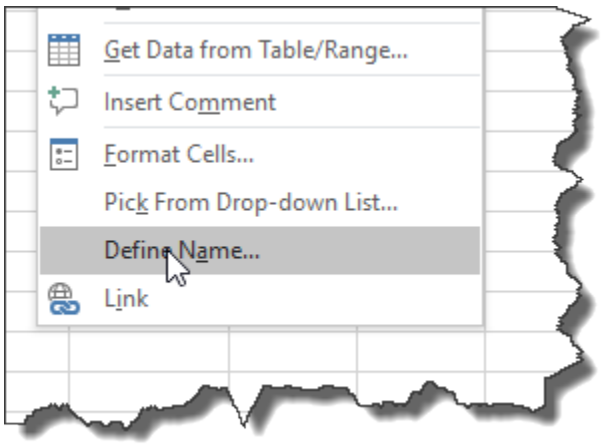
1. Create a new Excel file named DynamicChartDemo.xlsx
   Name the first sheet “DynamicChart”.

2. Identify the data and fields from which you want to make the dynamic chart. In our case, we are trying to generate a monthly sales report with data from two columns, 'Associates' and 'Sales'. We will create two named ranges with dummy values.

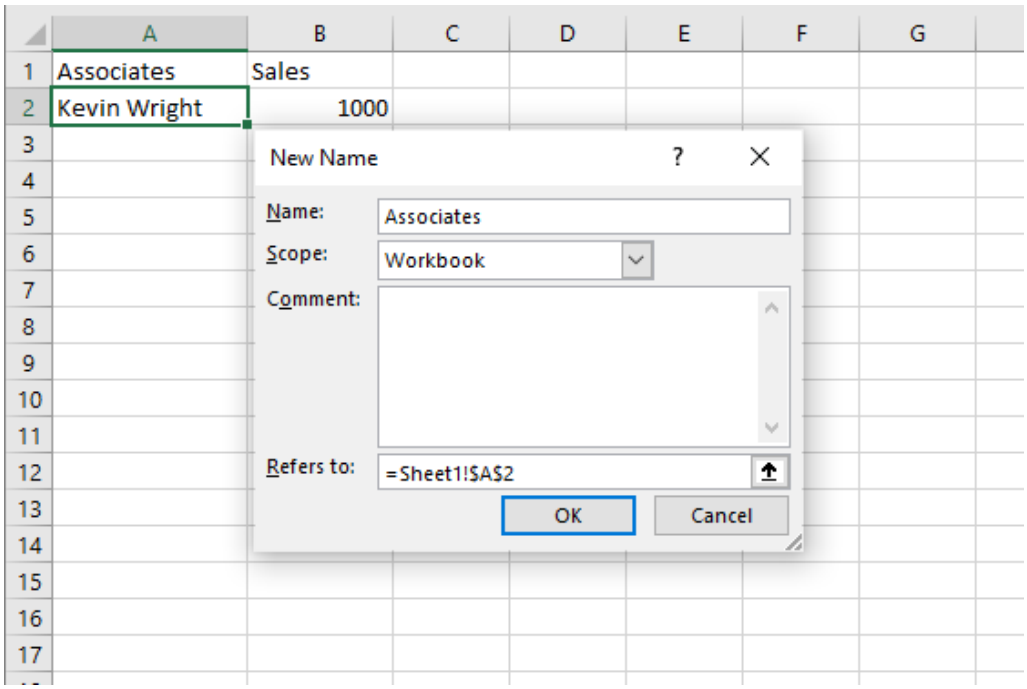
3. Enter the dummy data into the cells.

| | A | B | C |
|---|---|---|---|
| 1 | Associates | Sales | |
| 2 | Kevin Wright | 1000 | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

