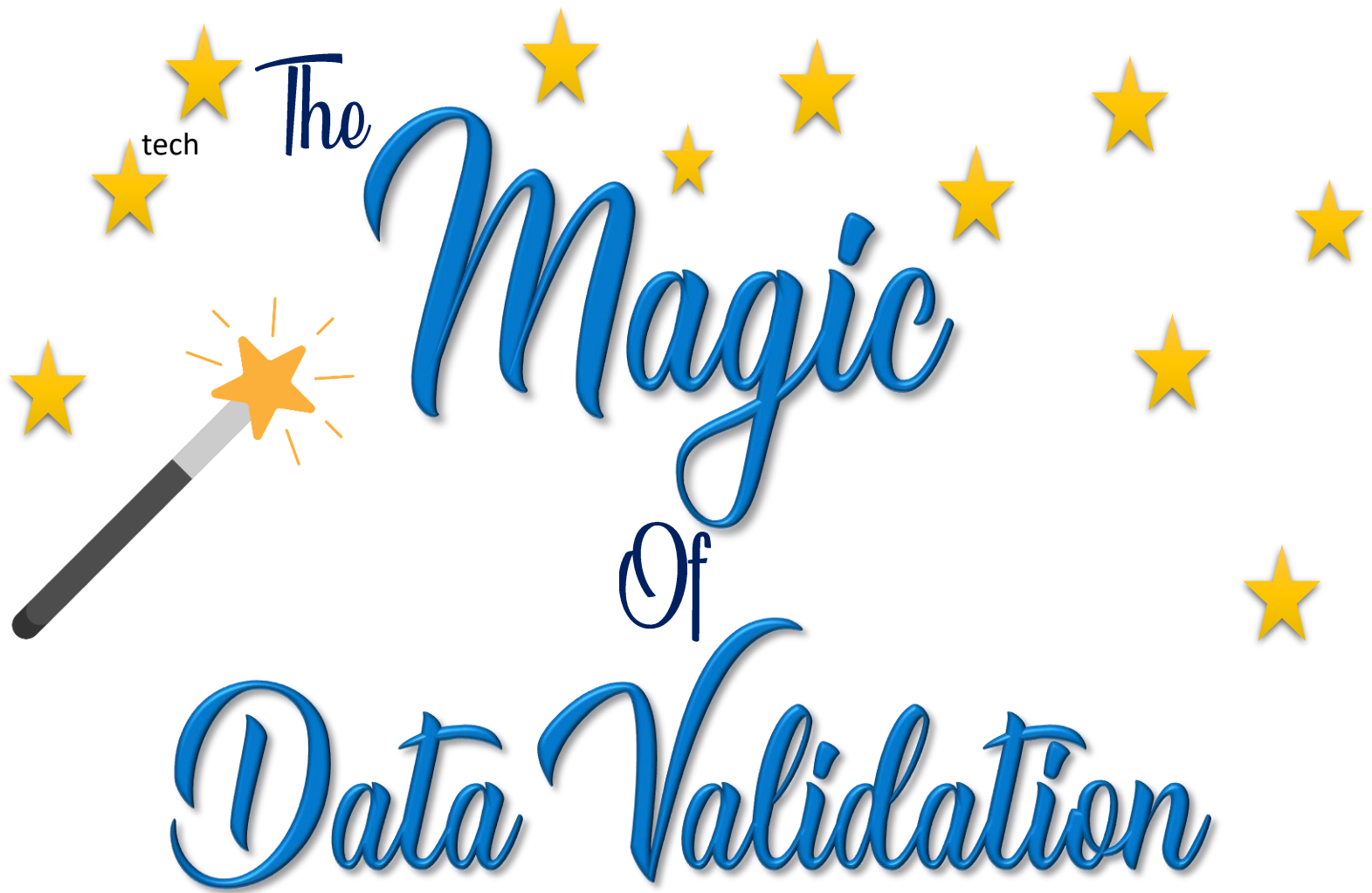


Push it to the limit... and Beyond



tech

The Magic Of Data Validation

All Rights Reserved©

Nabil Mourad



Table of Contents

The work situation	6
Creating our First drop list	8
Creating a "Dependent- Diminishing" drop list	10
The preparation list.....	11
Referencing a spilled array.....	11
Preparing for our drop lists with multiple levels of dependencies	13
Three functions - Same concept.....	14
Unique list of Floors	15
Creating the third drop list.....	16
Scratch the limit of Data Validation in Z4	18
List of Owned Suites in any Floor	19
The Six possible options	22
Preparing for the Switch function:	29
Hide contents of Cell AA1	33
Resetting 2 drop lists (Y3 & Z3) with a code	33
Open the VBE.....	33
Writing the Code.....	35
Getting Information based upon selection.....	35
First VLOOKUP	37
Second VLOOKUP.....	38
Combining the 2 VLOOKUP functions.....	39
A little issue to avoid.....	39

A second IFERROR function	39
An Input Message	39
Finalizing the project.....	40
Two conditional formatting rules.....	40
Hiding Columns.....	41
Zoom In and Out	41
Create 2 macros for the Zooming Functionality	42
Assigning Macros	42

Table of Figures

FIGURE 1- SOURCE DATA	7
FIGURE 2- PROJECT RANGE	8
FIGURE 3- PREPARATION AREA	8
FIGURE 4- DATA VALIDATION	9
FIGURE 5- LIST VALUES FOR THE FIRST DROP LIST	9
FIGURE 6- INPUT MESSAGE	10
FIGURE 7- THE FIRST DROP LIST IN Y2	10
FIGURE 8- COUNTIF AND FILTER FUNCTIONS	11
FIGURE 9- DEPENDENT DATA VALIDATION	12
FIGURE 10- TESTING THE SECOND DROP LIST	13
FIGURE 11- THE SOURCE LIST	14
FIGURE 12- FLOOR/ SUITE/ RESIDENT	15
FIGURE 13- UNIQUE LIST OF FLOORS	16
FIGURE 14- SWITCH FUNCTION FOR 3 OPTIONS	18
FIGURE 15- THREE POSSIBILITIES IN Y3 DROP LIST	19
FIGURE 16- OWNED SUITES IN FLOOR 01	20
FIGURE 17- LIST OF OWNED SUITES IN EACH FLOOR	20
FIGURE 18- RESIDENT NAMES IN OWNED SUITES IN FLOOR 01	21
FIGURE 19- OWNERS LIST IN EACH FLOOR	22
FIGURE 20- ONE SUITE RETURNED	23
FIGURE 21- LIST OF OWNED SUITES FOR A SELECTED FLOOR	24
FIGURE 22- FIRST RESIDENT NAME FOR A SELECTED FLOOR	25
FIGURE 23- ALL RESIDENT NAMES OF A SELECTED FLOOR	25
FIGURE 24- FLOOR NUMBER OF A SELECTED SUITE	26
FIGURE 25- RESIDENT NAME FOR A SELECTED SUITE	27
FIGURE 26- THE SIX PREPARATION FUNCTIONS	28
FIGURE 27- COMBINE SELECTIONS FROM THE 2 DROP LISTS	29
FIGURE 28- OPTION1 FLOOR- SUITE	31
FIGURE 29- OPTION2 FLOOR- RESIDENT	31
FIGURE 30- OPTION3 SUITE-FLOOR	32
FIGURE 31- OPTION4 SUITE-RESIDENT	32
FIGURE 32- OPTION5 RESIDENT- SUITE	33
FIGURE 33- OPTION6 RESIDENT- FLOOR	33
FIGURE 34- SWITCHING TO VBE	34
FIGURE 35- SELECT THE CHANGE EVENT	34
FIGURE 36-DELETE THE SELECTION CHANGE EVENT	34
FIGURE 37- THE CHANGE EVENT CODE	35
FIGURE 38- THE R1C1 REFERENCE	36
FIGURE 39- COMPARE COLUMN LETTERS AND NUMBERS	37
FIGURE 40- TESTING THE FIRST VLOOKUP	38
FIGURE 41- TESTING THE SECOND VLOOKUP	38

FIGURE 42- INPUT MESSAGE IN Z3	40
FIGURE 43- PAST DUE CONDITIONAL FORMATTING	40
FIGURE 44- OVER PMT CONDITIONAL FORMATTING	41
FIGURE 45- RESIZE AND REPOSITION ICONS	42
FIGURE 46- ASSIGN A MACRO TO ZOOM IN	43
FIGURE 47- THE ASSIGN MACRO DIALOG BOX	44
FIGURE 48- ASSIGNING A ZOOM OUT MACRO	44

The Magic of Data Validation

Introduction

Dynamic Data Validation... is a Game changer in Excel, at the basic level it controls what a user can enter or cannot enter in a cell. It also makes data entry easier and ensures the integrity of your worksheet by avoiding typing errors when users can simply select an option from a drop list.

However, in more advanced scenario drop lists can make a regular worksheet look as a dynamic dashboard.

With drop lists, you can select input values for formulas and functions, you can switch charts or source values for Power Query, You can create random quizzes or name sheets and you can also have the contents change based on another drop list.

In this article, we explore the full potential of drop lists by creating basic drop lists, diminishing drop lists and then we create dependent drop lists with 6 levels of dependency. We will learn about the extreme power of combining Drop lists with Dynamic arrays and touch the limit of what Data Validation can do.

So, Let's dive in

The work situation

Skymark is a nice high rise building of 35 floors and 8 units on each floor. While most units are owned, few units are rented. The building is well maintained due to the monthly fees each resident must pay beside some individual repair charges on some units.

The property management board wants to track the amounts paid by Owners and whether there are amounts past due or may be over payment.

To easily retrieve the information, they need a simple yet powerful dynamic way to search by Floor, by Suite number or by Resident's name.

Push it to the limit... and beyond

So we created a set of drop lists with up to 6 levels of dependency that can extract the information by any search criteria

You can download the exercise File and follow along.

