BIMPURE

presents

PAMPHLETS

by: Nicolas Catellier, Architect

ISSUE #30

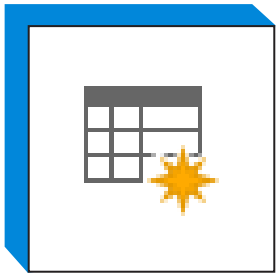
EXCEL

REVITPURE.COM

PAMPHLETS COLLECTION

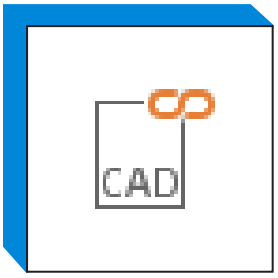
- BASICS / MODEL
- DESIGN / PRESENT
- ORGANIZE / AUTOMATE
- CULTURE / TECH

ISSUE #1
WORKSETS

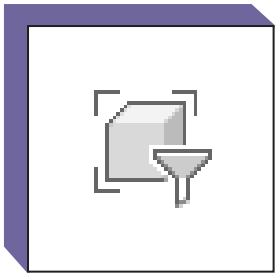


ISSUE #2
SCHEDULE

ISSUE #3
PHASES

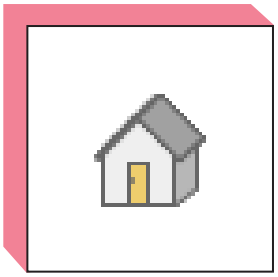


ISSUE #4
LINK CAD

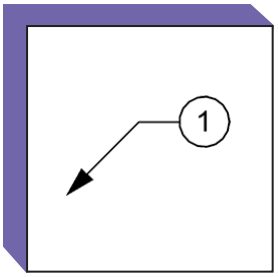


ISSUE #5
FILTERS

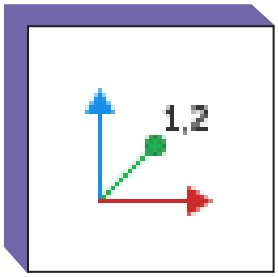
ISSUE #6
VIRTUAL REALITY



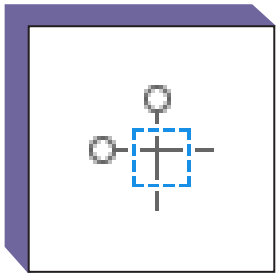
ISSUE #7
3D VIEWS



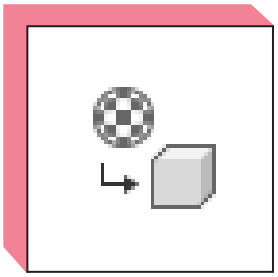
ISSUE #8
PLAN NOTES



ISSUE #9
COORDINATES



ISSUE #10
SCOPE BOXES

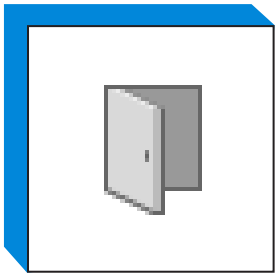


ISSUE #11
FINISHES

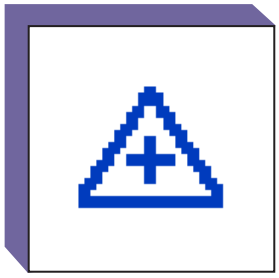
ISSUE #12
LINE WEIGHTS

PAMPHLETS COLLECTION

- BASICS / MODEL
- DESIGN / PRESENT
- ORGANIZE / AUTOMATE
- CULTURE / TECH



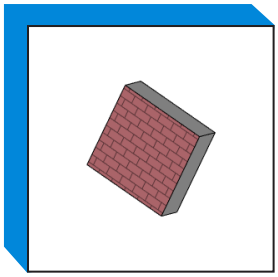
ISSUE #13
DOORS



ISSUE #14
SHARED SITE

ISSUE #15
PYREVIT

ISSUE #16
ADD-INS



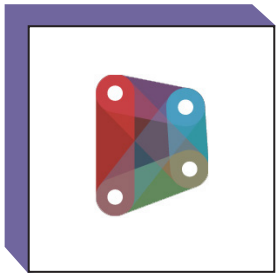
ISSUE #17
SLANTED WALLS

ISSUE #18
RAILINGS PART 1

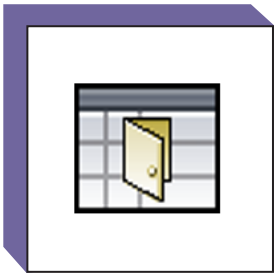
ISSUE #19
RAILINGS PART 2

ISSUE #20
PHASES PART 2

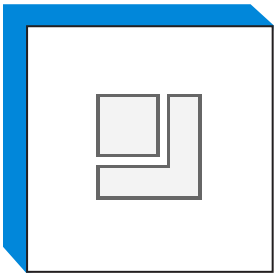
ISSUE #21
LANDSCAPE



ISSUE #22
DYNAMO



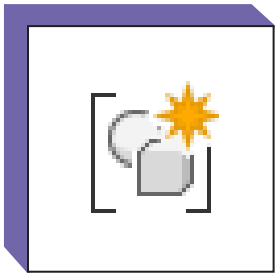
ISSUE #23
DOOR SCHEDULES



ISSUE #24
FAMILIES

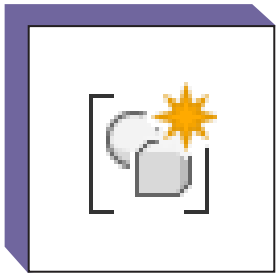
PAMPHLETS COLLECTION

- BASICS / MODEL
- DESIGN / PRESENT
- ORGANIZE / AUTOMATE
- CULTURE / TECH



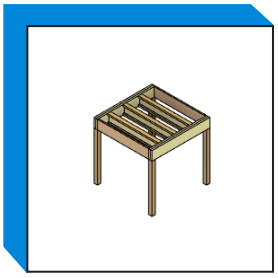
ISSUE #25

GROUPS PART 1



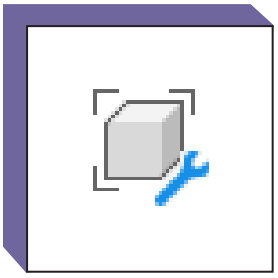
ISSUE #26

GROUPS PART 2



ISSUE #27

NESTED STRUCTURE

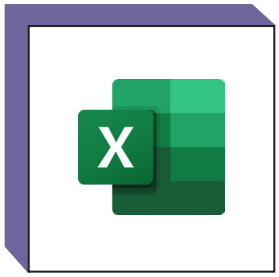


ISSUE #28

FAMILY VISIBILITY

ISSUE #29

AUTO NUMBERING



ISSUE #30

EXCEL

WHAT IS THIS "PAMPHLET"?

Revit Pure Pamphlets are published 4 times a year by email. Each edition covers a particular Revit theme. We like to pick themes that are complex and confusing. Our job is to make these topics simple for you.

ABOUT EXCEL

This pamphlet is about using Excel alongside Revit. First, you will learn about a few plugins that can be used to import and export Excel information.

The core of this pamphlet is about using Dynamo for advanced Excel interoperability with Revit.

Plugins are great, but for complete control, Dynamo is better.

You will learn how to create a list of rooms based on Excel. Then, you will learn how to push data from Excel to Revit.

Finally, you will learn how to use the amazing free Dynamo Multiplayer to batch and schedule Dynamo scripts, including the Excel to Revit workflow.

Click here to download the Dynamo scripts mentioned in the PDF.

Good luck.

3 PLUGINS TO LINK REVIT TO EXCEL

#1 - DiRoots SheetLink (part of DiRoots One)

If you want to get quick results from Excel to DiRoots, this is probably your best bet. The ideal workflow is to export to Excel, make some changes and import back.

Cost: Free
Link

#2 - Ideate BIM Link

If you want to go deeper than DiRoots, your best bet is Ideate BIM Link. It can be combined with Automation to schedule tasks.

Cost: ±$1000 per year/user
Link

#3 - Speckle

If you are using Excel not only with Revit, but with other tools such as Rhino, this can be a great way to easily spread the data. Speckle is a web-based data connector.

Cost: Free
Link

LINK EXCEL TO REVIT WITH DYNAMO

Using plugins is nice for casual workflows. But if you have complex workflows that needs to executed multiple times a week, a Dynamo script is your best bet.

New to Dynamo? Check out issue 22 to get started.

In this issue, we'll work with this assumption:

- Revit users are creating a lot of rooms.
- Information in rooms is sometimes entered in Excel by project managers who don't use Revit.
- Revit users want to use information to model, but also verify if the data is the same in Revit and Excel.

CREATE ROOMS FROM EXCEL LIST

This is the Excel spreadsheet created by the project manager.

	A	B	C	D
1	**Room Name**	**Room Number**	**Department**	**Area Required**
2	Bob Office	100	Office	13
3	Gina Office	101	Office	13
4	Mary Office	102	Office	13
5	Jimmy Office	103	Office	13
6	Sue Office	104	CEO	18

Now, let's say you don't want to manually create all these rooms. Here are all the steps you need to automate the process in Revit.