4. Highlight the 2nd cell in the first column, right click and select “Define Name”.

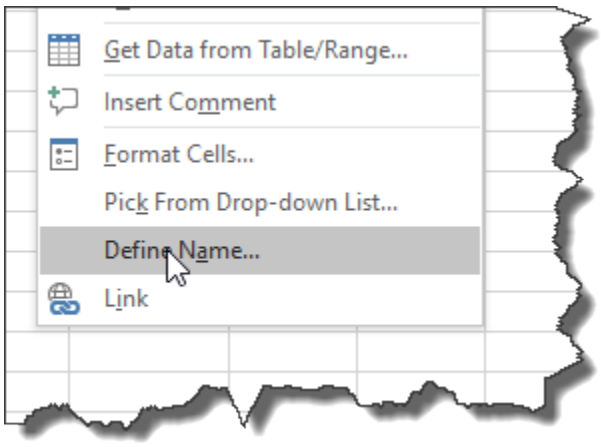



5. The Enter “Associates” as the “Name and the reference string should be “=Sheet1!$A$2” refering to the selected cell.

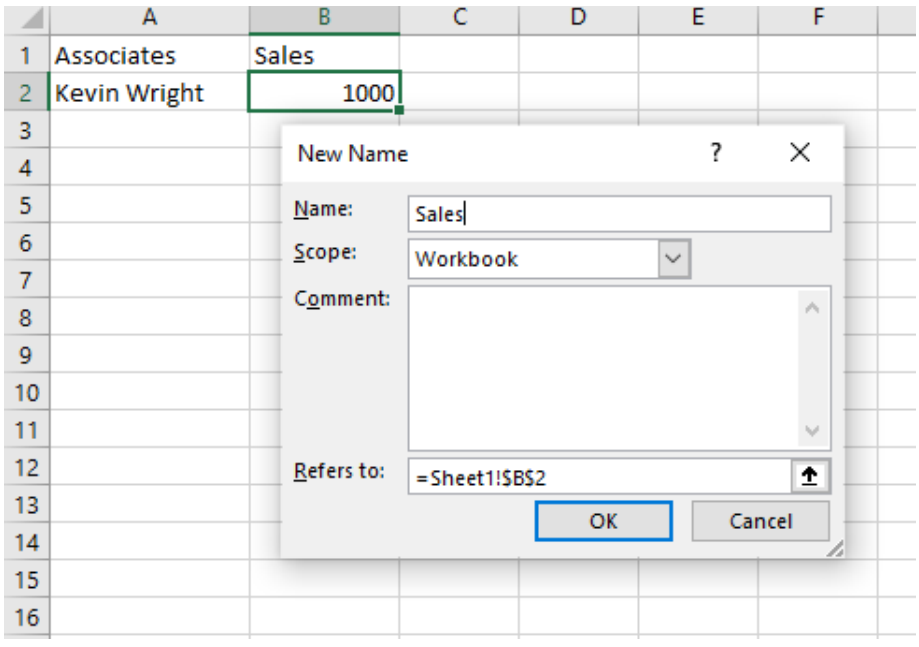



Click “OK”