The spreadsheet we have, shows a list of 23 columns and 274 rows.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Skymark Building Revenue 2019																						
2	Floor	Suite	Resident	Type	Phone	Maintenance/M	Repairs/y	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total Paid	Total Due	Past Due	Over Paid
3	01	0101	K Benedick	Owned	905 600-2436	\$1,156.00	\$1,467.00	\$924.00	\$1,046.00	\$1,045.00	\$951.00	\$1,033.00	\$1,169.00	\$947.00	\$843.00	\$940.00	\$1,000.00	\$1,215.00	\$804.00	\$11,917.00	\$16,339.00	\$3,422.00	\$0.00
4	01	0102	L Leroux	Owned	905 719-1815	\$1,234.00	\$1,488.00	\$847.00	\$906.00	\$891.00	\$1,208.00	\$1,145.00	\$873.00	\$1,181.00	\$1,113.00	\$988.00	\$989.00	\$834.00	\$1,240.00	\$12,216.00	\$16,296.00	\$4,081.00	\$0.00
5	01	0103	K Hassab	Owned	647 815-3003	\$876.00	\$1,014.00	\$1,318.67	\$1,318.67	\$500.00	\$500.00	\$1,318.67	\$1,318.67	\$0.00	\$1,318.67	\$1,318.67	\$1,318.67	\$1,318.67	\$0.00	\$11,549.33	\$11,526.00	\$0.00	\$23.33
6	01	0104	U Leedham	Owned	416 773-9208	\$795.00	\$661.00	\$1,202.00	\$999.00	\$846.00	\$845.00	\$1,247.00	\$1,055.00	\$1,147.00	\$805.00	\$808.00	\$866.00	\$902.00	\$851.00	\$11,573.00	\$10,201.00	\$0.00	\$1,372.00
7	01	0105	J Yazdani	Rental	647 269-3753	\$964.58	\$0.00	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$964.58	\$11,575.00	\$11,575.00	\$0.00	\$0.00
8	01	0106	K Beckford	Owned	905 225-2517	\$950.00	\$265.00	\$1,236.00	\$1,033.00	\$902.00	\$802.00	\$820.00	\$872.00	\$1,129.00	\$1,224.00	\$1,099.00	\$1,089.00	\$985.00	\$963.00	\$12,154.00	\$11,665.00	\$0.00	\$489.00
9	01	0107	Q Cole	Owned	416 853-4727	\$1,178.00	\$1,619.00	\$1,007.00	\$1,048.00	\$1,203.00	\$939.00	\$983.00	\$999.00	\$951.00	\$1,249.00	\$864.00	\$1,010.00	\$1,018.00	\$1,037.00	\$12,308.00	\$15,755.00	\$3,447.00	\$0.00
10	01	0108	Q Kosareva	Owned	416 355-8446	\$1,225.00	\$1,632.00	\$1,196.00	\$1,026.00	\$1,207.00	\$845.00	\$822.00	\$868.00	\$1,090.00	\$969.00	\$814.00	\$830.00	\$1,252.00	\$916.00	\$11,835.00	\$16,332.00	\$4,497.00	\$0.00
11	02	0201	X Cuentia	Owned	237 378-8104	\$1,156.00	\$1,652.00	\$878.00	\$1,183.00	\$846.00	\$955.00	\$1,163.00	\$1,165.00	\$943.00	\$939.00	\$1,191.00	\$1,103.00	\$1,001.00	\$842.00	\$12,209.00	\$15,524.00	\$3,315.00	\$0.00
12	02	0202	I Jin	Owned	647 914-2959	\$1,234.00	\$185.00	\$1,251.00	\$944.00	\$931.00	\$1,056.00	\$1,070.00	\$853.00	\$1,110.00	\$927.00	\$1,081.00	\$1,233.00	\$813.00	\$1,054.00	\$12,323.00	\$14,993.00	\$2,670.00	\$0.00
13	02	0203	O Nelson	Owned	647 398-2259	\$876.00	\$782.00	\$1,040.00	\$1,126.00	\$1,035.00	\$1,251.00	\$1,252.00	\$838.00	\$1,071.00	\$827.00	\$1,234.00	\$892.00	\$1,256.00	\$1,000.00	\$12,822.00	\$11,294.00	\$0.00	\$1,528.00
14	02	0204	F Pathak	Owned	647 806-5083	\$795.00	\$1,546.00	\$1,095.00	\$818.00	\$835.00	\$1,206.00	\$937.00	\$1,244.00	\$979.00	\$908.00	\$1,067.00	\$1,149.00	\$1,107.00	\$1,200.00	\$12,535.00	\$11,086.00	\$0.00	\$1,449.00
15	02	0205	S Harding	Owned	647 539-1971	\$910.00	\$1,266.00	\$1,318.67	\$1,318.67	\$1,318.67	\$1,318.67	\$1,318.67	\$600.00	\$1,318.67	\$1,318.67	\$500.00	\$1,318.67	\$1,318.67	\$820.00	\$13,788.00	\$12,186.00	\$0.00	\$1,602.00
16	02	0206	E Grefton	Rental	237 524-7784	\$1,012.33	\$0.00	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$1,012.33	\$12,148.00	\$12,148.00	\$0.00	\$0.00
17	02	0207	L Youkhanna	Rental	416 417-9744	\$1,181.50	\$0.00	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$1,181.50	\$14,178.00	\$14,178.00	\$0.00	\$0.00
18	02	0208	F Thiapa	Owned	237 954-2439	\$1,225.00	\$1,098.00	\$993.00	\$1,087.00	\$1,210.00	\$928.00	\$934.00	\$1,212.00	\$915.00	\$1,121.00	\$1,059.00	\$871.00	\$994.00	\$1,157.00	\$12,481.00	\$15,798.00	\$3,317.00	\$0.00
19	03	0301	Z Sparkes	Owned	905 700-4500	\$1,156.00	\$839.00	\$1,228.00	\$1,230.00	\$1,225.00	\$832.00	\$904.00	\$1,157.00	\$800.00	\$926.00	\$1,276.00	\$843.00	\$1,087.00	\$1,095.00	\$12,603.00	\$14,711.00	\$2,108.00	\$0.00
20	03	0302	P Wong	Owned	237 538-4816	\$1,234.00	\$694.00	\$1,173.00	\$1,131.00	\$1,013.00	\$1,236.00	\$996.00	\$1,036.00	\$1,275.00	\$1,170.00	\$1,051.00	\$1,219.00	\$1,257.00	\$804.00	\$13,361.00	\$15,502.00	\$2,141.00	\$0.00
21	03	0303	K Sung	Owned	647 901-1702	\$876.00	\$1,306.00	\$1,013.00	\$964.00	\$956.00	\$982.00	\$880.00	\$974.00	\$805.00	\$852.00	\$925.00	\$1,221.00	\$987.00	\$983.00	\$11,442.00	\$11,818.00	\$376.00	\$0.00
22	03	0304	H Rea	Owned	416 891-7534	\$795.00	\$1,343.00	\$1,318.67	\$1,118.67	\$840.00	\$840.00	\$840.00	\$840.00	\$840.00	\$840.00	\$840.00	\$840.00	\$840.00	\$957.00	\$10,954.34	\$10,883.00	\$0.00	\$71.34
23	03	0305	O Menezes	Owned	416 689-2614	\$910.00	\$371.00	\$1,203.00	\$1,106.00	\$1,004.00	\$1,231.00	\$1,021.00	\$991.00	\$1,252.00	\$1,020.00	\$900.00	\$868.00	\$1,166.00	\$1,200.00	\$12,962.00	\$11,291.00	\$0.00	\$1,671.00
24	03	0306	J Thomas	Owned	647 314-2423	\$872.00	\$1,846.00	\$1,034.00	\$868.00	\$1,082.00	\$863.00	\$1,066.00	\$1,253.00	\$1,284.00	\$1,010.00	\$823.00	\$918.00	\$861.00	\$1,188.00	\$12,230.00	\$12,310.00	\$80.00	\$0.00
25	03	0307	T Nolan	Owned	647 164-1392	\$1,178.00	\$27.00	\$1,146.00	\$805.00	\$1,129.00	\$982.00	\$843.00	\$809.00	\$1,028.00	\$1,189.00	\$1,034.00	\$1,113.00	\$1,164.00	\$1,021.00	\$12,263.00	\$14,163.00	\$1,900.00	\$0.00
26	03	0308	E Persaud	Owned	237 181-1102	\$1,225.00	\$467.00	\$1,009.00	\$1,222.00	\$823.00	\$1,018.00	\$876.00	\$1,174.00	\$1,142.00	\$1,113.00	\$890.00	\$880.00	\$881.00	\$1,272.00	\$12,300.00	\$15,167.00	\$2,867.00	\$0.00
27	04	0401	D Leung	Rental	905 932-9994	\$1,171.33	\$0.00	\$1,171.33	\$1,172.33	\$1,173.33	\$1,174.33	\$1,175.33	\$1,176.33	\$1,177.33	\$1,178.33	\$1,179.33	\$1,180.33	\$1,181.33	\$1,182.33	\$14,122.00	\$14,056.00	\$0.00	\$66.00
28	04	0402	I Hance	Owned	905 547-2344	\$1,234.00	\$1,231.00	\$891.00	\$922.00	\$1,191.00	\$1,079.00	\$888.00	\$928.00	\$811.00	\$1,051.00	\$1,016.00	\$900.00	\$1,221.00	\$1,156.00	\$12,054.00	\$16,039.00	\$3,985.00	\$0.00
29	04	0403	U Chen	Owned	905 482-1576	\$876.00	\$1,418.00	\$1,267.00	\$1,051.00	\$1,052.00	\$1,217.00	\$838.00	\$928.00	\$1,079.00	\$1,023.00	\$802.00	\$972.00	\$1,270.00	\$832.00	\$12,331.00	\$11,930.00	\$0.00	\$401.00

Figure 1- Source data

The columns store:

Floor, Suite number, Resident name, Type (Owned/Rental), Maintenance, Repairs, 12 Months collection, Total Paid, Total due, Past Due, Over Payment.

Rental units pay their obligations with postdated cheques, so we are mainly concerned about tracking payments for **Owned** units.

We will create our solution in the same worksheet in the range Y1:AD3 where we just have some labels in row 2

Push it to the limit... and beyond

	Y	Z	AA	AB	AC	AD	AE
1	Select an Option ↓	Search By Details ↓					
2	Floor	Resident	Total Paid	Total Due	Past Due	Over PMT	
3							
4							

Figure 2- Project Range

We also need to have a ***Preparation Area*** which is the real kitchen where we prepare for our work. My preparation area will be to the right side of column AF

AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR
<div> <div> <div> <div>Floor</div> <div>Suite</div> <div>Resident</div> </div> <div> <div>Floor-Suite FS</div> <div>Floor-Resident FR</div> <div>Suite-Floor SF</div> <div>Suite-Resident SR</div> <div>Resident-Suite RS</div> <div>Resident-Floor RF</div> </div> <div> <div>In Y3 for DL</div> <div>In Z3 for DL</div> <div>LEN</div> </div> <div>Owned Units</div> <div> <div>Floor</div> <div>Suites</div> <div>Residents</div> </div> </div> <div> <div>Data Preparation Section</div> <div> <div>Filter Function with 2 Conditions</div> <div> <div>Floor</div> <div>Suite</div> </div> <div> <div>Floor</div> <div>Resident</div> </div> </div> </div> </div>													

Figure 3- Preparation Area

The user should be able to search either by Floor, Suite or Resident, so we start by creating a basic Drop List in Cell Y2 for the 3 options (available in AG2:AG4)

Creating our First drop list

Select cell Y2.

Then click on the Data Tab of the Ribbon ► and click on Data Validation

Push it to the limit... and beyond

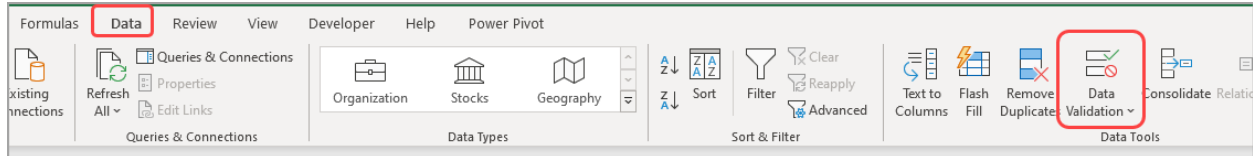


Figure 4- Data Validation

Under "Allow", select "List". With the mouse pointer in the source ► click and drag to select AG2:AG4 (the range having the source values).