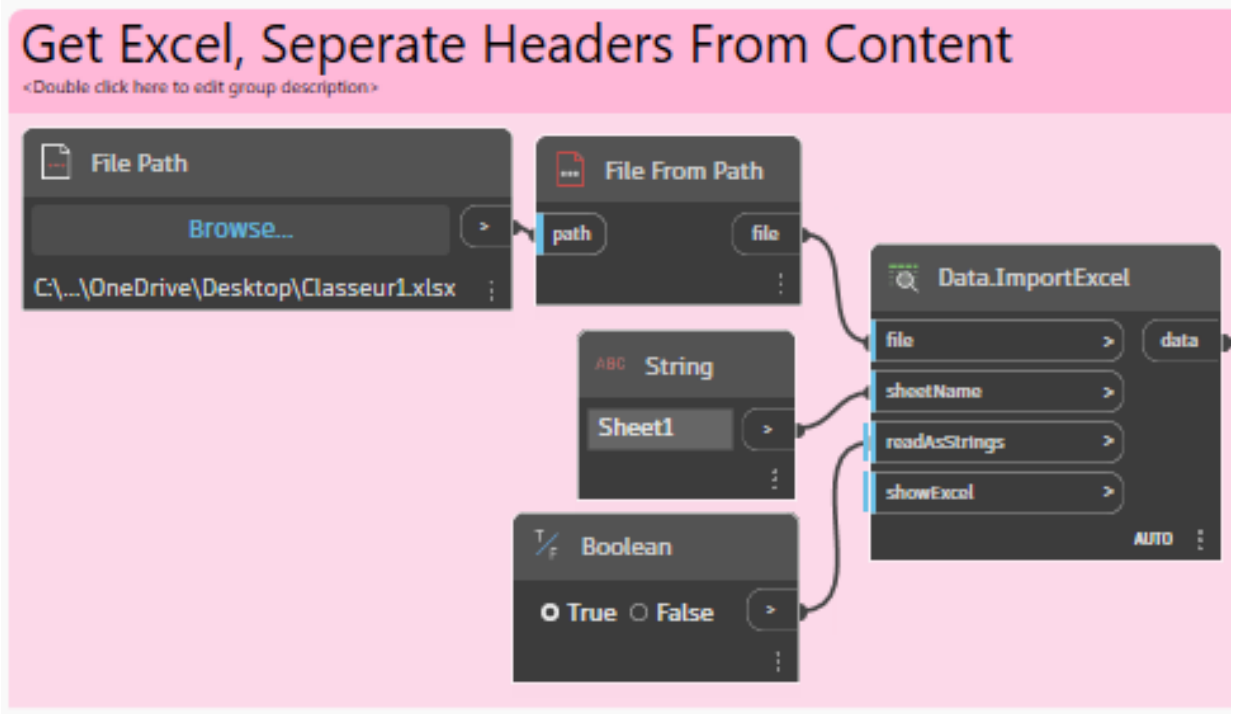
1- CREATE AREA REQUIRED SHARED PARAMETER

First, add a new shared project parameter called: **Area Required**. This should be an area type of parameter assigned to rooms. That will allow you to verify if the minimum area of each room is met.

Analysis Results	
Area Required	13.000 m^2

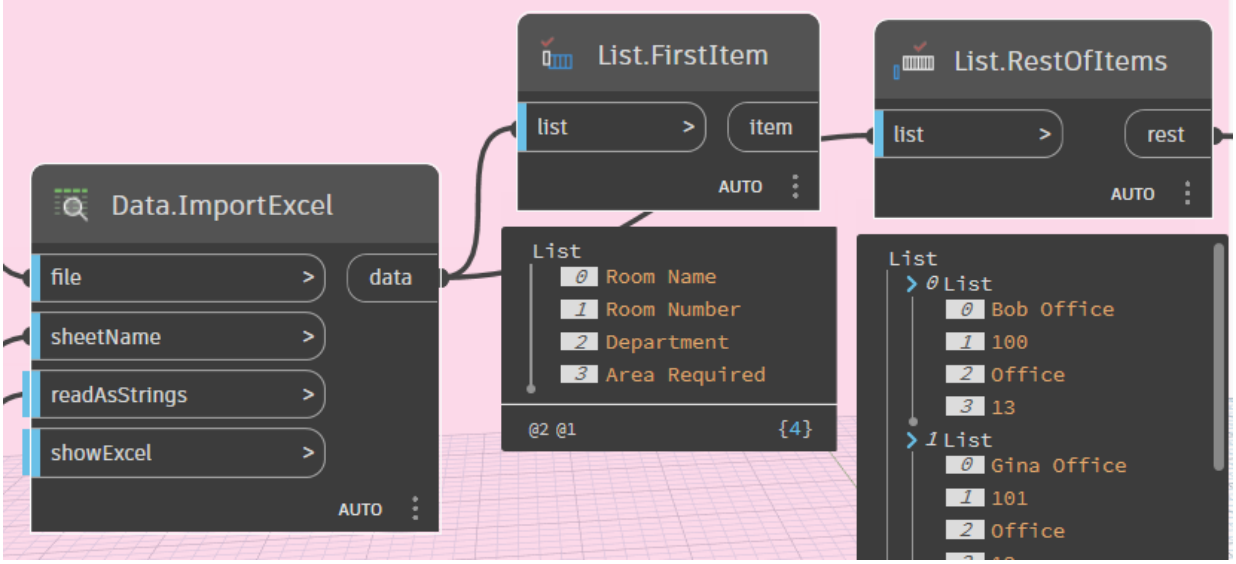
2- IMPORT EXCEL IN DYNAMO

Open Dynamo and let's start scripting. First, let's bring in Excel. Here are the nodes you should use. You need to specify a path. Then, use the **Data.ImportExcel** node. Specify the sheet, and read the value as strings.



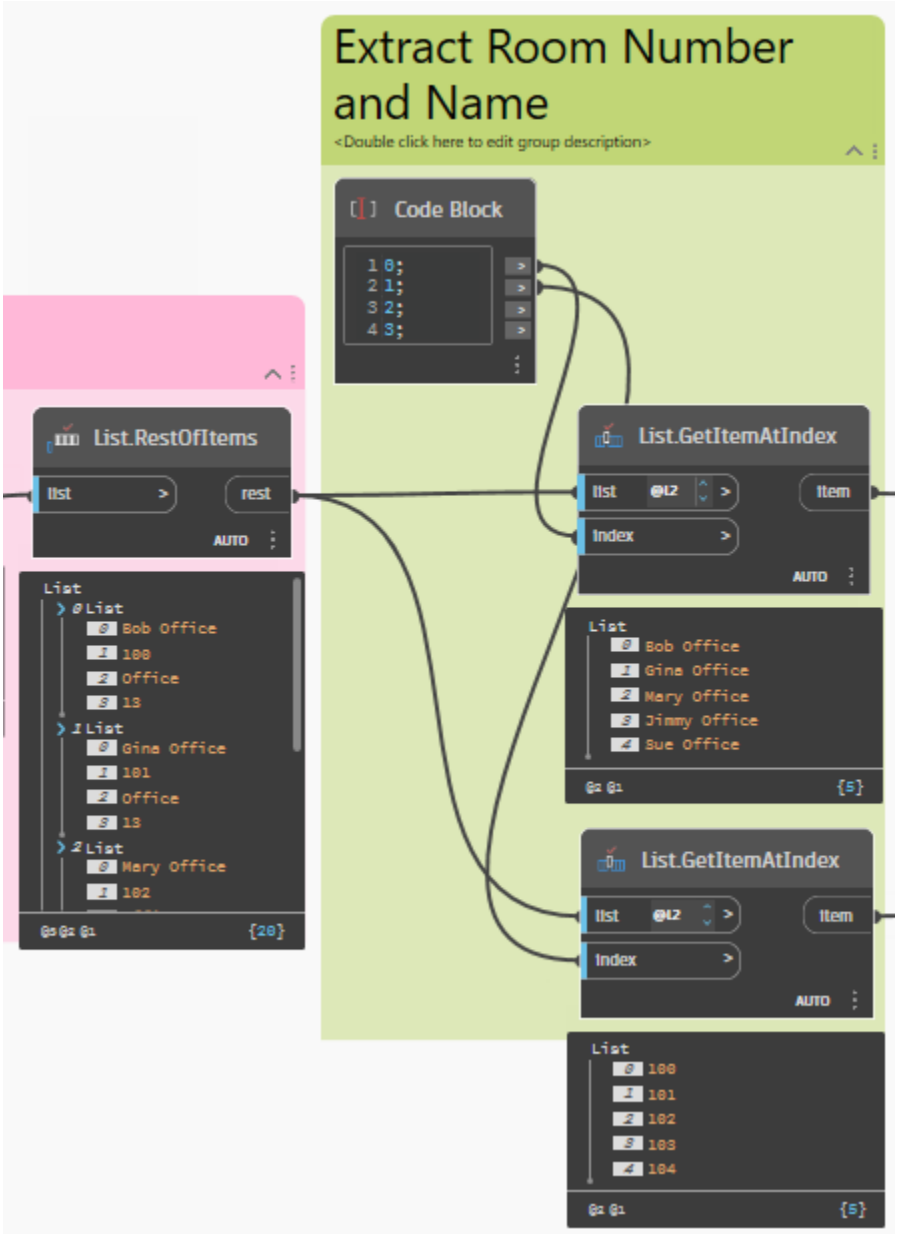
3- SEPARATE HEADERS FROM DATA

This script assume that there is a header on the first row, and the rest is data. We use **List.FirstItem** to get the headers and **List.RestOfItems** to get the rest. You can see that each sublist contains all information from the rooms.



4- GET LISTS OF SINGLE PARAMETERS

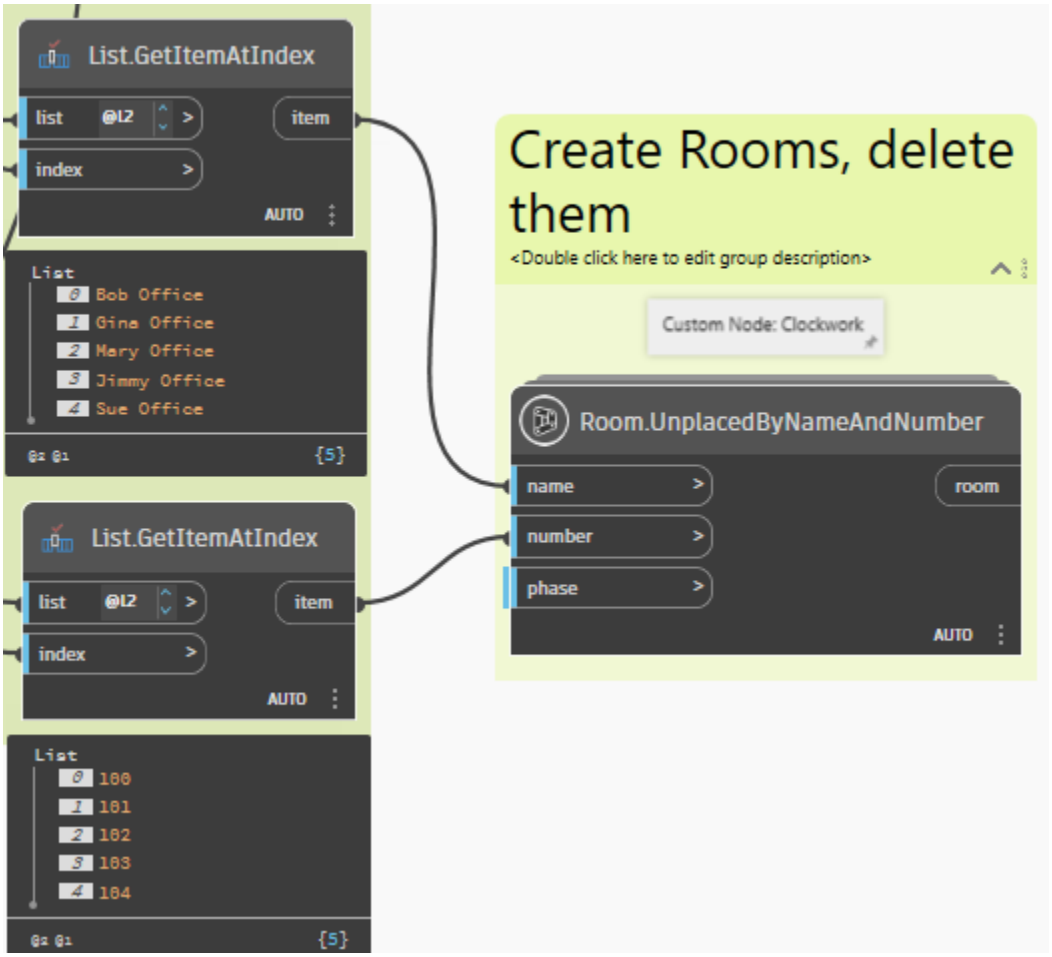
Let's separate the name, number, department and area required into different values. In this case, we know that room number is index 0, and room number is index 1 and so on. Use the node **GetItemAtIndex** to retrieve specific elements. We get a list of names and a list of numbers.