6. Repeat the same steps for column B.
   Highlight the 2nd cell in the column, right click and select “Define Name”.

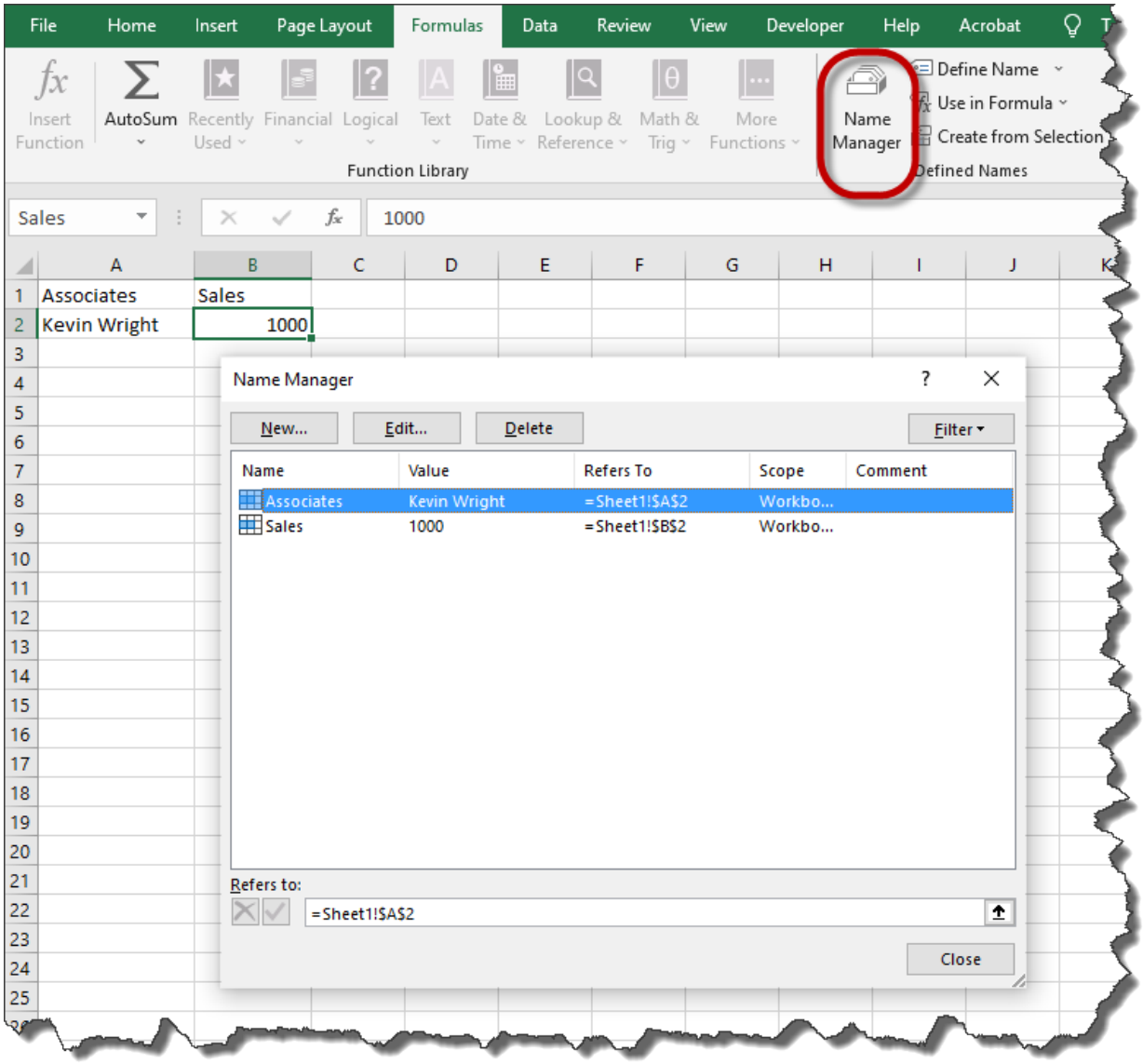



7. Enter “Sales” as the “Name” and the reference string