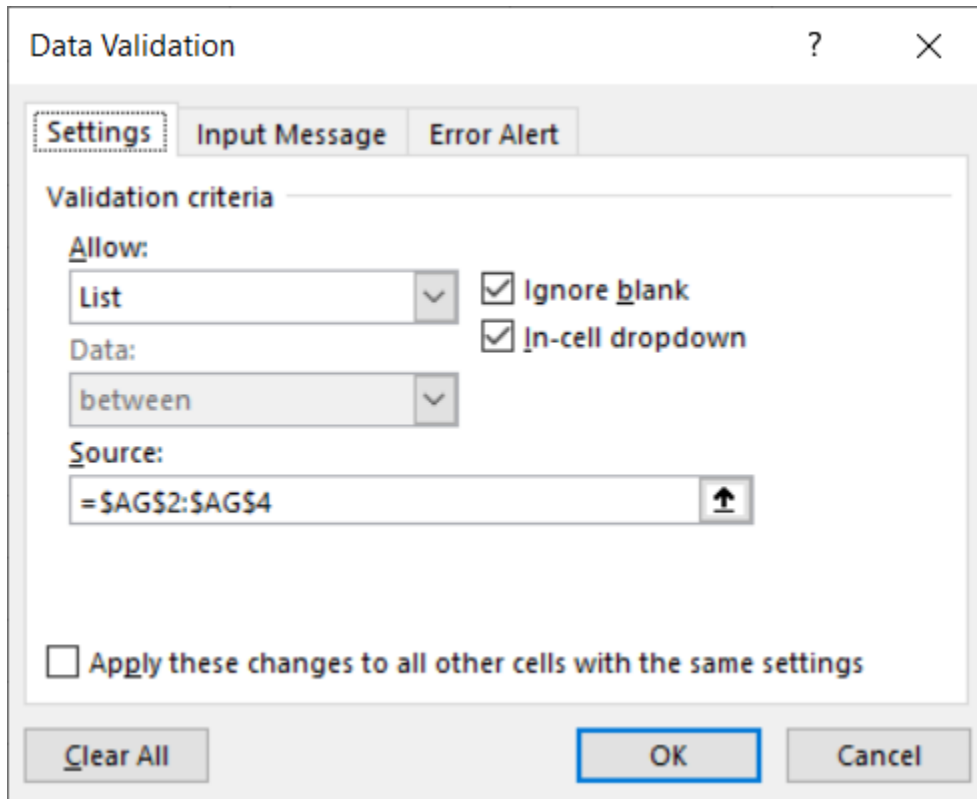


Figure 5- List Values for the first drop list

Click on the "Input Message" Tab ► enter a "Title" > "**Select an option to start search**"

Note: The limit for the title text is 32 characters

In the "Input Message" box, type

"You can initiate your search either by Floor, Suite number or Resident's name"

Hit **OK**

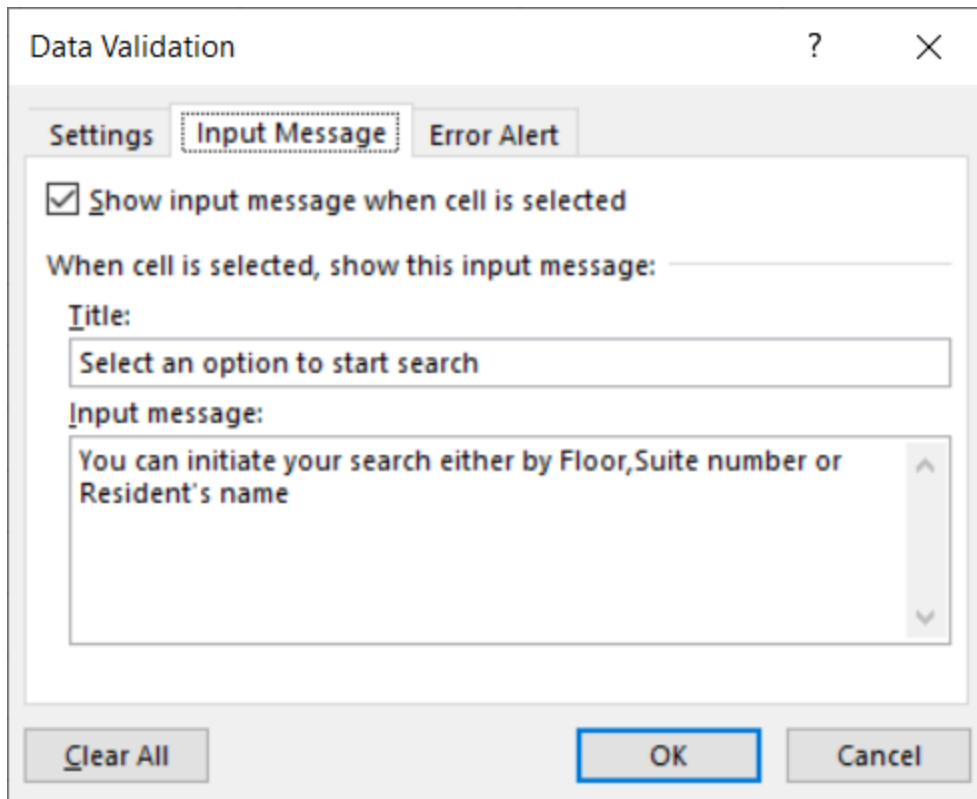


Figure 6- Input message

The first basic drop list is created and looks like this:

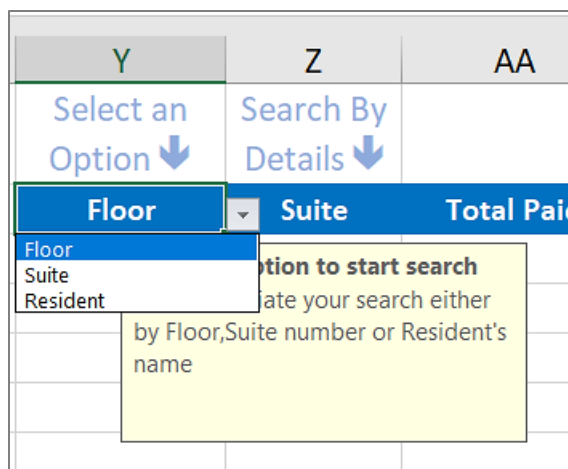


Figure 7- The first drop list in Y2

Creating a "Dependent- Diminishing" drop list

My next step is to replace the static text in cell Z2 with a Dependent Drop list. The contents of this drop list exclude the option you selected in Y2 (e.g. if in Y2 the user selects "Floor" then the drop list in Z2 will exclude "Floor")

Push it to the limit... and beyond

and show only the 2 other options "Suite" and "Resident". This requires creating a **preparation list**

THE PREPARATION LIST

Select cell AH2

in Cell AH2 create a Dynamic Array function (that spills to the cells below)

```
=FILTER(AG2:AG4,COUNTIF(Y2,AG2:AG4)<1)
```

Function Arguments:

=FILTER(array,include,if_empty)

=COUNTIF(range,criteria)

The COUNTIF function looks at cell Y2 (currently "Floor" is selected) and counts the number of the selected option in the range AG2:AG4 if it is less than 1 ► returning a TRUE for the unselected options and False for the selected option.

Then we wrap the COUNTIF function in a FILTER function that returns the 2 unselected options (The TRUE values of the COUNTIF).

AG	AH	AG	AH
	1		2
Floor	FALSE	Floor	Suite
Suite	TRUE	Suite	Resident
Resident	TRUE	Resident	

Figure 8- COUNTIF and FILTER functions

REFERENCING A SPILLED ARRAY

Now, let's create the Dependent Diminishing drop list in cell Z2 referencing the result of the previous function. To reference the result of a Dynamic Array function:

We Select Only the Top Cell (where the function lives)

Followed by a number Sign #

Push it to the limit... and beyond

Note: The number sign is a Spilled array symbol. We may call it pound sign, Hashtag and refers to the entire range returned by the dynamic array function.

Select Z2 Open the Data Validation dialog box (as before)

Note: You can use the shortcut ALT > D > L followed by TAB > L > Tab sequentially

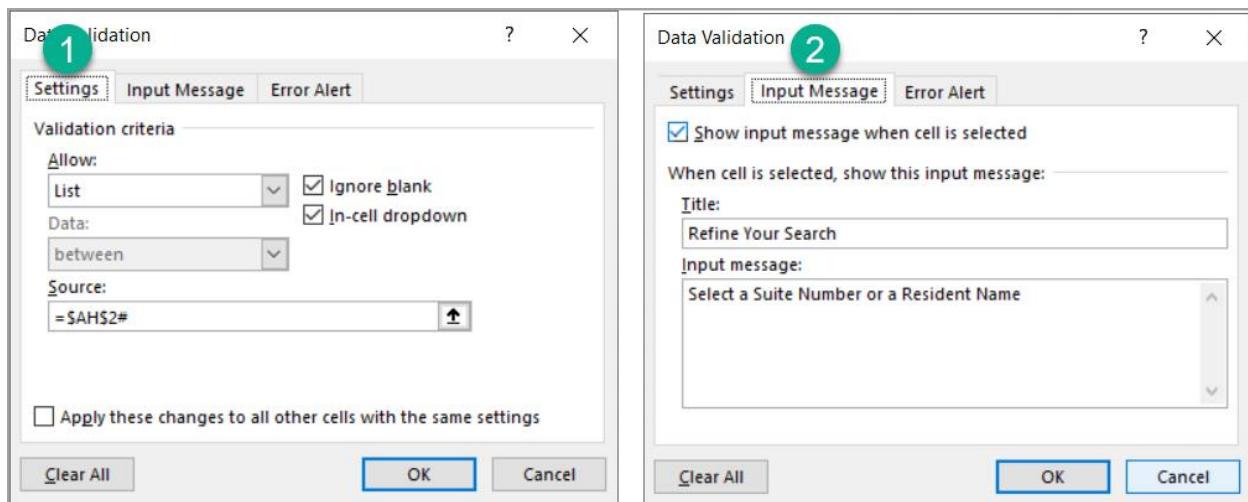


Figure 9- Dependent Data Validation

In the Source box, Select AH2 followed by # for the source (=\$AH\$2#)

On the Input Message tab, enter a Title and Input Message ► Hit OK

Test your second drop list in cell Z2 while changing options from the first drop list in cell Y2.