5- CREATE ROOMS AND DELETE THEM

Rooms in Revit are odd. They can be deleted, but remain in the database. This is what we want here, since all the information is imported from Excel, but the user will individually place the rooms themselves.

In this case, we use a node from the custom package **Clockwork** to create this task. It is called **Room.UnplacedByNameAndNumber**.



6- VERIFY THE REVIT ROOM SCHEDULE

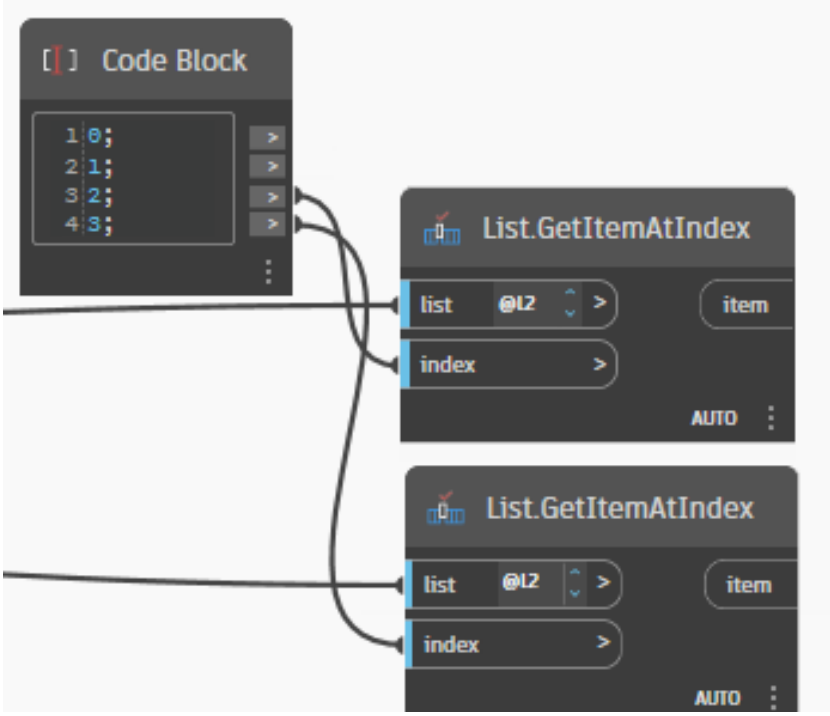
When you check out the Revit schedule, you should see this. This is good, but we aren't done. We need the department name and the area required fields.

<Room Schedule>

A	B
Name	Number
Room Name	100
Bob Office	100
Gina Office	101
Mary Office	102
Jimmy Office	103
Sue Office	104

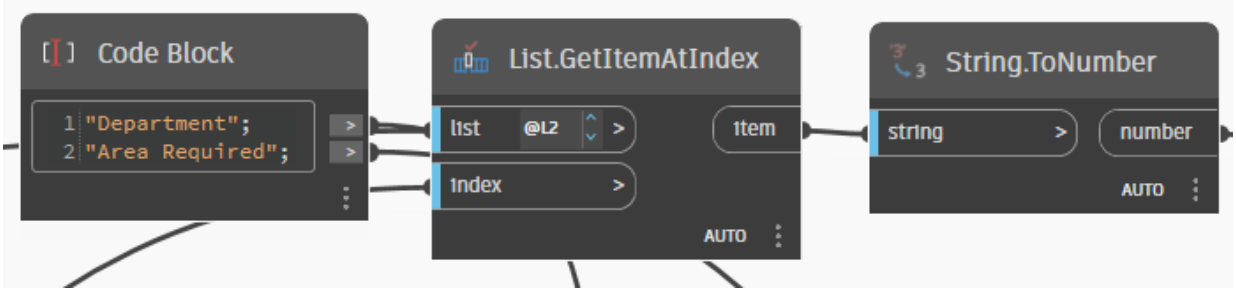
7- GET THE DEPARTMENT AND REQUIRED AREA

Once again, use the **GetItemAtIndex** node to get the department and required area values.



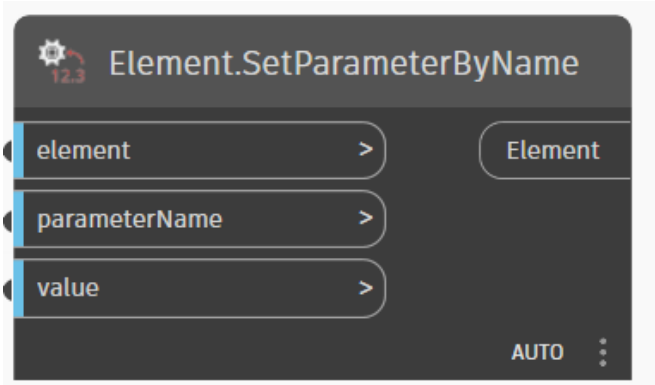
8- CONVERT AREA FROM STRING TO NUMBER

The Required Area from Excel is currently a string. We need a number, since Revit expects an area value. Use the **String.ToNumber** node to accomplish this.

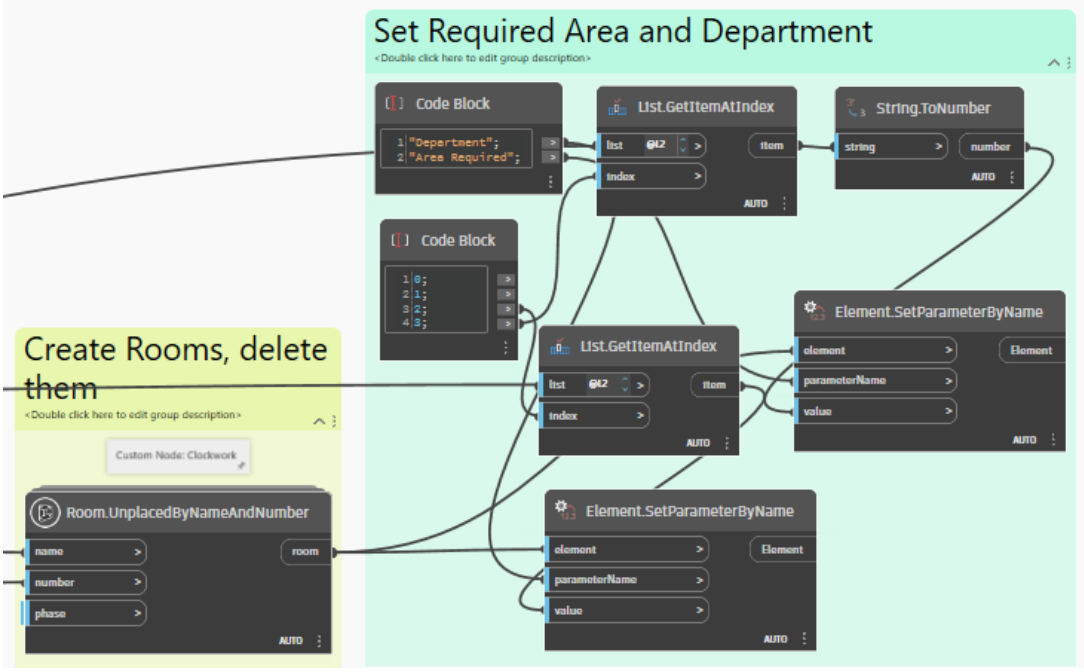


9- SET AREA REQUIRED AND DEPARTMENT VALUES

The data is now ready, in separate lists. Use the **SetParameterByName** node to set both these values. Make sure to use the proper parameter name.



These values can be added after the rooms have been created and deleted. Here are the final nodes:



10- VERIFY THE REVIT SCHEDULE

Head back to Revit. Your Revit schedule should look like this.

<Room Schedule>

A	B	C	D
Name	Number	Department	Area Required
Bob Office	100	Office	13 m²
Gina Office	101	Office	13 m²
Mary Office	102	Office	13 m²
Jimmy Office	103	Office	13 m²
Sue Office	104	CEO	18 m²

When you place a Revit room, make sure to pick from the available list of rooms:

PUSH EXCEL DATA TO REVIT

Let's create another script.

After the rooms have been placed, the project manager still updates the room spreadsheet with important information. In addition, we have more requirement:

1- The script should work with any projects.
2- The data might not always be in the same order.

That means the Name column might sometimes be in column 1, sometimes in column 2 and so on. We'll need to search for the field.

Also, we'll need to find a way to have the Excel spreadsheet location in as a parameter in the Revit model, so users don't have to manually pick it each time. That makes the script **universal**.