   should be “=Sheet1!$B$2” refering to the selected cell.

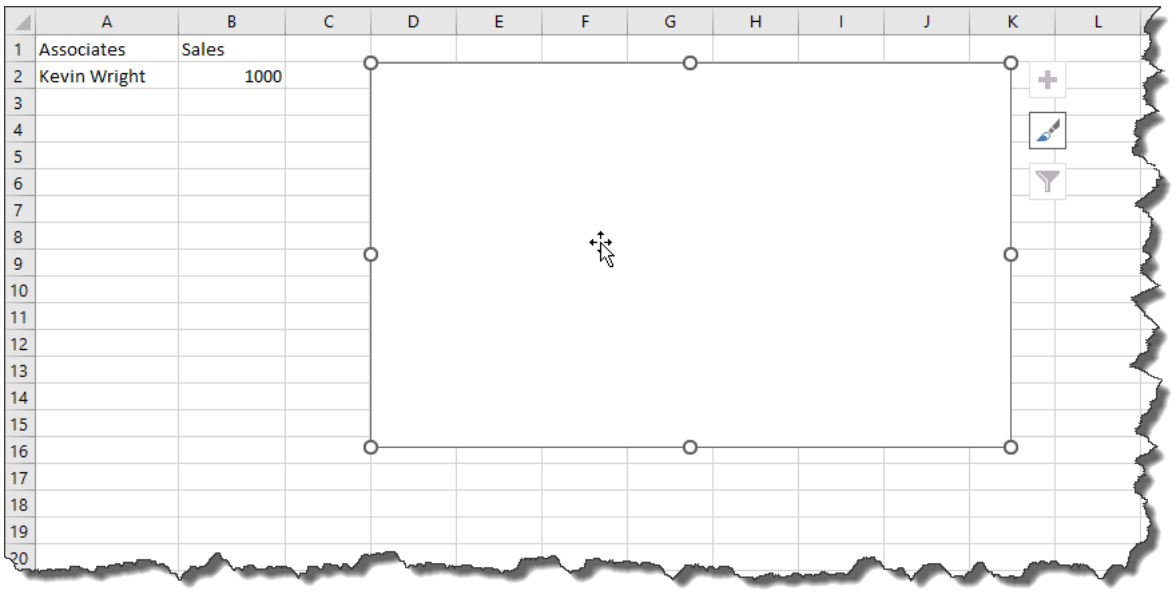



Click “OK”

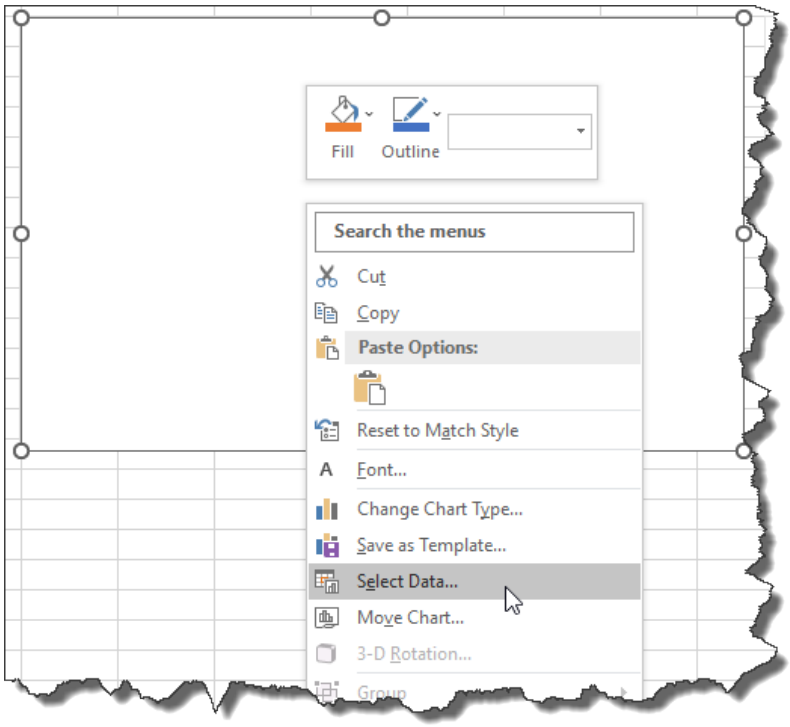
8. You can view and/or edit all defined named ranges with the “Name Manager”