Y	Z	AA
Select an Option ↓	Search By Details ↓	
Floor	Suite	Total Paid
	Suite Resident	ur Search
		Select a Suite Number or a Resident Name

Figure 10- Testing the second drop list

Preparing for our drop lists with multiple levels of dependencies

Our next step is to prepare for our second level of dependent drop lists in Y3 (3 levels of dependency) and Z3 (6 levels of dependency) by creating **15 Dynamic Array functions** in the preparation area. Remember, dynamic array functions:

- Spill to the adjacent cells (in one single direction).
- If there are obstacles in the spill range, they return a #SPILL! Error.
- They do not require locking cell references (unless you copy them in the other direction after spilling)
- The function lives in one single cell (other spilled cells are greyed out).
- These functions are available in Microsoft 365

Remember the source list stores the **Floor** in column A, the **Suite** number in column B, the **Resident** name in Column C, the **Type** in column D

Push it to the limit... and beyond

	A	B	C	D	E
2	Floor	Suite	Resident	Type	Phone
3	01	0101	K Benedick	Owned	905 600-242
4	01	0102	L Leroux	Owned	905 719-181
5	01	0103	K Hassab	Owned	647 815-300
6	01	0104	U Leedham	Owned	416 773-920
7	01	0105	J Yazdani	Rental	647 269-375

Figure 11- The source list

THREE FUNCTIONS - SAME CONCEPT

Function #1

In cell **AG19** ► Extracts the Floor of the **Owned** suites (excluding the Rental suites)

```
=FILTER(A3:A274,D3:D274="Owned")
```

Function #2

In cell **AH19** ► Extracts the Suite number of the **Owned** suites

```
=FILTER(B3:B274,D3:D274="Owned")
```

Function #3

In cell **AI19** ► Extracts the Resident name of the **Owned** suites

```
=FILTER(C3:C274,D2:D273="Owned")
```

	AF	AG	AH	AI
17		Owned Units		
18		Floor	Suites	Residents
19		01 1	0101 2	L Leroux 3
20		01	0102	K Hassab
21		01	0103	U Leedham
22		01	0104	J Yazdani
23		01	0106	Q Cole
24		01	0107	Q Kosareva
25		01	0108	X Cuenta
26		02	0201	I Jin
27		02	0202	O Nelson
28		02	0203	F Pathak
29		02	0204	S Harding
30		02	0205	E Gretton
31		02	0208	Z Sparkes
32		03	0301	P Wong
33		03	0302	K Sung
34		03	0303	H Rea
35		03	0304	O Menezes
36		03	0305	J Thomas
37		03	0306	T Nolan

Figure 12- Floor/ Suite/ Resident

UNIQUE LIST OF FLOORS

Assuming that we search first by **Floor** in the first drop list in Y2 then, the contents of the drop list below it in Y3 should show a unique list of **Floor**. So, our next preparation function will extract the unique list of Floor twice in Rows #3 & # 13 as follows:

Function #4

Push it to the limit... and beyond

In cell **AM3** ► Extracts a unique list of Floors switched in a single row.

I use the UNIQUE dynamic array function and I wrap it in a Transpose function.

Function arguments:

=UNIQUE(array,by_col,exactly_once)

=TRANSPOSE(array)

The function Spills to the adjacent 34 columns (since there are 35 floors).

=TRANSPOSE(UNIQUE(A3:A274))

Function #5

In cell **AM13** ► Extracts another unique list of Floors switched in a single row.

I use the same exact function as the previous one.

The function Spills to the adjacent 34 columns

=TRANSPOSE(UNIQUE(A3:A274))

	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX
2	Filter Function with 2 Conditions												
3	Floor	01	4 02	03	04	05	06	07	08	09	10	11	12
4	Suite												
5													
6													
7													
8													
9													
10													
11													
12													
13	Floor	01	5 02	03	04	05	06	07	08	09	10	11	12
14	Resident												
15													
16													
17													
18													
19													

Figure 13- Unique List of Floors

CREATING THE THIRD DROP LIST

We can now create our third drop list in cell **Y3** using the available data from the previous 5 functions.

This drop list has three levels of dependency based on which option the user selects from the drop list above: Floor/ Suite/ Resident.

Push it to the limit... and beyond

For each option, we want our drop list to refer to one of the three spilled arrays returned in cells: **AM13/ AH19/ AI19** respectively,

This requires combining the 3 options in a **SWITCH** function.

Function arguments:

`=SWITCH(expression,value1,result1,default_or_value2,result2,...)`

The SWITCH function will evaluate the value in Y2 (the first drop list) and returns one of the following:

If Y2 = Floor ► it returns the Spilled array of AM13# (list of floors)

If Y2 = Suite ► it returns the Spilled array of AH19# (list of Suites)

If Y2 = Resident ► it returns the Spilled array of AI19# (list of Residents)

As a test, we will be creating our **SWITCH** function temporarily in cell **AH13** preceded by a single apostrophe to disable it, as follows:

Function #6

```
=SWITCH($Y$2,"Floor",$AM$3#,"Suite",$AH$19#,"Resident",$AI$19#)
```

Now we can copy this function (*without the single apostrophe*) and paste it in the Data Validation dialog box. Let's do it

Select cell Y3 (for the third drop list) ► Open the Data validation dialog box (as we did before) ► in the Source box, paste the SWITCH function above
► Hit OK

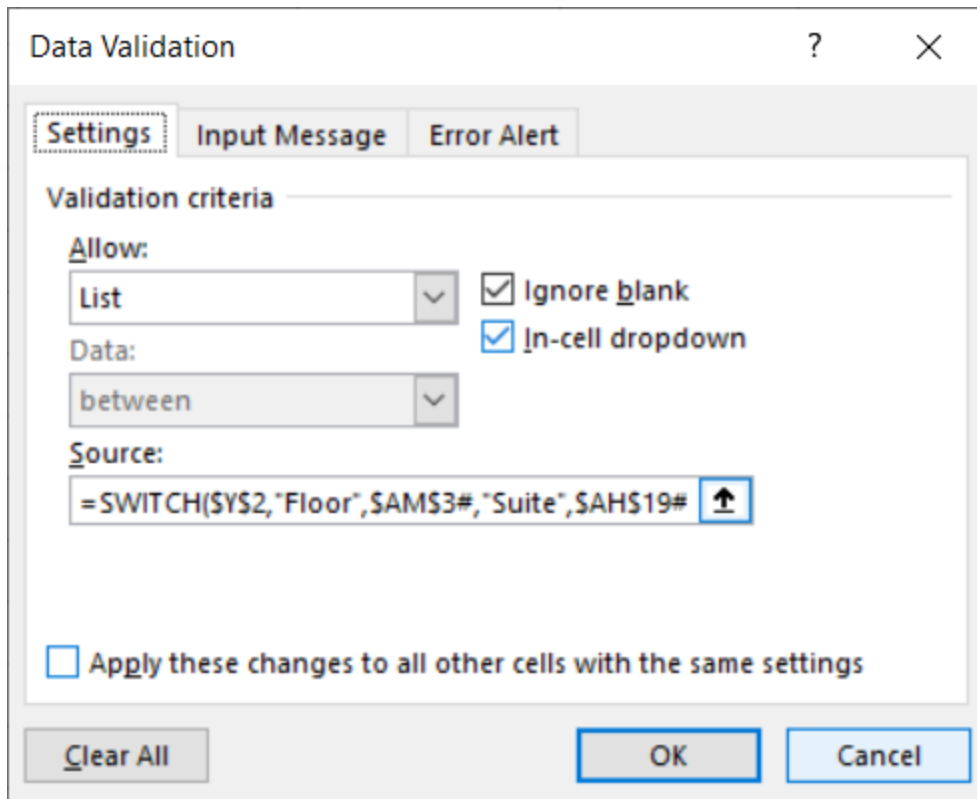


Figure 14- Switch function for 3 options

Now I want to test the contents of the drop list in Y3 three times by changing options in the first drop list in Y2.

SCRATCH THE LIMIT OF DATA VALIDATION IN Z4

Our most robust function, is the one we use for our fourth drop list in cell Z4 as it switches between 6 levels of dependency and it requires a lot more doing.

Push it to the limit... and beyond

	X	Y
2	A	Floor
3		
4		01
5		02
6		03
7		04
8		05

	X	Y
2	B	Suite
3		
4		0101
5		0102
6		0103
7		0104
8		0106

	X	Y
2	C	Resident
3		
4		L Leroux
5		K Hassab
6		U Leedham
7		J Yazdani
8		Q Cole

Figure 15- Three possibilities in Y3 drop list

The drop list we created in Y3 may display either a list of Floors, or a list Suite numbers or a list of Resident names depending on our selection from the drop list above in Y2. Wonderful...

Now we understand the concept of switching dependency in a drop list using the SWITCH function.

LIST OF OWNED SUITES IN ANY FLOOR

In the preparation section, in Cell AM4, we'll create a function that extracts the Suite number of the Owned units in each floor.

Function #7

This function (in **AM4**) will filter the Suites in column B based upon a condition in the Floor (Column A) and another condition in the Type (Column D). Because I want both conditions to be met, I multiply them (that is the equivalent of the AND operator).

My function in **AM4** should read:

```
=FILTER($B$3:$B$274,($A$3:$A$274=AM3)*($D$3:$D$274="Owned"))
```

AM3 is the specific Floor (cell above)

"Owned" is the specific Type we are analyzing.

The Function Spills to the cells below (only 1 direction), returning only the Suite numbers where both conditions are met.

Note that we are locking the ranges to copy in a second direction.

Push it to the limit... and beyond

	AL	AM	AN	AO
2	Filter Function with 2 Conditions			
3	Floor	01	02	03
4	Suite	0101		
5		0102		
6		0103		
7		0104		
8		0106		
9		0107		
10		0108		

Figure 16- Owned Suites in Floor 01

But, because the function cannot spill in two directions, we drag it horizontally all the way to Floor 35 (last Floor) to extract the corresponding values for each Floor.

Note: The number of returned Suites differs in each Floor.

	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU
2	Filter Function with 2 Conditions									
3	Floor	01	02	03	04	05	06	07	08	09
4	Suite	0101	0201	0301	0402	0501	0601	0701	0801	0902
5		0102	0202	0302	0403	0502	0602	0705	0801	0903
6		0103	0203	0303	0404	0503	0603	0706	0801	0904
7		0104	0204	0304	0405	0504	0607	0707	0801	0906
8		0106	0205	0305	0406	0506	0608	0708		0907
9		0107	0208	0306	0407	0508				0908
10		0108		0307	0408					
11				0308						

Figure 17- List of Owned Suites in each floor

Function #8

This function (in **AM14**) will filter the Residents in column C based upon a condition in the Floor (Column A) and another condition in the Type (Column

Push it to the limit... and beyond

D). Because I want both conditions to be met, I multiply them (that's the equivalent of the AND operator).

My function in **AM14** should read:

```
=FILTER($C$3:$C$274,($A$3:$A$274=AM13)*($D$3:$D$274="Owned"))
```

AM13 is the specific Floor (cell above)

"Owned" is the specific Type we are analyzing.

The Function Spills to the cells below, returning only the Suite numbers where both conditions are met.

	AL	AM	AN	AO
12				
13	Floor	01	02	03
14	Resident	K Benedick		
15		L Leroux		
16		K Hassab		
17		U Leedham		
18		K Beckford		
19		Q Cole		
20		Q Kosareva		

Figure 18- Resident names in Owned Suites in Floor 01

Because the function cannot spill in two directions, we drag it horizontally all the way to Floor 35 (last Floor) to extract the corresponding values for each Floor. The number of returned Residents differs in each Floor, it matches the Suites returned by the previous function.