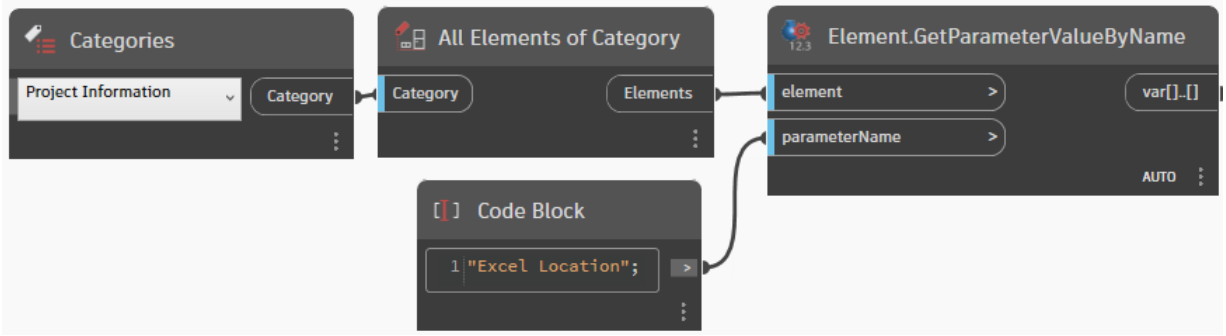
1- ADD "EXCEL LOCATION" PARAMETER

There is a different associated Excel sheet for each Revit model. Create a new parameter called Excel Location. It is a **text** parameter and should be set to **Project Information**. Set the windows location of the spreadsheet to this parameter:

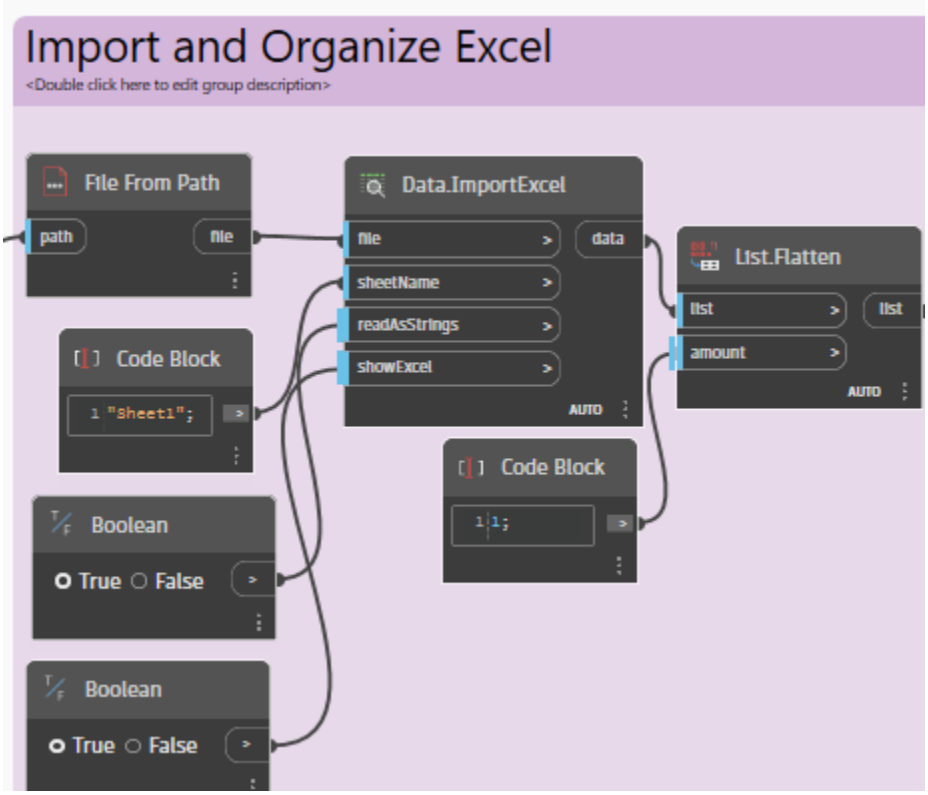
Parameter	Value
Text	
Excel Location	C:\Users\nicoc\OneDrive\Desktop\Pamphlet30-Excel-Rooms.xlsx

2- IN DYNAMO, EXTRACT EXCEL LOCATION

Use these nodes to get the Excel Location text parameter:

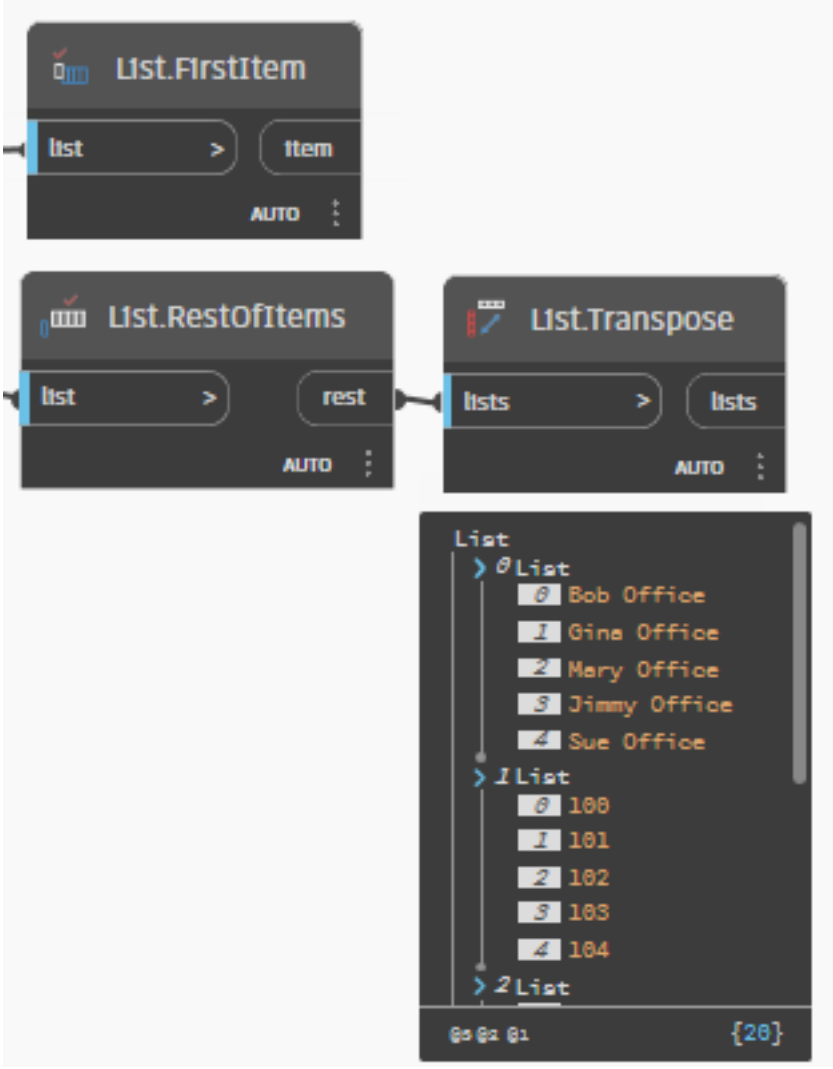


The next steps are quite similar to the previous script. Retrieve Excel data. In this case, you must flatten the list by an amount of 1. This is because we are using a text parameter for the file path.



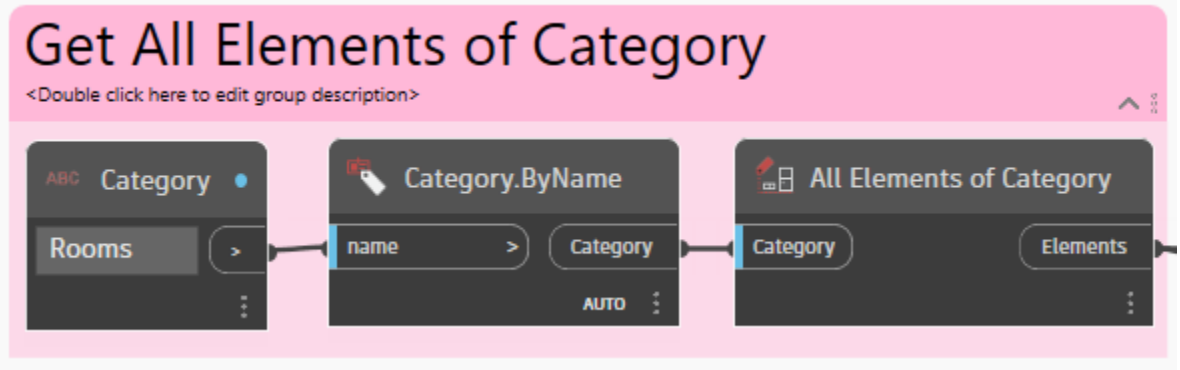
3- TRANSPOSE REST OF DATA

Separate the data into the **FirstItem**, then into the **RestOfItems**. For the **RestOfItems**, use the **List.Transpose** node. This will regroup the values for each parameter.

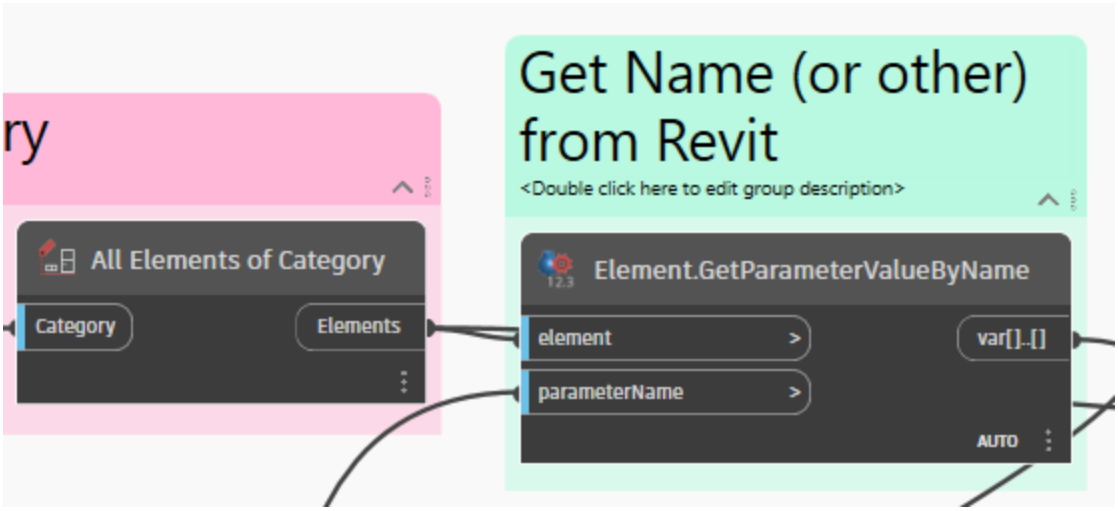


4- GET ALL ROOMS, EXTRACT THE NAME

Now, let's get all the rooms in the project.