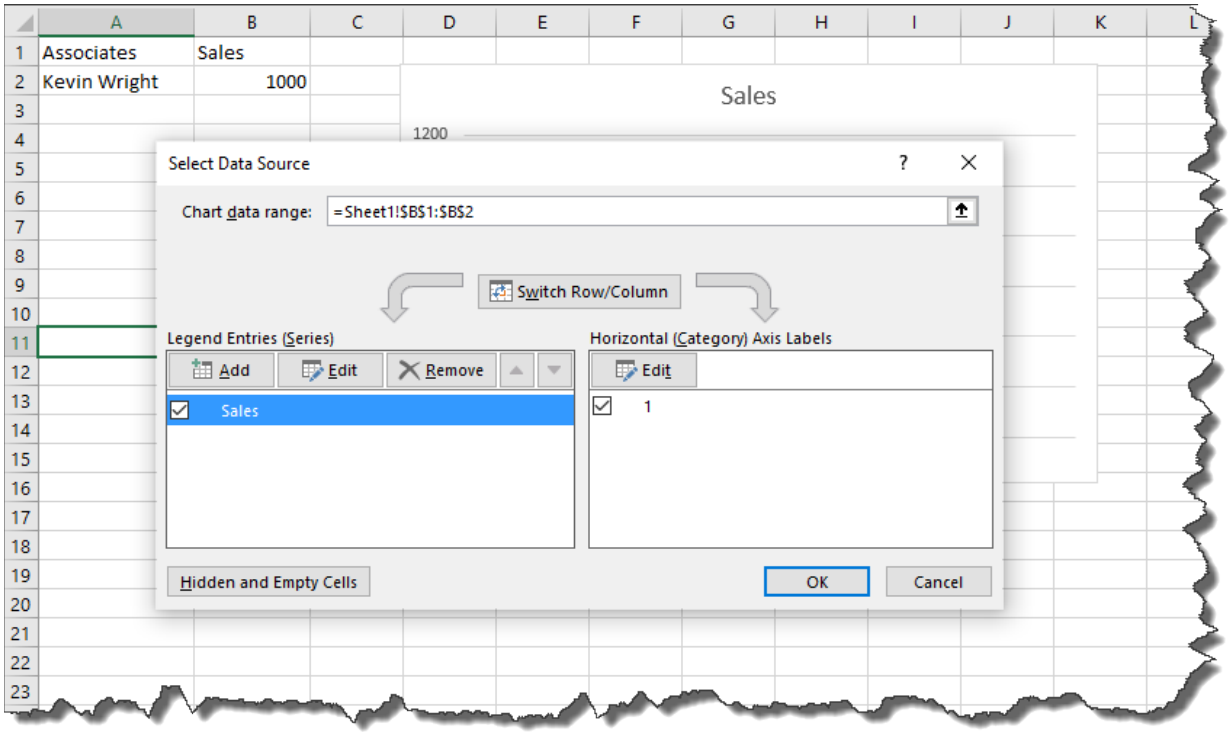
9. Insert a column chart to the sheet by clicking on the "Column chart" in the Insert menu tab..



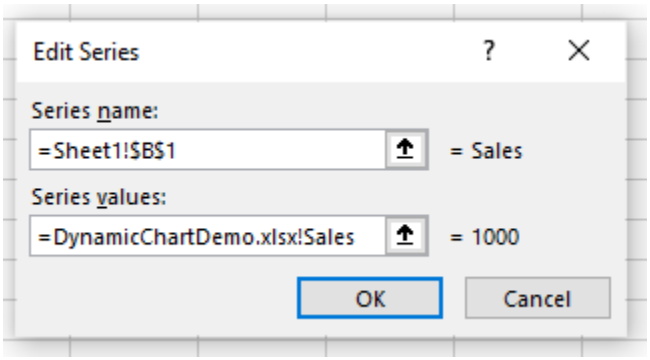

10. Right click on the chart and click on 'Select Data' option to open the 'Select Data Source' dialogue box.



11. Highlight cell range B1 to B2 (both cells).
    Make sure the series is “Sales” and is listed on the left side.
    If it is not, click the “Switch Row/Column” button.

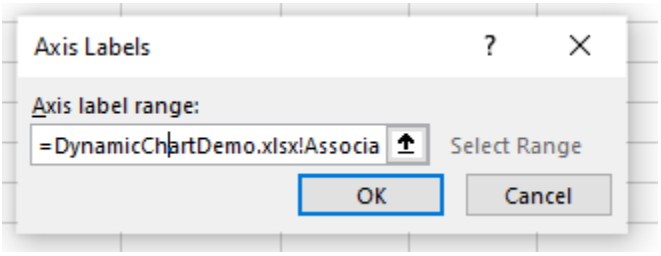


12. With the "Sales" series highlighted, click "Edit"
    Enter "Sheet1!$B$1" as the series name (value of cell B1) and
    "=DynamicChartDemo.xlsx!Sales" (the named range*) as the series values




    Click "OK"

13. With the "Axis Labels" highlighted, click "Edit"
    Enter "=DynamicChartDemo.xlsx!Associates" (the named range*) as the label range.