Push it to the limit... and beyond

	AN	AO	AP	AQ	AR	AS	AT	AU
12								
13	02	03	04	05	06	07	08	09
14	X Cuenta	Z Sparkes	I Hance	H Pitamber	W Najam	G Prado	M Galoyan	P Adam
15	I Jin	P Wong	U Chen	U Salah	Q Khaleghy	X Sutherland	G Felipe	Z Adarsh
16	O Nelson	K Sung	U Yang	E Yu	F Koishybay	R Sagoo	S Cernuto	N Alessand
17	F Pathak	H Rea	A Dostmoh	T Noh	P Tran	M Bui	K Graham	D Anees
18	S Harding	O Menezes	O Virk	W Gorji	E Klimov	T Perez		R Anxhela
19	F Thapa	J Thomas	T Bell	Y Jauch				R Bridget
20		T Nolan	Z Kongolo					
21		E Persaud						
22								

Figure 19- Owners list in each Floor

THE SIX POSSIBLE OPTIONS

The next step requires envisioning the different combinations (Six possible options) that a user can select from our first 2 drop lists in cells Y2 & Z2. These combinations are:

1. Floor-Suite
2. Floor-Resident
3. Suite-Floor
4. Suite-Resident
5. Resident-Suite
6. Resident-Floor

For each one of these combinations we will create a dynamic array function that returns the values corresponding to that combination.

Then, we will put them all in a second **SWITCH** function.

Finally, we use the SWITCH function in our last (Fourth) drop list in cell Z3.

We will create the functions in the preparation area in cells AI6 to AI11, and park them with a single apostrophe, to have them ready to use in the powerful SWITCH function later.

Push it to the limit... and beyond

So, Let's create these preparation functions. All of them look at our selection from the drop list in cell Y3.

Function #9

This function will deal with combination # 1: **Floor - Suite**

It will extract the list of Suites for any selected Floor using our preparation functions starting in AM3# Down and to the Right.

This Function will look at the specific Floor in Y3 then pulls the entire list of Suites from the preparation area.

This will be an **XLOOKUP** function that requires 3 arguments:

Lookup Value ► Y3 (the third drop list)

Lookup Array ► The Spilled Array of Floors AM3#

Return Array ► The unequal number of Suites Spilled below the corresponding Floor in the previous Lookup Array. Because they **are unequal** then we will use for this argument an **OFFSET** function.

Function arguments:

`=XLOOKUP(lookup_value,lookup_array,return_array,if_not_found,match_mode,search_mode)`

`=OFFSET(reference,rows,cols,height,width)`

Our Function that we will create for now in **AI6** will read:

`=XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))`

	X	Y	Z	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
1		Select an Option ↓	Search By Details ↓											
2		Floor	Suite			Floor	Suite							
3		02				Suite	Resident							
4						Resident								
5														
6						Floor-Suite	FS	0201						
7						Floor-Resident	FR							
8						Suite-Floor	SF							
9						Suite-Resident	SR							
10						Resident-Suite	RS							
11						Resident-Floor	RF							

Data Preparation Section			
Filter Function with 2 Conditions			
Floor	01	02	03
Suite	0101	0201	0301
	0102	0202	0302
	0103	0203	0303
	0104	0204	0304
	0106	0205	0305
	0107	0208	0306
	0108		0307
			0308

Figure 20- One Suite returned

Push it to the limit... and beyond

This Function when tested returns **ONLY** the first Suite number for the selected Floor in Y3. To extract all the Spilled list of Suites, I add another # sign at the end. My function looks like that:

=XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#

	X	Y	Z	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
1		Select an Option ↓	Search By Details ↓											
2		Floor	Suite			Floor	Suite							
3		02				Suite	Resident							
4														
5														
6						Floor-Suite	FS	0201						
7						Floor-Resident	FR	0202						
8						Suite-Floor	SF	0203						
9						Suite-Resident	SR	0204						
10						Resident-Suite	RS	0205						
11						Resident-Floor	RF	0208						

Data Preparation Section			
Filter Function with 2 Conditions			
Floor	01	02	03
Suite	0101	0201	0301
	0102	0202	0302
	0103	0203	0303
	0104	0204	0304
	0106	0205	0305
	0107	0208	0306
	0108		0307
			0308

Figure 21- List of Owned Suites for a selected Floor

Let's park this function by typing a single apostrophe before the = sign.

Function #10

This function will deal with combination # 2: **Floor - Resident**

We will temporarily create it in **A17**.

It will extract the list of Residents for any selected Floor using our preparation functions starting in AM13# Down and to the Right.

This Function will look at the specific Floor in Y3 then pulls the entire list of Residents from the preparation area.

We could use a similar XLOOKUP function like the one above, but let's learn another technique.

This will be an **OFFSET** function that requires 5 arguments, the first 3 describe a starting point and the next 2 describe the Height and Width of the range to return

Reference ► AM13

Rows ► 1 (We want to move the start point 1 row down)

Push it to the limit... and beyond

Columns ► To move the starting point to the column corresponding to the Floor in Y3. So we use a Match function for this argument. When added to the starting cell in AM13 >> It takes me to the next Floor, so I subtract 1.

My Function that I will create for now in AI7 will read:

```
=OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)
```

It returns ONLY the name of the first resident in the list of residents for the selected Floor.

	X	Y	Z	AE	AF	AG	AH	AI
1		Select an Option	Search By Details					
2		Floor	Resident					
3		02						
4								
5								
6								
7								
8								
9								
10								
11								

	AL	AM	AN	AO
13	Floor	01	02	03
14	Resident	K Benedick	X Cuenta	Z Sparkes
15		L Leroux	I Jin	P Wong
16		K Hassab	O Nelson	K Sung
17		U Leedham	F Pathak	H Rea
18		K Beckford	S Harding	O Menez
19		Q Cole	F Thapa	J Thomas
20		Q Kosareva		T Nolan
21				E Persaud
22				
23				
24				

Figure 22- First Resident name for a selected Floor

To return the entire list of residents, I add another spilled array symbol after the closing bracket of the OFFSET function.

```
=OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#
```

	X	Y	Z	AE	AF	AG	AH	AI
1		Select an Option	Search By Details					
2		Floor	Resident					
3		02						
4								
5								
6								
7								
8								
9								
10								
11								

	AL	AM	AN	AO
13	Floor	01	02	03
14	Resident	K Benedick	X Cuenta	Z Sparkes
15		L Leroux	I Jin	P Wong
16		K Hassab	O Nelson	K Sung
17		U Leedham	F Pathak	H Rea
18		K Beckford	S Harding	O Menez
19		Q Cole	F Thapa	J Thomas
20		Q Kosareva		T Nolan
21				E Persaud
22				
23				
24				

Figure 23- All Resident names of a selected Floor

Let's park this function in AI7 by typing a single apostrophe before the = sign.

Function #11

Push it to the limit... and beyond

This function will deal with combination # 3: **Suite - Floor**

We will temporarily create this function in **A18**

It's a simple XLOOKUP function, that looks at cell Y3 (The drop list), finds a match in the spilled array range AH19#, and returns a corresponding value from the Spilled array in AG19# (in the preparation area). The function reads:

=XLOOKUP(Y3,AH19#,AG19#)

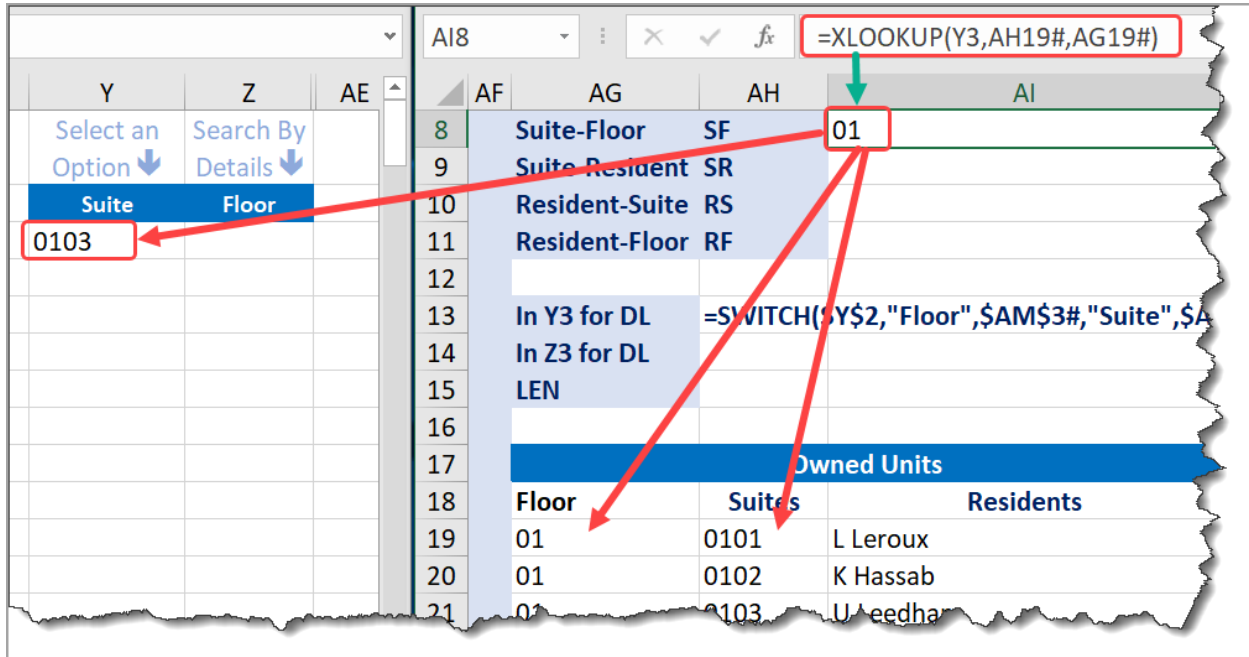


Figure 24- Floor number of a selected Suite

Let's park this function in **A18** by typing a single apostrophe before the = sign.

Function #12

This function will deal with combination # 4: **Suite - Resident**

We will temporarily create this function in **A19**

It's a simple XLOOKUP function, that looks at cell Y3 (The drop list), finds a match in the spilled array range AH19#, and returns a corresponding value from the Spilled array in AI19# (in the preparation area). The function reads:

=XLOOKUP(Y3,AH19#,AI19#)

Push it to the limit... and beyond

The image shows an Excel spreadsheet with a complex XLOOKUP formula in cell A19. The formula is `=XLOOKUP(Y3,AH19#,AI19#)`. The spreadsheet is divided into several sections. On the left, there are columns Y, Z, and AE. Column Y contains a drop-down menu with 'Suite' selected. Column Z contains a drop-down menu with 'Floor' selected. Column AE contains the value '0103'. In the center, there is a table with columns AF, AG, and AH. The table lists various unit types and their corresponding floor numbers. On the right, there is a table with columns AI, AJ, and AK. The table lists residents and their corresponding unit numbers. The formula in A19 is linked to the 'Residents' column of the table on the right.

AF	AG	AH	AI	AJ	AK
8	Suite-Floor	SF	=XLOOKUP(Y3,AH19#,AG19#)		
9	Suite-Resident	SR		U Leedham	
10	Resident-Suite	RS			
11	Resident-Floor	RF			
12					
13	In Y3 for DL	=SWITCH(\$Y\$2,"Floor",\$AM\$3#,"Suite",\$AM\$3#,"Floor")			
14	In Z3 for DL				
15	LEN				
16					
17					
18					
19	Floor	Suites		Residents	
20	01	0101		L Leroux	
21	01	0102		K Hassab	
22	01	0103		U Leedham	
23	01	0104		J Yazdani	
24	01	0106		Q Cole	

Figure 25- Resident name for a selected Suite

Let's park this function in **A18** by typing a single apostrophe before the = sign.