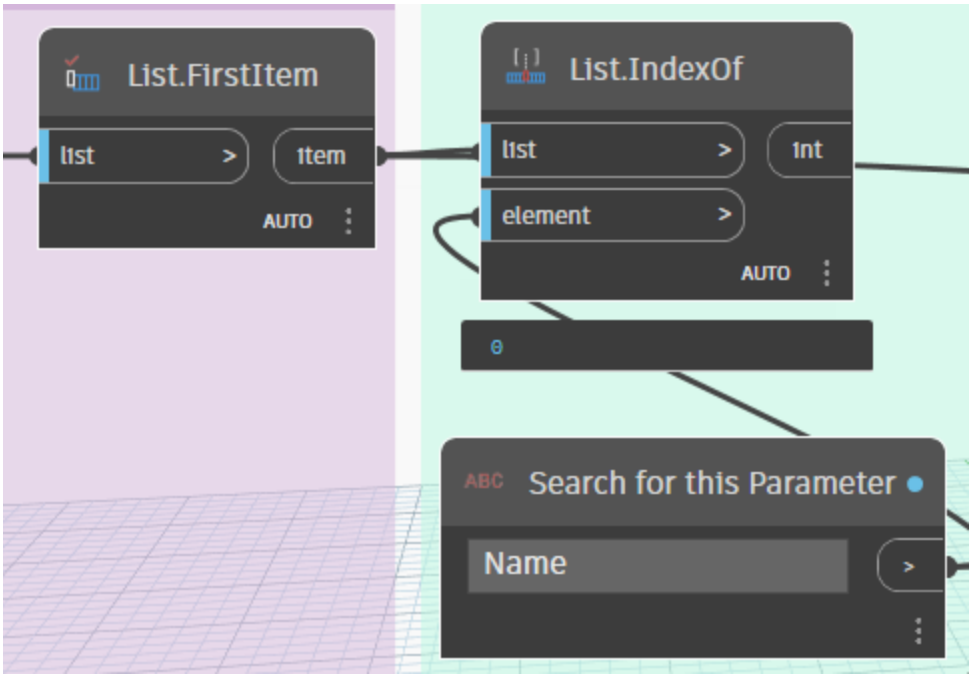


Extract the name from the rooms.



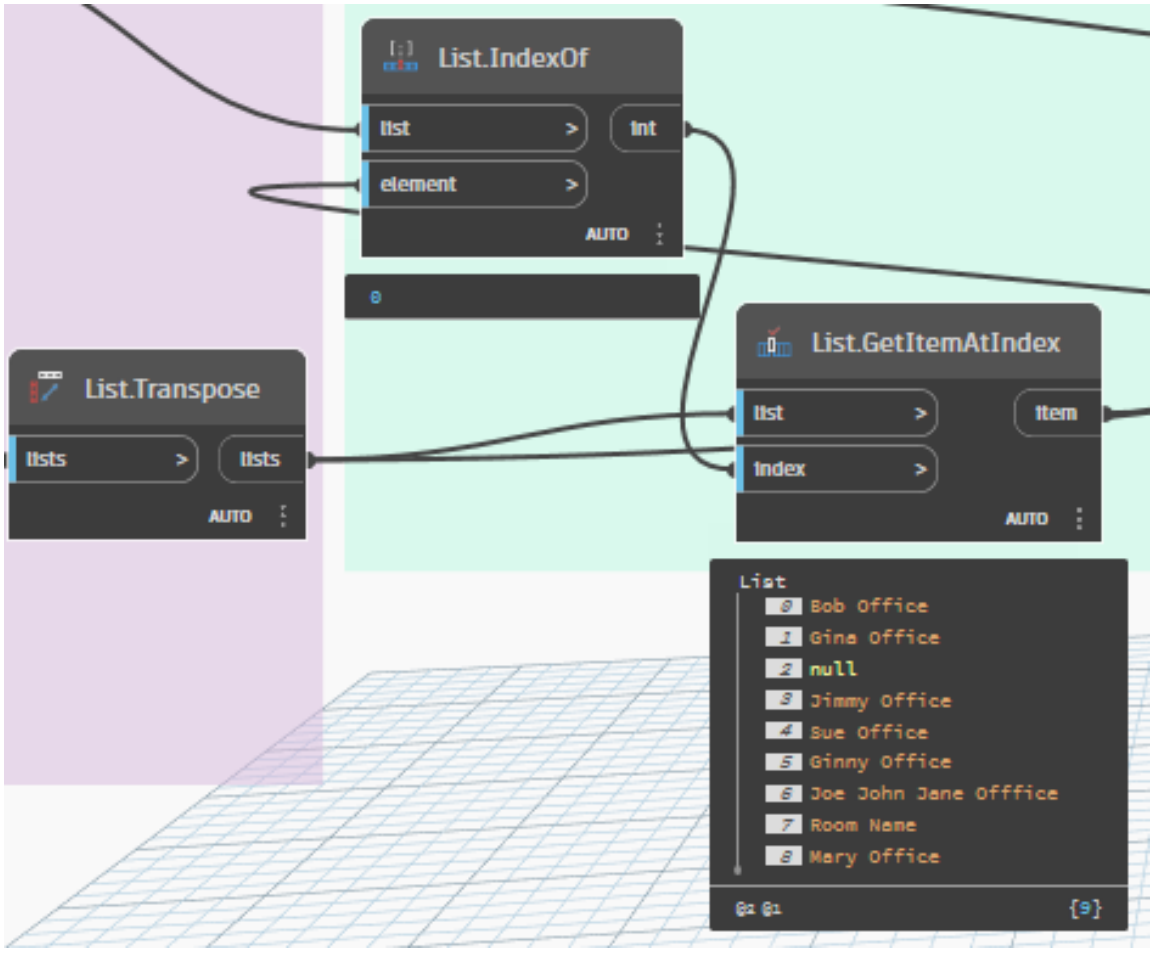
5- GET COLUMN NUMBER FOR NAME PARAMETER

The **List.IndexOf** node lets us know in what column the Name parameter is located. This way, the script works regardless of which column the name parameter is located. Since the name is on the first Excel column, we get the value 0.



6- EXTRACT EXCEL ROOM NAMES

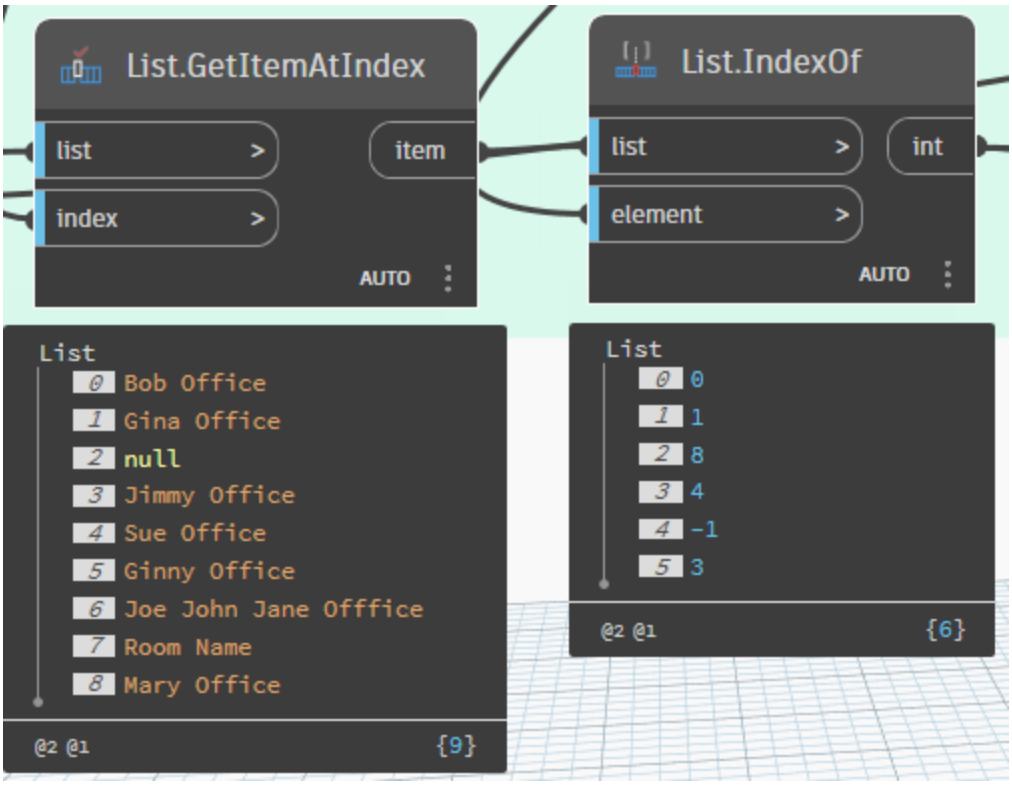
Use the **List.GetItemAtIndex** node to get all the values in column 0.



Now, we have a list of Excel room names and a list of Revit room names. You only want to keep elements that appear in both list. Read on.

7- GET INDEX OF REVIT ROOM NAMES IN EXCEL

Once again, use the **List.IndexOf** node. This time, the elements are the revit room names, and the list is the Excel room names.

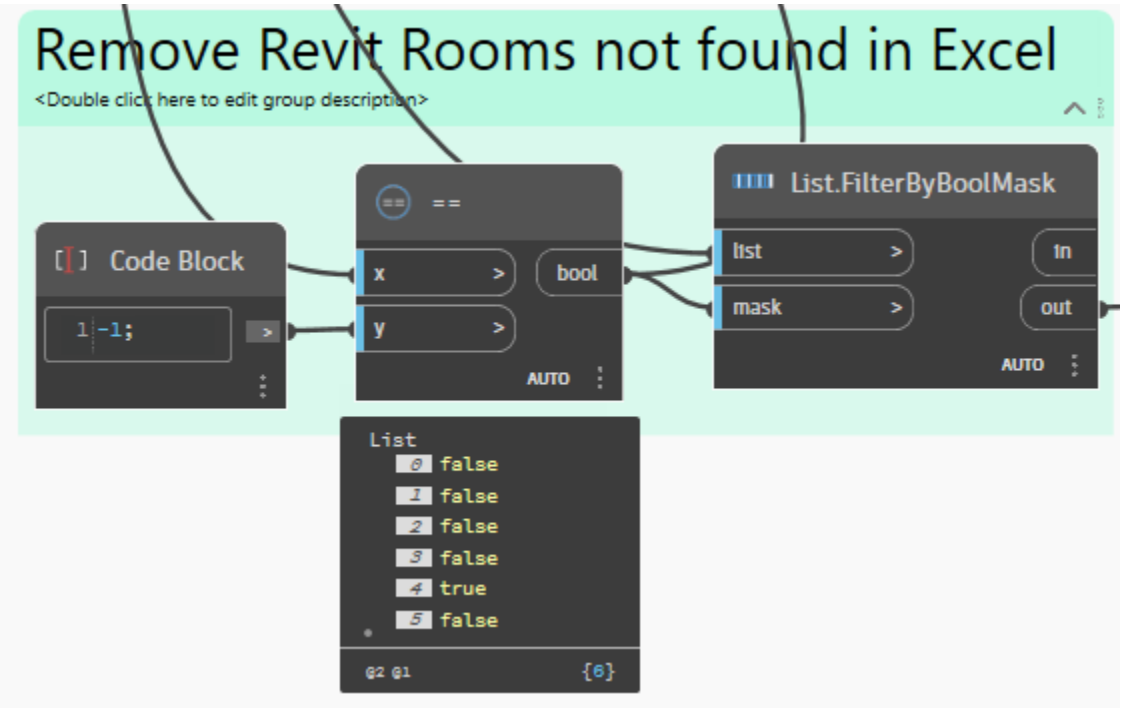


You can see we get a list of values. Revit room #1 is located at index 0, which is the first row. Revit room #3 is located at index 8, which is the 9th row. Remember that in Dynamo, the first item is set at 0, and the second at 1.