    Click "OK"

Your template is now complete and ready to use with the code in *"05_chart_dynamic.cfm"*

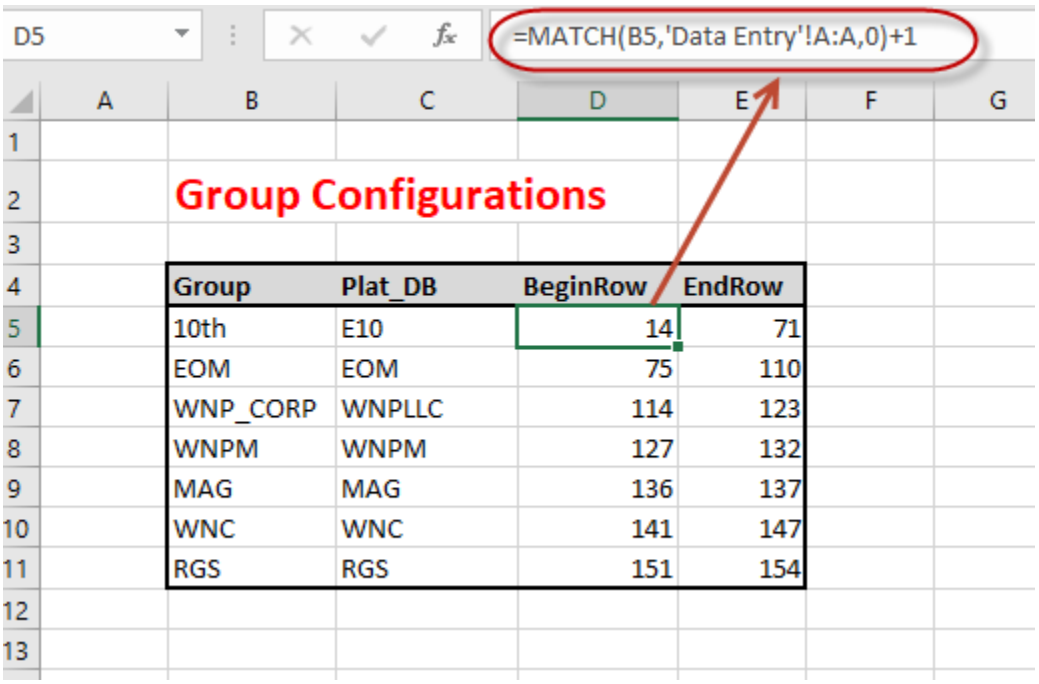
* *"DynamicChartDemo.xlsx" refers to the scope of the named range.*
*Scopes can be workbook level or individual sheet level.*

# Excel Functions

## Various 'Look-up' / 'Reference' functions used in the 'real-world example' demos

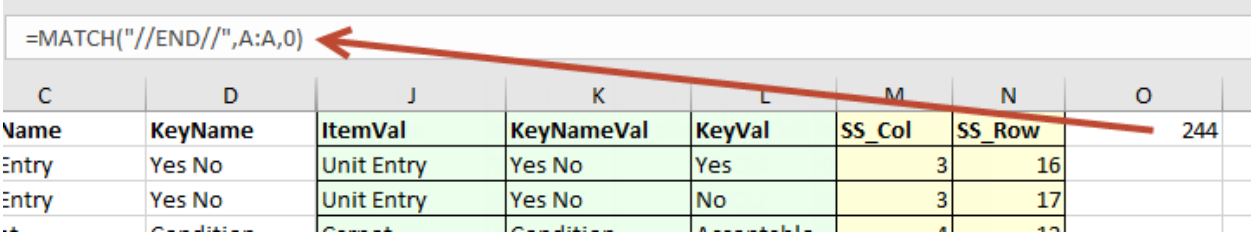
1. **MATCH ( )**

   This function 'matches' a string search in a range of cells and returns the row number. Here I am searching for the value entered in cell B5, and the range of cells is column A on the sheet named 'Data Entry'.