Function #13

This function will deal with combination # 5: **Resident - Suite**

We will temporarily create this function in **A10** (similar to previous)

It's a simple XLOOKUP function, that looks at cell Y3 (The drop list), finds a match in the spilled array range AI19#, and returns a corresponding value from the Spilled array in AH19# (in the preparation area). The function reads:

```
=XLOOKUP(Y3,AI19#,AH19#)
```

Function #14

This function will deal with combination # 6: **Resident - Floor**

We will temporarily create this function in **A11** (similar to previous)

Push it to the limit... and beyond

It's a simple XLOOKUP function, that looks at cell Y3 (The drop list), finds a match in the spilled array range AI19#, and returns a corresponding value from the Spilled array in AG19# (in the preparation area). The function reads:

=XLOOKUP(Y3,AI19#,AG19#)

	X	Y	Z	AA	AB	AF	AG	AH	AI	AJ	AK
1		Select an Option	Search By Details								
2		Floor	Resident	Total Paid	Total Due						
3		06									
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											

Figure 26- The six preparation functions

Here are the 6 preparation functions needed for the fourth drop list. To show all of them I split the window and they are all temporarily parked in column AI from row 6 to 11. We will be later combining them together in a SWITCH function inside the data Validation dialog box.

Problem with the SWITCH function in the Data Validation box:

The SWITCH function is intended to return one of SIX different ranges for the drop list in Z3 based upon the 6 possible combinations in Y2 & Z2. So it should originally read like this:

```
=SWITCH(Y2&"-"&Z2," Floor-Suite",
XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#, "Floor-Resident",
OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#,"Suite-Floor",
XLOOKUP(Y3,AH19#,AG19#),"Suite-Resident",
XLOOKUP(Y3,AH19#,AI19#), "Resident-Suite",
XLOOKUP(Y3,AI19#,AH19#),"Resident-Floor",
XLOOKUP(Y3,AI19#,AG19#))
```

The LEN (length) of this function is **281 characters**, while the maximum limit of any function in the Data Validation dialog box is **255 Characters**. So we need to create a shorter form of the SWITCH function to be able to use it in a drop list.

Push it to the limit... and beyond

PREPARING FOR THE SWITCH FUNCTION:

We abbreviated in Column AH from AH6 to AH11 all the six possible combinations from the drop lists Y2 & Z2, as follows:

Floor-Suite	FS
Floor-Resident	FR
Suite-Floor	SF
Suite-Resident	SR
Resident-Suite	RS
Resident-Floor	RF

In cell AA1, I create a **VLOOKUP** function that returns the abbreviation based upon the selection from Both Y2 & Z2.

Function arguments:

`=VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)`

The VLOOKUP function reads:

`=VLOOKUP(Y2&"-"&Z2,AG6:AH11,2,0)`

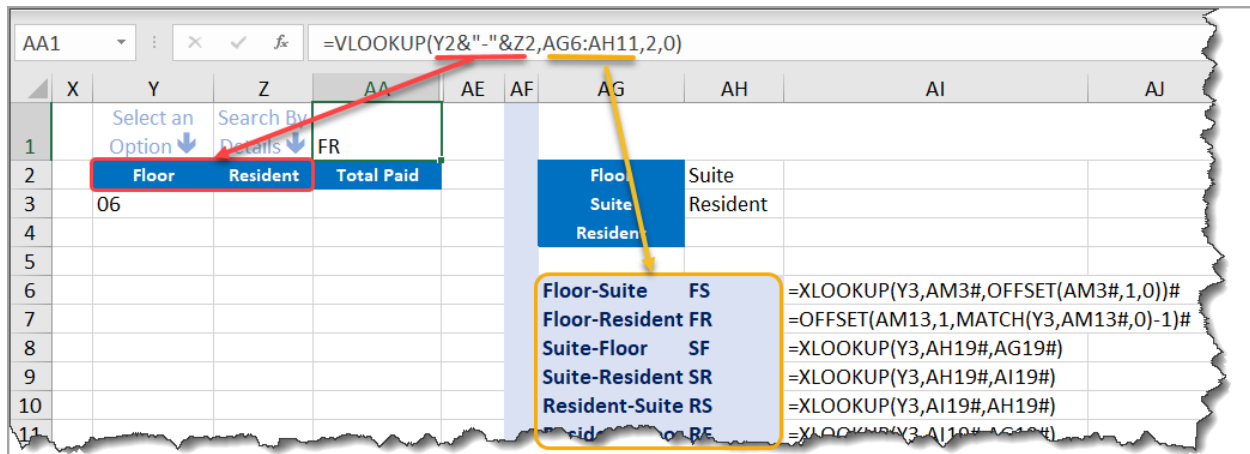


Figure 27- Combine selections from the 2 drop lists

Now we can rebuild our SWITCH function using the abbreviation returned in AA1 Rather than the long names in Y2 & Z2.

We will temporarily build our SWITCH function in cell **AH14** and test it (►#Spill! Error because we have cell contents below), measure it's length in AH15 ►now it becomes 209 character instead of the long format of 281

Push it to the limit... and beyond

characters. Park it with a single apostrophe at the beginning. The function in **AH14** reads:

```
=SWITCH(AA1,"FS",XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#,  
"FR",OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#,  
"SF",XLOOKUP(Y3,AH19#,AG19#),  
"SR",XLOOKUP(Y3,AH19#,AI19#),  
"RS",XLOOKUP(Y3,AI19#,AH19#),  
"RF",XLOOKUP(Y3,AI19#,AG19#))
```

We are ready to create our last drop list in cell Z3 using the shortened SWITCH function (now 209 characters)

Select cell Z3

Open the Data Validation dialog box ► Select "List"

In the source box, paste the new SWITCH function.

Make a selection from Y2, Z2 and Y3 ► then Test the 6 options for Z3

We just touched the limit of what data validation can do with 6 levels of dependencies. Let's explore them:

Option #1

Floor-Suite ► **FS**

```
=SWITCH(AA1,"FS",XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#,  
"FR",OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#,  
"SF",XLOOKUP(Y3,AH19#,AG19#),"SR",XLOOKUP(Y3,AH19#,AI19#),"RS",  
XLOOKUP(Y3,AI19#,AH19#),"RF",XLOOKUP(Y3,AI19#,AG19#))
```

Push it to the limit... and beyond

1	X	Y	Z	AA
1		Select an Option ↓	Search By Details ↓	FS
2		Floor	Suite	Total Paid
3		03		
4			0301	
5			0302	
6			0303	
7			0304	
8			0305	

Figure 28- Option1 Floor- Suite

Option #2

Floor-Resident ► **FR**

=SWITCH(AA1,"FS",XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#,
 "FR",OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#,
 "SF",XLOOKUP(Y3,AH19#,AG19#),"SR",XLOOKUP(Y3,AH19#,AI19#),"RS",
 XLOOKUP(Y3,AI19#,AH19#),"RF",XLOOKUP(Y3,AI19#,AG19#))

2	X	Y	Z	AA
1		Select an Option ↓	Search By Details ↓	FR
2		Floor	Resident	Total Paid
3		03		
4			Z Sparkes	
5			P Wong	
6			K Sung	
7			H Rea	
8			O Menezes	

Figure 29- Option2 Floor- Resident

Option #3

Suite-Floor ► **SF**

=SWITCH(AA1,"FS",XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#,"FR",OFF
 SET(AM13,1,MATCH(Y3,AM13#,0)-1)#,

Push it to the limit... and beyond

"SF",XLOOKUP(Y3,AH19#,AG19#),"SR",XLOOKUP(Y3,AH19#,AI19#),"RS",XLOOKUP(Y3,AI19#,AH19#),"RF",XLOOKUP(Y3,AI19#,AG19#))

3	X	Y	Z	AA
1		Select an Option ↓	Search By Details ↓	SF
2		Suite	Floor	Total Paid
3		0106		
4			01	
5				

Figure 30- Option3 Suite-Floor

Option #4

Suite-Resident ► SR

=SWITCH(AA1,"FS",XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#,"FR",OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#,"SF",XLOOKUP(Y3,AH19#,AG19#),"SR",XLOOKUP(Y3,AH19#,AI19#),"RS",XLOOKUP(Y3,AI19#,AH19#),"RF",XLOOKUP(Y3,AI19#,AG19#))

4	X	Y	Z	AA
1		Select an Option ↓	Search By Details ↓	SR
2		Suite	Resident	Total Paid
3		0106		
4			Q Cole	
5				

Figure 31- Option4 Suite-Resident

Option #5

Resident-Suite ► RS

=SWITCH(AA1,"FS",XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#,"FR",OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#,"SF",XLOOKUP(Y3,AH19#,AG19#),"SR",XLOOKUP(Y3,AH19#,AI19#),"RS",XLOOKUP(Y3,AI19#,AH19#),"RF",XLOOKUP(Y3,AI19#,AG19#))

5	X	Y	Z	AA
1		Select an Option ↓	Search By Details ↓	RS
2		Resident	Suite	Total Paid
3		Q Kosareva		
4			0107	
5				

Figure 32- Option5 Resident- Suite

Option #6

Resident-Floor ► RF

=SWITCH(AA1,"FS",XLOOKUP(Y3,AM3#,OFFSET(AM3#,1,0))#,"FR",OFFSET(AM13,1,MATCH(Y3,AM13#,0)-1)#,"SF",XLOOKUP(Y3,AH19#,AG19#),"SR",XLOOKUP(Y3,AH19#,AI19#),"RS",XLOOKUP(Y3,AI19#,AH19#),"RF",XLOOKUP(Y3,AI19#,AG19#))

6	X	Y	Z	AA
1		Select an Option ↓	Search By Details ↓	RF
2		Resident	Floor	Total Paid
3		Q Kosareva		
4			01	
5				

Figure 33- Option6 Resident- Floor

HIDE CONTENTS OF CELL AA1

Since Cell AA1 is just providing a Value for our SWTCH function we want to hide it from the view by changing the Font color to White.