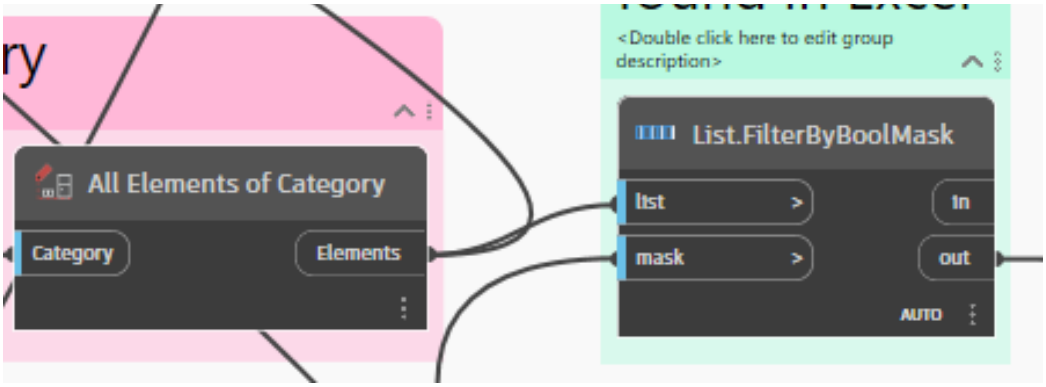
A value of -1 means that the Revit room wasn't found in Excel. You'll want to get rid of it.

8- FILTER OUT REVIT ROOMS NOT FOUND IN EXCEL

Use the **==** and the **List.FilterByBoolMask** nodes to remove all -1 values. Since all -1 values will return a true value, keep the elements in the **out** output from the Filter node.



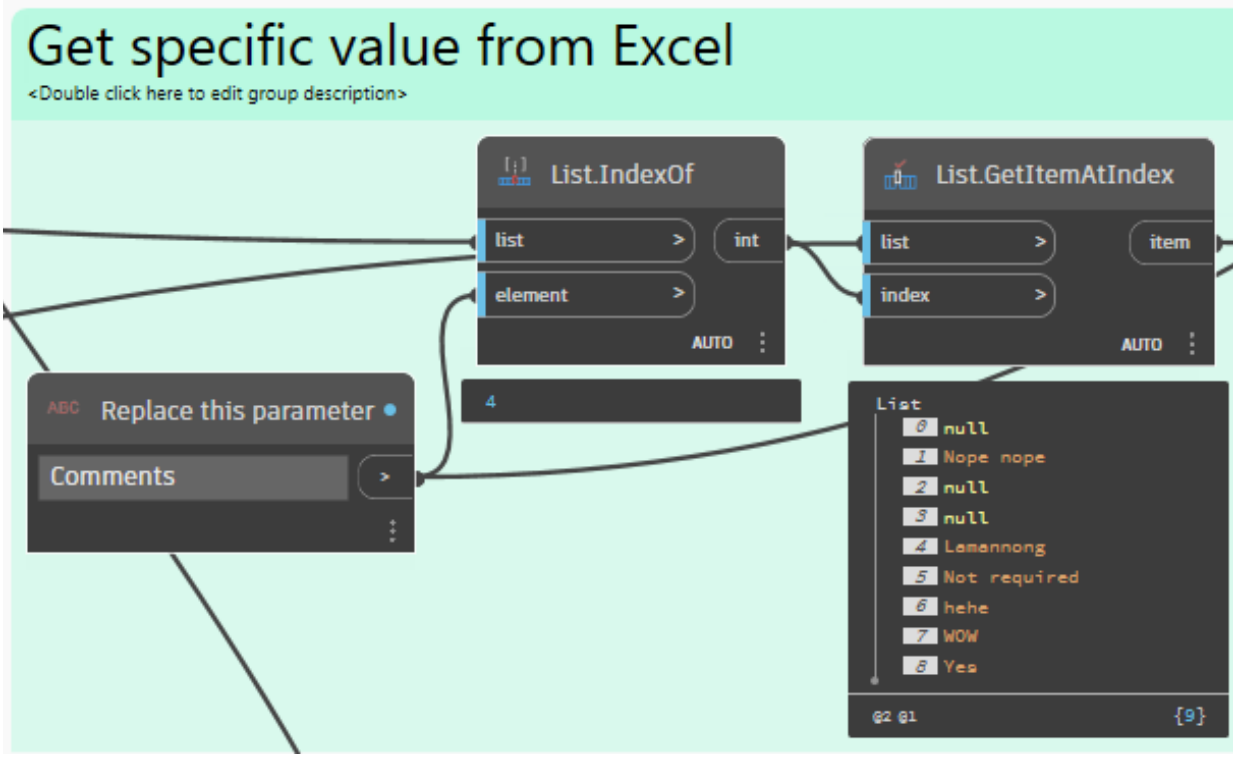
You'll also want to filter out the actual Revit rooms with the same output from the == node.



9- GET PARAMETER VALUES TO MODIFY IN REVIT

In this example, let's say you want to modify the **Comments** value. All Revit rooms that have a name match in Excel will have their comments value modified.

The first part is similar to extracting the name value form Excel. Except that this time, we search for the Comments value. This is set at index 4, so the 5th column in Excel. Then, we acquire all the comment values from Excel.

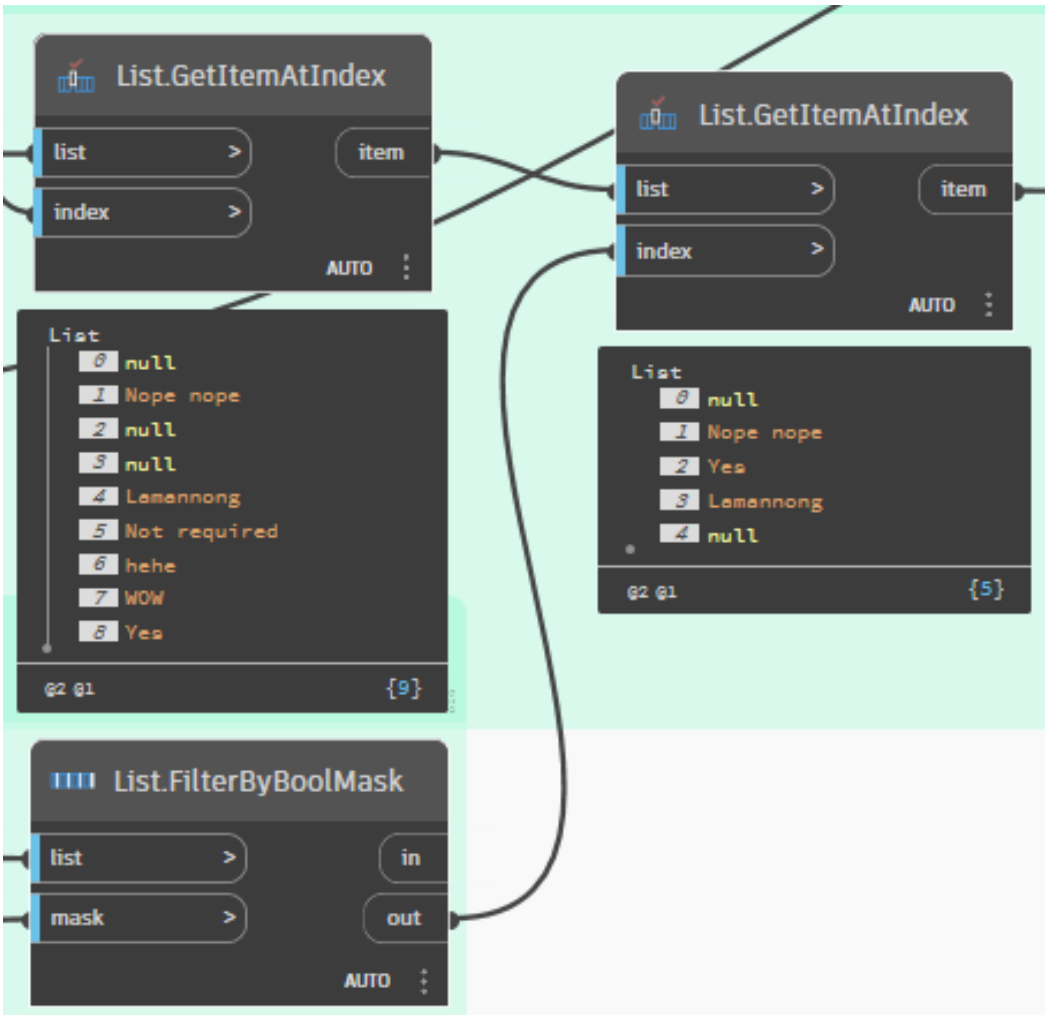


10- ONLY KEEP FILTERED, ORDERED VALUES

Use the output from the **FilterByBoolMask** node you've created, and stick it in the **GetItemAtIndex** node.

This way, you only keep the comments values from room names that are found in both Revit and Excel.