| Group | Plat_DB | BeginRow | EndRow |
|---|---|---|---|
| 10th | E10 | 14 | 71 |
| EOM | EOM | 75 | 110 |
| WNP_CORP | WNPLLC | 114 | 123 |
| WNPM | WNPM | 127 | 132 |
| MAG | MAG | 136 | 137 |
| WNC | WNC | 141 | 147 |
| RGS | RGS | 151 | 154 |

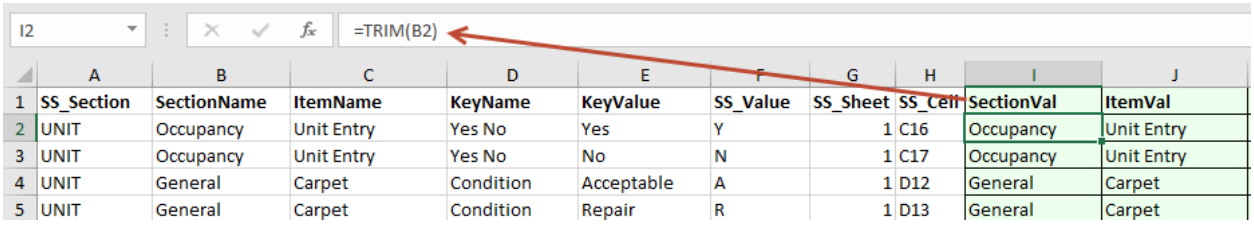
Here I am searching for the string "//END//" in column A of the current sheet. It returned the row of the first instance found (244).




| Name | KeyName | ItemVal | KeyNameVal | KeyVal | SS_Col | SS_Row |
|---|---|---|---|---|---|---|
| Entry | Yes No | Unit Entry | Yes No | Yes | 3 | 16 |
| Entry | Yes No | Unit Entry | Yes No | No | 3 | 17 |

2. ## TRIM ( )

This function 'trims' leading and trailing spaces from a string.
Pretty self explanatory, but this is quite useful when receiving spreadsheets with text or string values entered as data. This is a common problem especially as end users copy/paste from other applications.

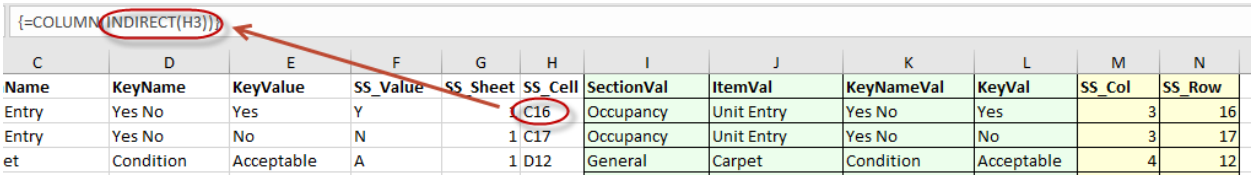


I2 =TRIM(B2)

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | SS_Section | SectionName | ItemName | KeyName | KeyValue | SS_Value | SS_Sheet | SS_Cell | SectionVal | ItemVal |
| 2 | UNIT | Occupancy | Unit Entry | Yes No | Yes | Y | 1 | C16 | Occupancy | Unit Entry |
| 3 | UNIT | Occupancy | Unit Entry | Yes No | No | N | 1 | C17 | Occupancy | Unit Entry |
| 4 | UNIT | General | Carpet | Condition | Acceptable | A | 1 | D12 | General | Carpet |
| 5 | UNIT | General | Carpet | Condition | Repair | R | 1 | D13 | General | Carpet |

3. ## INDIRECT ( )

This function evaluates the value of the referenced cell.
Here the value 'C16' is used in the formula by reference.