Resetting 2 drop lists (Y3 & Z3) with a code

Every time we make a change in any drop list in Y2 or Z2, we want the previously selected value in Y3 & Z3 to be deleted in order to make a new fresh selection in these cells. This requires creating a "Change Event" code for the worksheet as follows:

OPEN THE VBE

Right click the sheet Tab and select ► View Code

Push it to the limit... and beyond

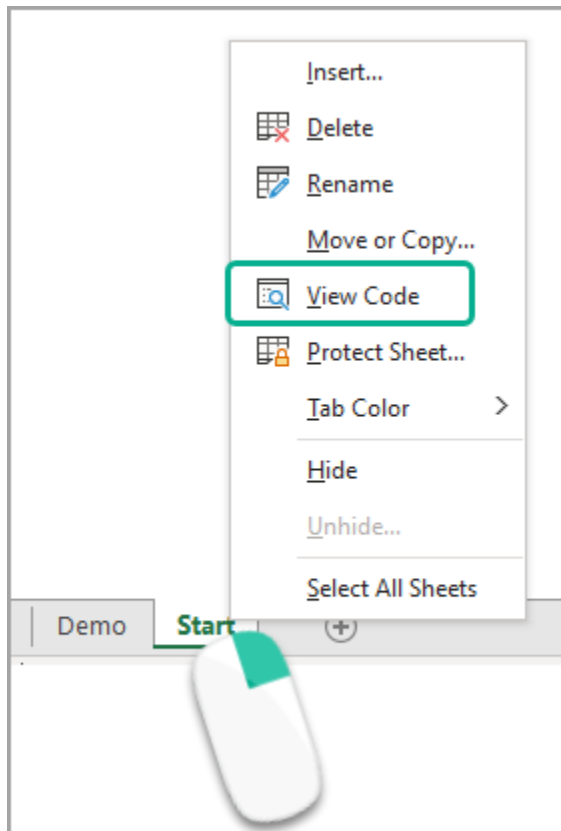


Figure 34- Switching to VBE

The Visual Basic Editor opens. There are 2 drop lists at the top. Click on the Left drop List and select ► Worksheet

Then Click on the Right drop list and select ► Change

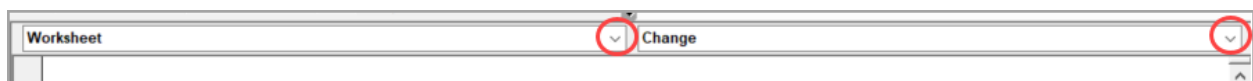


Figure 35- Select the Change Event

Keep only the Worksheet Change event and delete the Worksheet Selection Change event

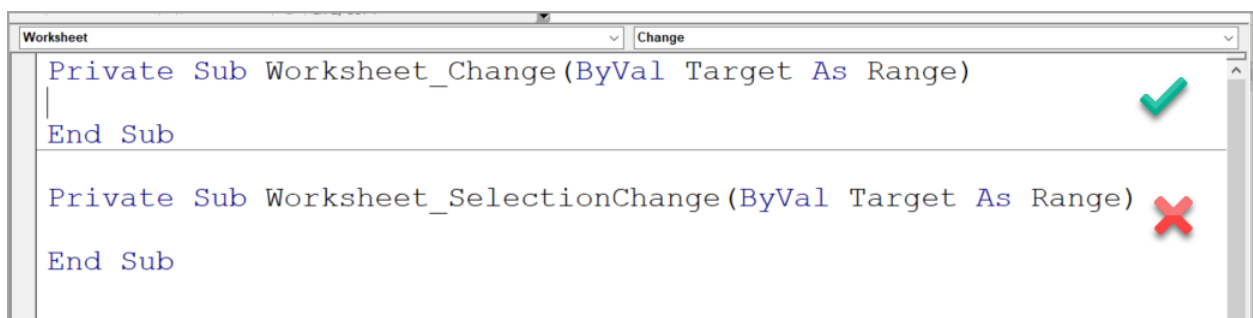


Figure 36-Delete the Selection Change Event

WRITING THE CODE

Between the Private Sub and the End Sub lines, copy and paste this code:

```
If Not Intersect(Target, Range("Y2:Z2")) Is Nothing Then  
    Range("Y3:Z3").ClearContents  
End If
```

This code means:

If you change your selection from either drop lists in Y2 or Z2 ► Clear the contents of Y3 and Z3

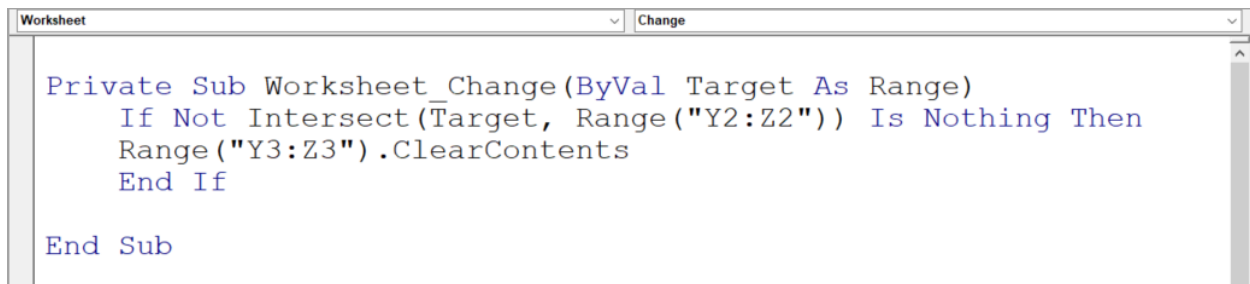


Figure 37- The Change event code

Close the Visual Basic editor to go back to Excel and test. Although the values in Y3 and Z3 are deleted with any change in Y2 or Z2, however the drop lists in Y3 and Z3 are fully functional.

Note: Your Excel file must be stored in a .XLSM format.

Getting Information based upon selection

Our next step will be extracting the corresponding: **Total Paid**, **Total Due**, **Past Due** and **Over Payment** for which ever option we select (6 combinations) from Y3 and Z3.

Note: we will use nested VLOOKUP functions to do that

Function arguments:

=VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)

We will be creating 2 VLOOKUP functions.

Push it to the limit... and beyond

To avoid confusion, you can create each VLOOKUP function separately and copy it to the Clipboard.

Also Note the Column Numbers from which we will extract the return values (Column Index Number argument of the VLOOKUP)

To show the column numbers instead of the default column letters:

File ► Options ► Formulas ► check R1C1 reference

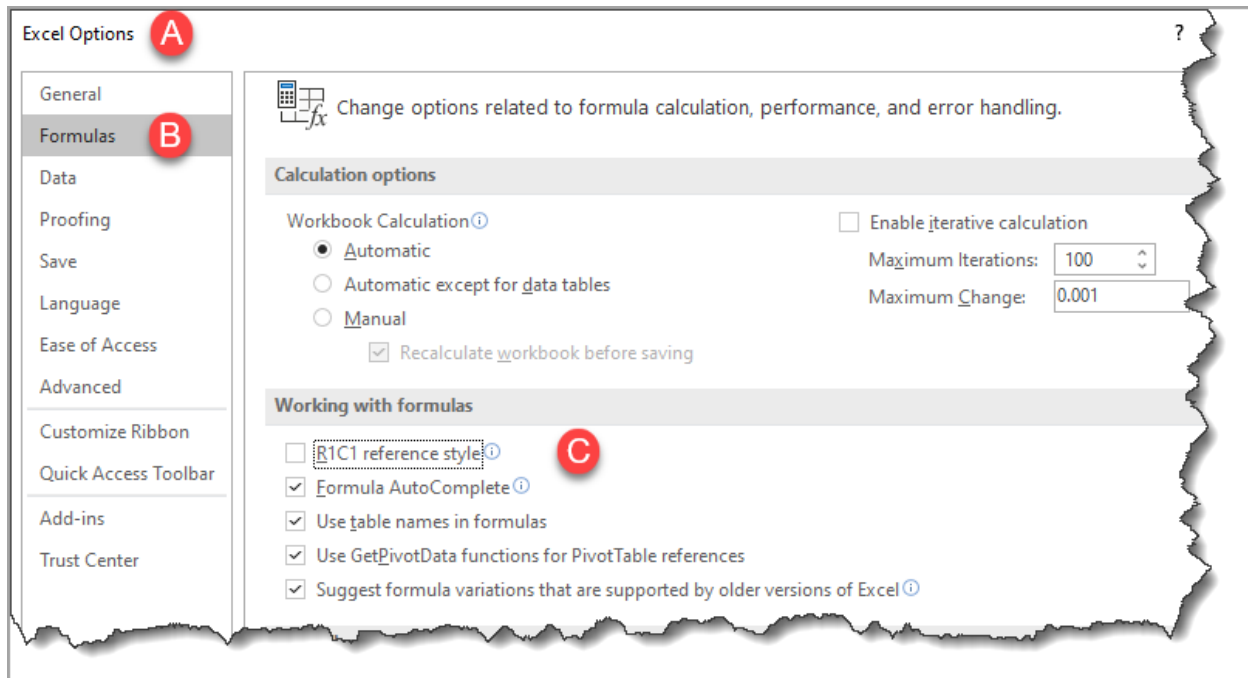


Figure 38- The R1C1 reference

To make it easier for you, we will hide columns 5 to 19 since we are not extracting values from any of these columns and we will bring back the column letters (File ► Options ► Formulas ► uncheck R1C1 reference)

Push it to the limit... and beyond

	1	2	3	4	20	21	22	23	24
2	Floor	Suite	Resident	Type	Total Paid	Total Due	Past Due	Over PMT	
3	01	0101	K Benedick	Owned	\$11,917.00	\$15,339.00	\$3,422.00	\$0.00	
4	01	0102	L Leroux	Owned	\$12,215.00	\$16,296.00	\$4,081.00	\$0.00	
5	01	0103	K Hassab	Owned	\$11,549.33	\$11,526.00	\$0.00	\$23.33	
6	01	0104	U Leedham	Owned	\$11,573.00	\$10,201.00	\$0.00	\$1,372.00	
7	01	0105	J Yazdani	Rental	\$11,575.00	\$11,575.00	\$0.00	\$0.00	

	A	B	C	D	T	U	V	W	X
2	Floor	Suite	Resident	Type	Total Paid	Total Due	Past Due	Over PMT	
3	01	0101	K Benedick	Owned	\$11,917.00	\$15,339.00	\$3,422.00	\$0.00	
4	01	0102	L Leroux	Owned	\$12,215.00	\$16,296.00	\$4,081.00	\$0.00	
5	01	0103	K Hassab	Owned	\$11,549.33	\$11,526.00	\$0.00	\$23.33	
6	01	0104	U Leedham	Owned	\$11,573.00	\$10,201.00	\$0.00	\$1,372.00	
7	01	0105	J Yazdani	Rental	\$11,575.00	\$11,575.00	\$0.00	\$0.00	

Figure 39- Compare Column Letters and Numbers

FIRST VLOOKUP

Our first VLOOKUP function will extract values by Suite number:

- The Lookup Column is column B
- The Lookup Array is: B3:W274
- The column index number will be **19, 20, 21, 22**

Since I create a single function (in cell AA3 under Total Paid) and drag it to the right, I want the column index number to increment as I drag the function, I do that by using a **COLUMNS** function that refers to any cell twice, lock the first reference only, then add 18

Note in this array starting from Column B ► The Total Paid is column 19

So, the Column Index argument of this VLOOKUP will look like this:

COLUMNS(\$X\$1:X1)+18

And the VLOOKUP function will look like this:

=VLOOKUP(\$Z\$3,\$B\$3:\$W\$274,COLUMNS(\$X\$1:X1)+18,0)

Push it to the limit... and beyond

	Y	Z	AA	AB	AC	AD	AE
	Floor	Suite	Total Paid	Total Due	Past Due	Over PMT	
	01	0103	=VLOOKUP(\$Z\$3,\$B\$3:\$W\$274,COLUMNS(\$X\$1:X1)+18,0)				
		A					

Figure 40- Testing the first VLOOKUP

What if the user searches by Resident name instead of by Suite? Then, the VLOOKUP function is different:

SECOND VLOOKUP

Our second VLOOKUP function will extract values by Resident

- The Lookup Column is column C (Resident)
- The Lookup Array is: C3:W274
- The column index number will be **18, 19, 20, 21**

Since I create a single function (in cell AA3 under Total Paid) and drag it to the right, I want the column index number to increment as I drag the function, I do that by using a COLUMNS function that refers to any cell twice, lock the first reference only, then add 17

Note in this array starting from Column C ► The Total Paid is column 18

So, the Column Index argument of this VLOOKUP will look like this:

COLUMNS(\$X\$1:X1)+17

And the VLOOKUP function will look like this:

=VLOOKUP(\$Z\$3,\$C\$3:\$W\$274,COLUMNS(\$X\$1:X1)+17,0)

	Y	Z	AA	AB	AC	AD	AE
	Floor	Resident	Total Paid	Total Due	Past Due	Over PMT	
	01	L Leroux	=VLOOKUP(\$Z\$3,\$C\$3:\$W\$274,COLUMNS(\$X\$1:X1)+17,0)				
		B					

Figure 41- Testing the second VLOOKUP

To avoid confusion, you can create each VLOOKUP function separately and copy it to the Clipboard.