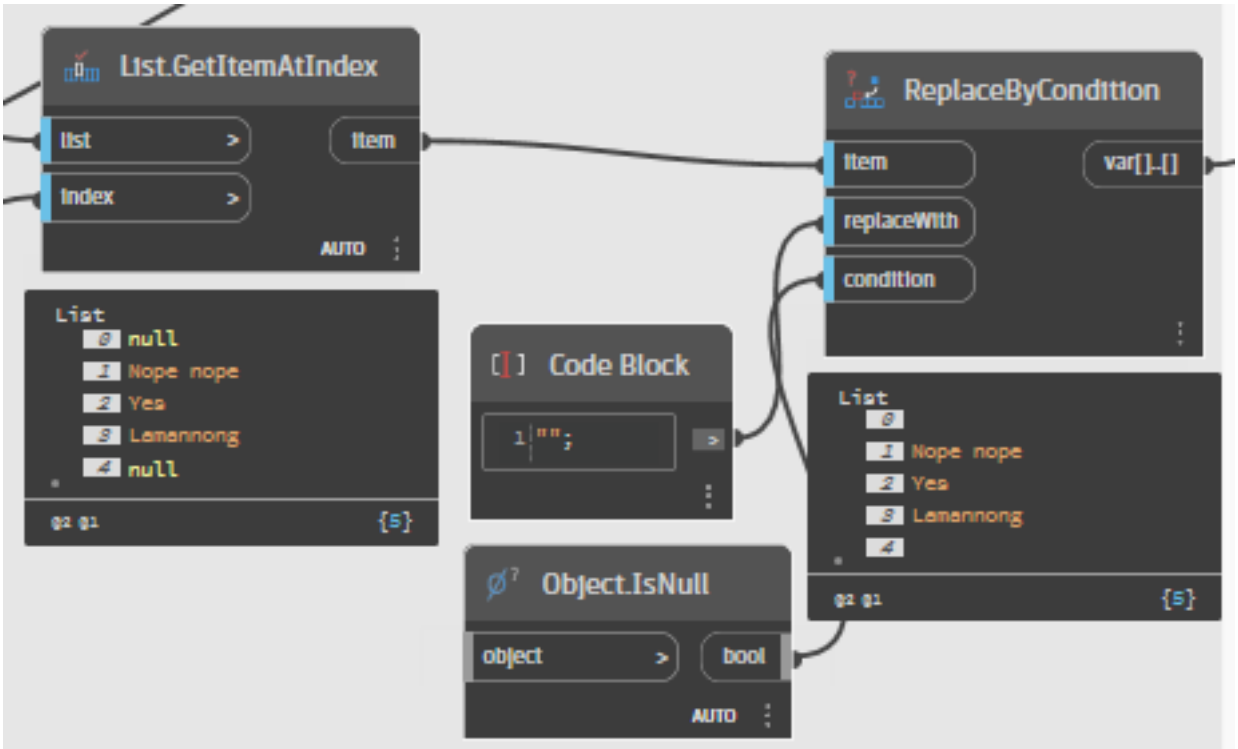
Also, the comments value are reordered to be in the same order as the Revit rooms in the list.



11- REPLACE NULL VALUES BY EMPTY STRING

If a value is empty in Excel, this will cause a “Null” element in Dynamo. That can mess things up and create warnings. Use the **ReplaceByCondition** and **Object.IsNull** nodes to replace the Null elements by an empty string.

As you can see, the yellow “null” are replaced by emptiness.



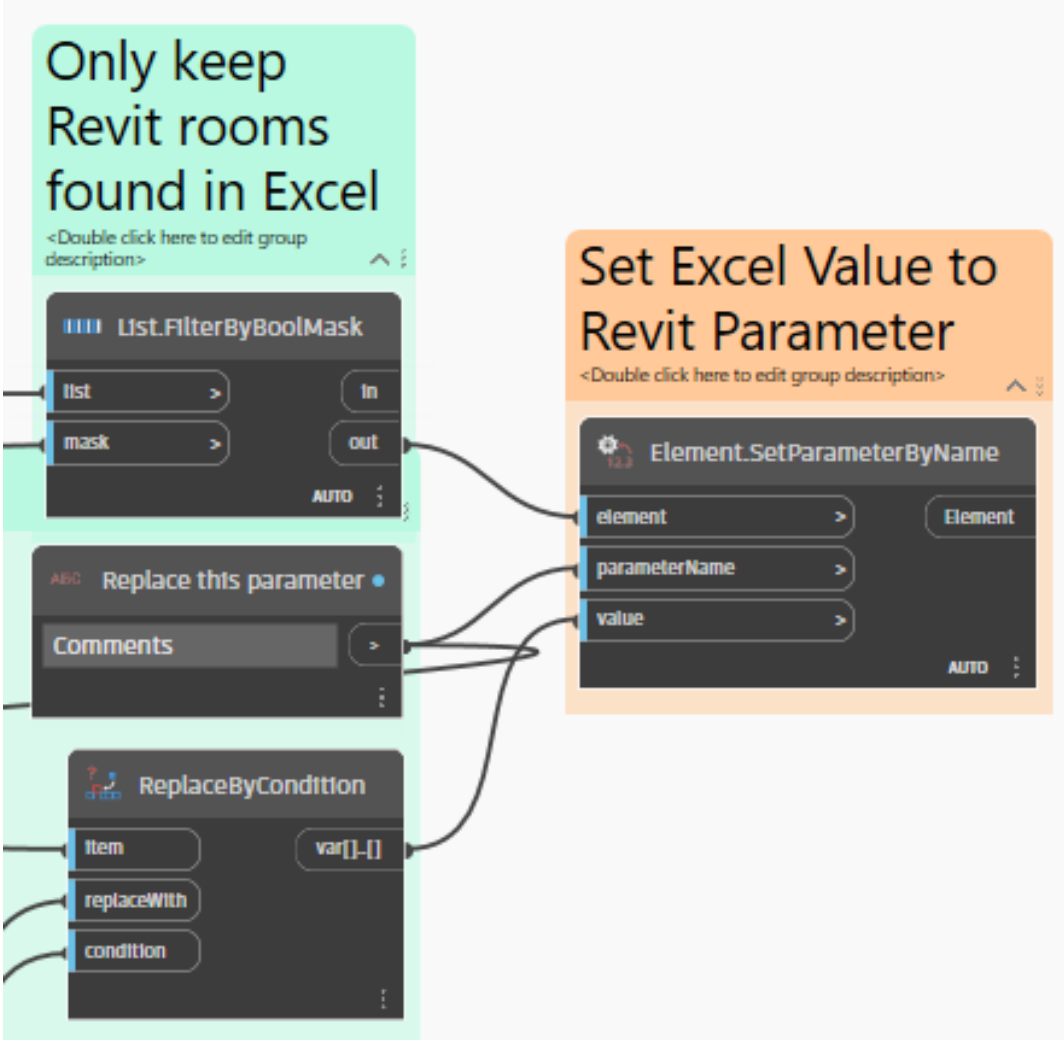
12- SET PARAMETER VALUE BY NAME TO ROOMS

The final step. Use the **Element.SetParameterByName** node.

The elements are the Revit rooms, excluding those not found in Excel.

The parameter is chosen by the user. In this case, Comments.

The values are the strings extracted from Excel.



13- VERIFY REVIT SCHEDULE DATA

Verify the Revit schedule. Ensure the values match Excel.

<Room Schedule>

A	B	C	D	E
Name	Number	Department	Area Required	Comments
Bob Office	100	Office	13 m²	
Gina Office	101	Office	13 m²	Nope nope
Mary Office	102	Office	13 m²	Yes
Sue Office	104	CEO	18 m²	Lamannong
Geimini John	106			
Jimmy Office	109			

What is unique about this script is that:

- You don't need to pick the Excel file each time. This is extracted from a field in each Revit model.

- The parameters in Excel don't have to be in a specific order. The names only need to match.

- The script only keeps elements that are both in Excel and Revit. Not a big deal if the list is not identical.

UNLEASHING DYNAMO MULTIPLAYER

Dynamo Multiplayer is an amazing free plugin developed by Bird Tools. You can click here to download it.

This plugin allows you too:

- Batch run one or multiple scripts on several Revit models, including ACC/BIM 360 models.

- Schedule these scripts to run while you are away.

Let's use the script we've just created and see how it can be batched and scheduled on multiple models.

UPDATING ROOM INFO ON EXCEL

Let's clearly define the hypothetical scenario:

- Each Revit model has one Excel file associated.

- Project manager frequently modify the Excel file with up to date information about rooms.

- BIM managers want to push the Excel information back to Revit, and ensure this task is automated.

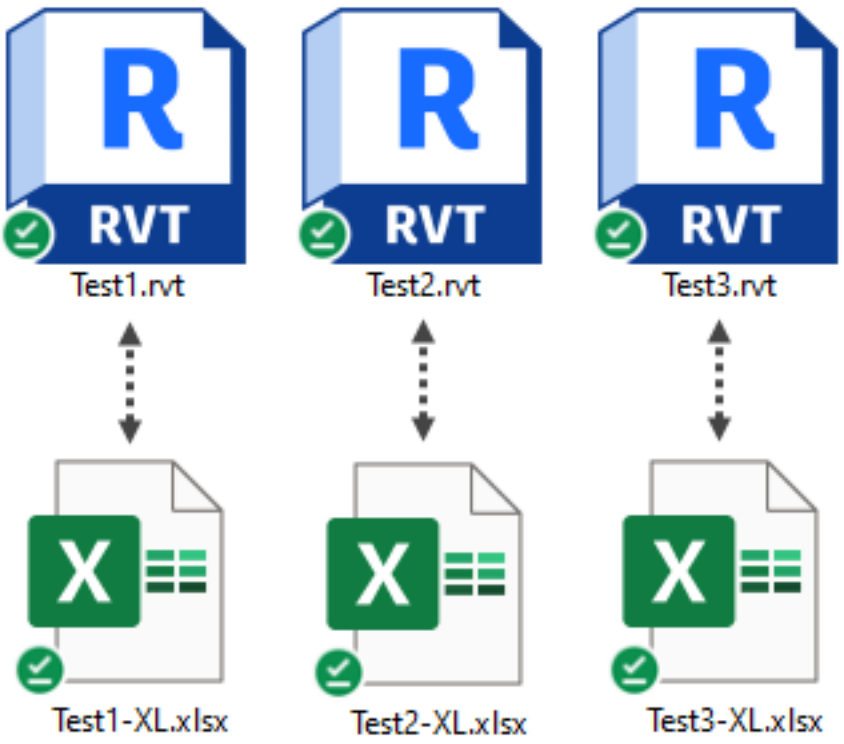
1- SET THE "EXCEL LOCATION" PARAMETER FOR EACH REVIT MODEL

Each Revit file has an Excel Location parameter in the project information. Make sure this is correctly defined for all models.