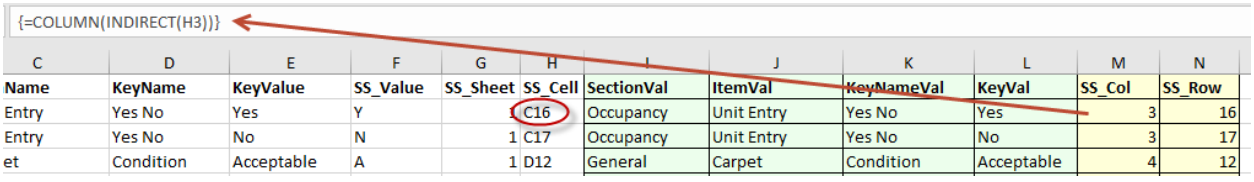


{=COLUMN(INDIRECT(H3))}

| C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | KeyName | KeyValue | SS_Value | SS_Sheet | SS_Cell | SectionVal | ItemVal | KeyNameVal | KeyVal | SS_Col | SS_Row |
| Entry | Yes No | Yes | Y | 1 | C16 | Occupancy | Unit Entry | Yes No | Yes | 3 | 16 |
| Entry | Yes No | No | N | 1 | C17 | Occupancy | Unit Entry | Yes No | No | 3 | 17 |
| et | Condition | Acceptable | A | 1 | D12 | General | Carpet | Condition | Acceptable | 4 | 12 |

4. ## COLUMN ( )

This function returns the column of a referenced cell.
Here it is returning the column of C16 (column 3).

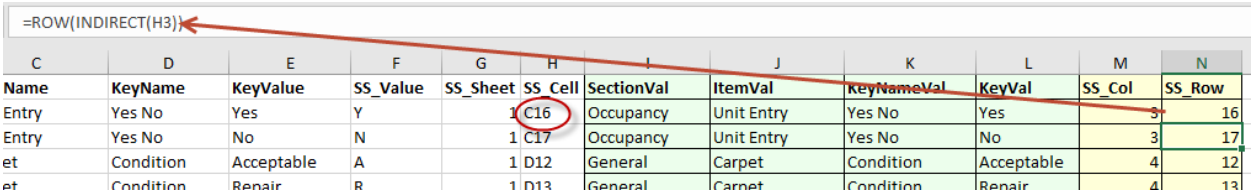


{=COLUMN(INDIRECT(H3))}

| C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | KeyName | KeyValue | SS_Value | SS_Sheet | SS_Cell | SectionVal | ItemVal | KeyNameVal | KeyVal | SS_Col | SS_Row |
| Entry | Yes No | Yes | Y | 1 | C16 | Occupancy | Unit Entry | Yes No | Yes | 3 | 16 |
| Entry | Yes No | No | N | 1 | C17 | Occupancy | Unit Entry | Yes No | No | 3 | 17 |
| et | Condition | Acceptable | A | 1 | D12 | General | Carpet | Condition | Acceptable | 4 | 12 |

5. ## ROW ( )

This function returns the row of a referenced cell.
Here it is returning the row of C16 (row 16).



=ROW(INDIRECT(H3))

| C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | KeyName | KeyValue | SS_Value | SS_Sheet | SS_Cell | SectionVal | ItemVal | KeyNameVal | KeyVal | SS_Col | SS_Row |
| Entry | Yes No | Yes | Y | 1 | C16 | Occupancy | Unit Entry | Yes No | Yes | 3 | 16 |
| Entry | Yes No | No | N | 1 | C17 | Occupancy | Unit Entry | Yes No | No | 3 | 17 |
| et | Condition | Acceptable | A | 1 | D12 | General | Carpet | Condition | Acceptable | 4 | 12 |
| et | Condition | Repair | R | 1 | D13 | General | Carpet | Condition | Repair | 4 | 13 |

# Resources

## Where to go from here

1. **Java POI Documentaiton**
   Documentation for recent releases of the Java POI API.

   https://poi.apache.org/apidocs/

2. **cfsimplicity**
   Standalone library for working with spreadsheets in CFML by Julian Halliwell.
   Adds extra functionality not available in ColdFusion.

   https://github.com/cfsimplicity
   or
   https://www.forgebox.io/view/spreadsheet-cfml

3. **Excel Functions**
   A list of functions by category

   https://exceljet.net/excel-functions

Kinetic INTERACTIVE

Our solutions are one part technology,
one part creativity,
and all business.