COMBINING THE 2 VLOOKUP FUNCTIONS

Now I want to combine both functions together to include both options (whether the user searches by Suite or by Resident) so I will wrap them together in an **IFERROR** function, pasting them from the clipboard.

My combined function will look like this:

```
=IFERROR(VLOOKUP($Z$3,$B$3:$W$274,COLUMNS($X$1:X1)+18,0),VLOOKUP($Z$3,$C$3:$W$274,COLUMNS($X$1:X1)+17,0))
```

A LITTLE ISSUE TO AVOID

Finally, If the user selects either a Suite or Resident in Y3 and a **Floor** in Z3 ► Our VLOOKUP function will return a **#N/A error**. Because Z3 is always our lookup value, we do not want to search by Floor in this cell. But if it happens, we also do not want to see the #N/A error.

A SECOND IFERROR FUNCTION

To do that we will wrap the previous function in a second **IFERROR** function that returns a blank.

Our final function should look like this:

```
=IFERROR(IFERROR(VLOOKUP($Z$3,$B$3:$W$274,COLUMNS($X$1:X1)+18,0),VLOOKUP($Z$3,$C$3:$W$274,COLUMNS($X$1:X1)+17,0)), "")
```

AN INPUT MESSAGE

To make it practical, add an input message to the drop list in cell Z3 saying:

Select either a Suite number or a Resident name

Copy the Final function in AA3 to the right to AD3

Push it to the limit... and beyond

X	Y	Z	AA	AB	AC	AD
	Suite	Resident	Total Paid	Total Due	Past Due	Over PMT
	0103	U Leedham	\$11,573	\$10,201	\$0	\$1,372

Refine Selection
Select either a Suite
number or a Resident
name

Figure 42- Input Message in Z3

Finalizing the project

TWO CONDITIONAL FORMATTING RULES

To clearly spot any amount past due or any overpayment, we will be applying Conditional Formatting for the Amount **Past Due** and **Over PMT**

Select Cell AC3

Home Tab ► Conditional Formatting ► Greater Than.. ► Type 0 and keep the red color

X	Y	Z	AA	AB	AC	AD
	Suite	Resident	Total Paid	Total Due	Past Due	Over PMT
	0103	U Leedham	\$11,573	\$10,201	\$0	\$1,372

Greater Than
Format cells that are GREATER THAN:
0 with Light Red Fill with Dark Red Text
OK Cancel

Figure 43- Past Due conditional formatting

Select Cell AD3

Home Tab ► Conditional Formatting ► Greater Than.. ► Type 0 and change the color to Green

Push it to the limit... and beyond

X	Y	Z	AA	AB	AC	AD
	Suite	Resident	Total Paid	Total Due	Past Due	Over PMT
	0103	U Leedham	\$11,573	\$10,201	\$0	\$1,372

Greater Than

Format cells that are GREATER THAN:

with Green Fill with Dark Green Text

OK Cancel

Figure 44- Over PMT conditional formatting

Test again by selecting different options in Z3

Our project is finished!

But we have a lot of numbers and calculations (in the preparation area) so we want to make our sheet much more user friendly.

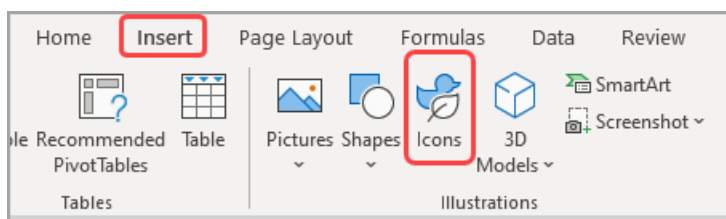
HIDING COLUMNS

Select then ► Hide Columns from D to S

Select then ► Hide Columns from AG to BT

ZOOM IN AND OUT

Insert 2 icons (Insert Tab ► Icons ► Search for "Zoom" ► select the 2 magnifying lenses



Resize and position each icon to fit on top of cells AA1 and AB1. Change the color however you like (Graphics Format ► Graphics Fill). Flip one of them Horizontally

They will look like this:

Push it to the limit... and beyond



X	Y	Z	AA	AB	AC	AD	
	Select an Option ↓	Search By Details ↓					
	Suite	Resident	Total Paid	Total Due	Past Due	Over PMT	
	0103	U Leedham	\$11,573	\$10,201	\$0	\$1,372	

Figure 45- Resize and reposition icons

These 2 icons will be used to **Zoom In** and **Zoom Out** respectively on the range from Y1 to AD3, for now they are not functioning.

CREATE 2 MACROS FOR THE ZOOMING FUNCTIONALITY

Switch to the Visual Basic editor by hitting **ALT + F11**

Insert a Module (Blank Space): Insert Menu ► Module

Note: You can use the shortcut **ALT > I > M**

Click in the white space

Copy and paste these two codes

```
Sub RangeZoom()
    Sheets("Start").Range("Y1:AE3").Select
    ActiveWindow.Zoom = True
    Range("Y1").Select
End Sub
```

```
Sub UnZoom()
    Sheets("Start").Range("Y1:AE3").Select
    ActiveWindow.Zoom = False
    Range("Y1").Select
End Sub
```

Close the visual Basic Editor ► Back to Excel

ASSIGNING MACROS

Select the magnifying icon for zooming in (plus sign) ► Right click it ► Assign macro.

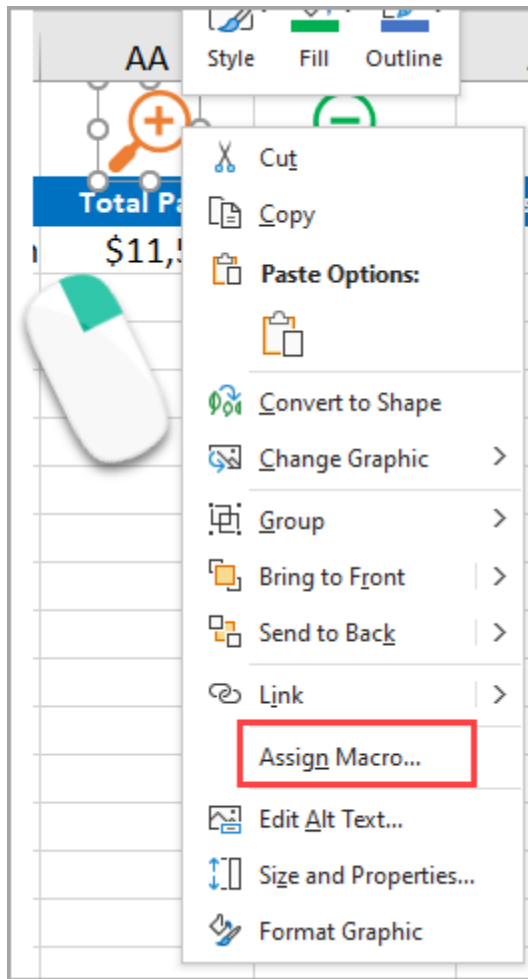


Figure 46- Assign a macro to Zoom In

The Assign Macro dialog box opens ► select **RangeZoom** macro ► Hit OK.

Every time you click on this icon ► the range Y1: AE3 will fill the entire screen.

Push it to the limit... and beyond

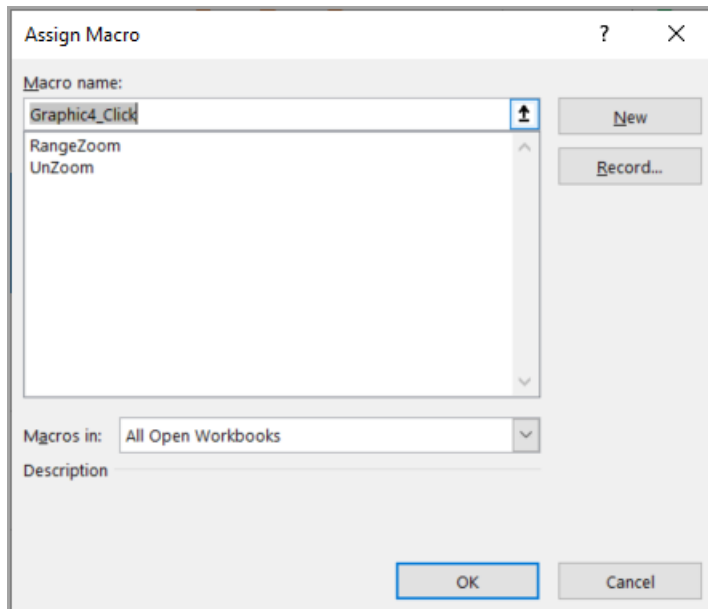


Figure 47- The Assign Macro dialog box

Select the magnifying icon for zooming Out (minus sign) ► Right click it ► Assign macro.

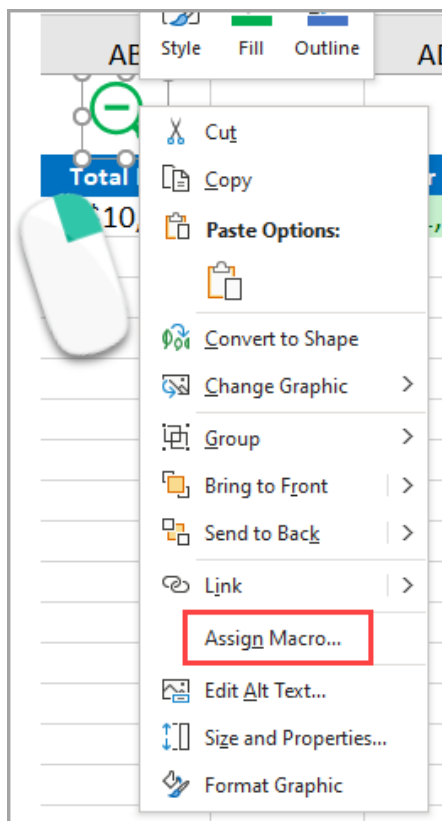


Figure 48- Assigning a zoom out macro

The Assign Macro dialog box opens ► select **UnZoom** macro ► Hit OK.

Push it to the limit... and beyond

Every time you click on this icon ► the range Y1: AE3 will zoom out to 100%

Test Zooming in and Out.

Test all your drop lists with different combinations

Test the extraction of records and conditional formatting.

Congratulations the project is now complete! 😊