Parameter	Value
Text	
Excel Location	C:\Users\nicoc\OneDrive\Desktop\Pamphlet30-Excel-Rooms.xlsx

In this case, we have 3 Revit models. Each of these models is associated to a different Excel file.

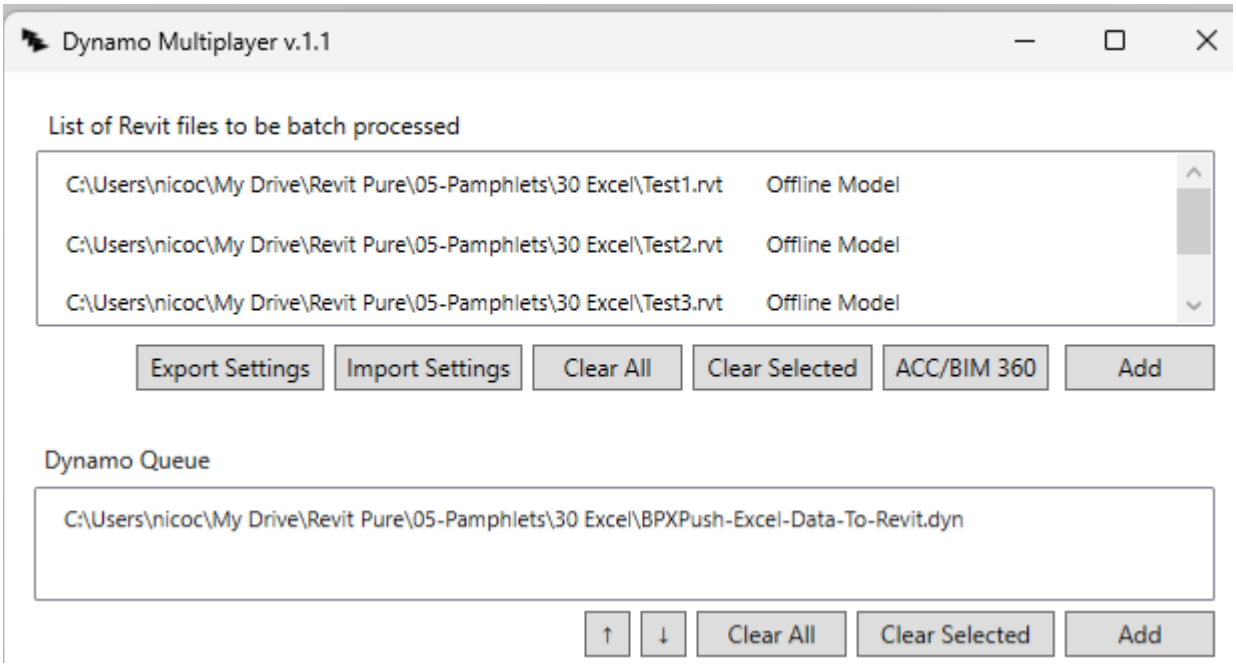
In the following steps, Excel data will be pushed to the Revit models in a batch, scheduled task.



2- OPEN DYNAMO MULTIPLAYER, IMPORT REVIT PROJECTS, SET DYNAMO SCRIPT

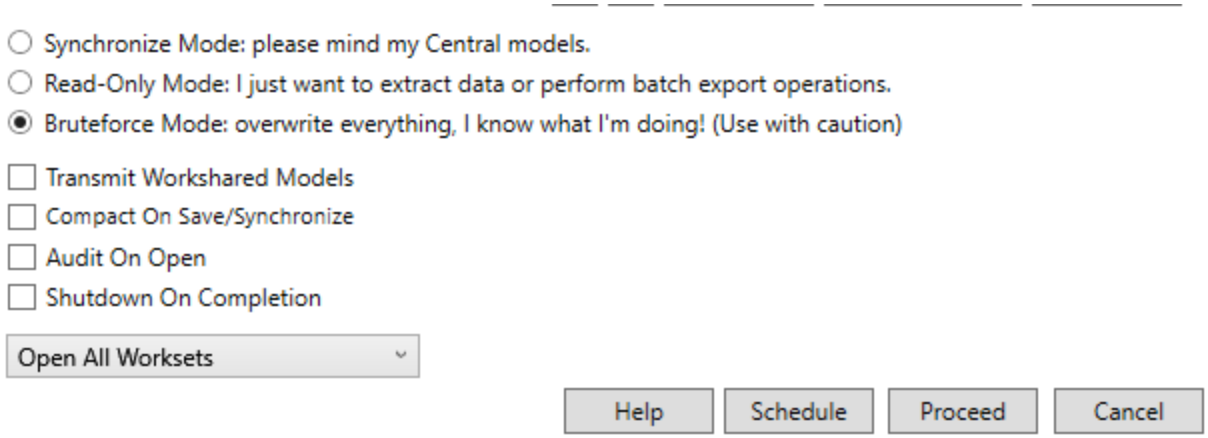
Dynamo Multiplayer is quite simple. On the top, add Revit models. At the bottom, add one or multiple Dynamo scripts. The files you add don't need to be currently opened.

in this case, a single Dynamo script is fine. We are using local Revit files, but this can work with cloud models too.



3- RUN THE SCRIPT BRUTEFORCE MODE

At the bottom of the Dynamo multiplayer menu, you have to pick between **Synchronize Mode**, **Read-Only Mode** and **Bruteforce Mode**.



Since these are all local models, you should pick the **Bruteforce Mode**. This will overwrite everything. Of course, you need to be cautious. In this case, we know that the script only affects the room comments, so not too much to worry about.

For more complex scripts, you should test and double-check everything to ensure you don't lose important data.

When you are ready, click on **Proceed**. It might take a few moments, but the script should run through the 3 models.

If you are working with cloud workshared models, use the Synchronize Mode.

4- VERIFY IF THE RESULTS ARE AS EXPECTED

Open each of the 3 Revit files. The information on Excel should be identical to the information on the Revit room schedules.

As you can see in Test3.rvt, the script has indeed push the information from Excel to Revit.

<Room Schedule>

A	B	C	D	E
Name	Number	Department	Area Required	Comments
Bob Office	100	Office	13 m²	wow
Gina Office	101	Office	13 m²	Change flooring
Mary Office	102	Office	13 m²	Yes
Sue Office	104	CEO	18 m²	Sue is off
Geimini John	106			
Jimmy Office	109			

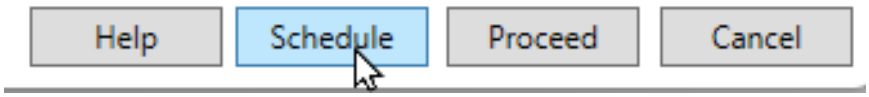
	A	B	C	D	E
1	Name	Number	Department	Area Required	Comments
2	Bob Office	100	Office	13	wow
3	Gina Office	101	Office	13	Change flooring
4					
5	Jimmy Office	103	Office	13	
6	Sue Office	104	CEO	18	Sue is off
7	Ginny Office	105	Office	14	Remove this room
8	Joe John Jane Offfice	110	fasdf	18	hehe
9	Room Name	106	keke	22	
10	Mary Office	102	Office	13	Yes

5- SCHEDULE THE BATCH RUN

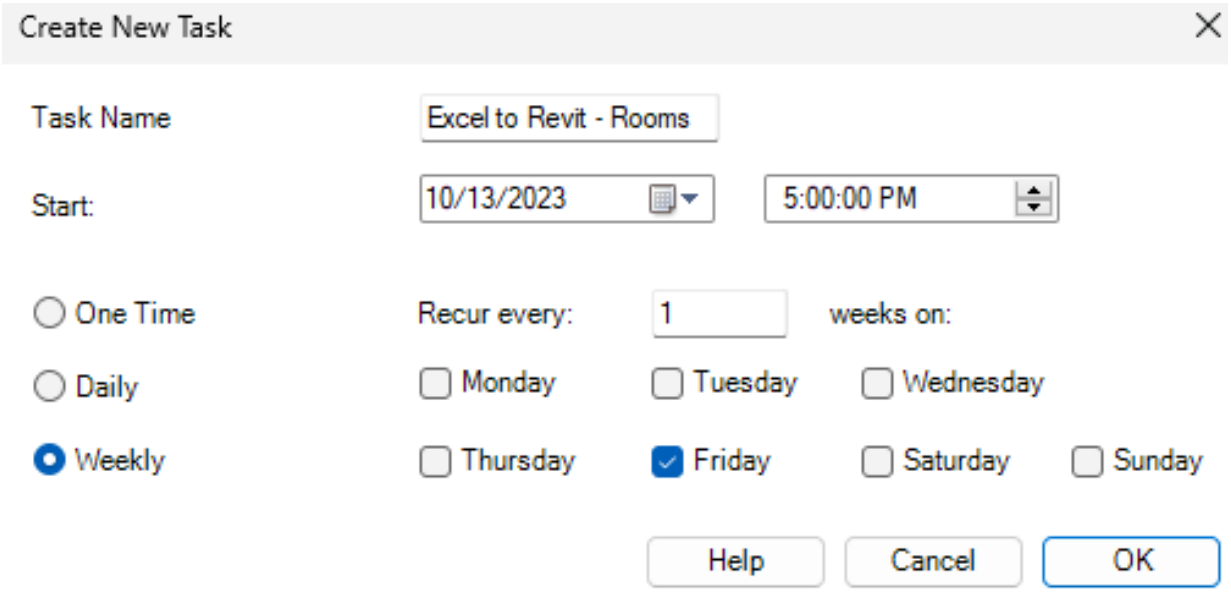
Let's say that you want to run this routine once a week. For example, each friday afternoon.

Go back to Dynamo Multiplayer and select the Revit models and the Dynamo script.

Click on **Schedule**.

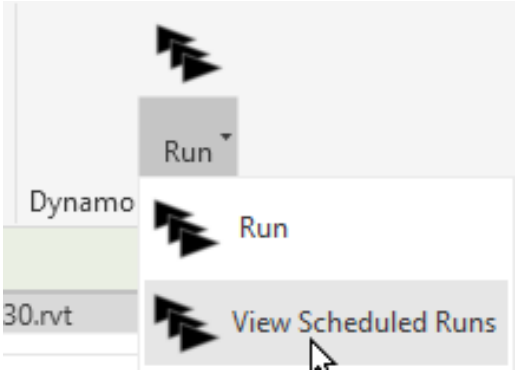


Select the moment to run the task. In this case, we pick a weekly schedule.

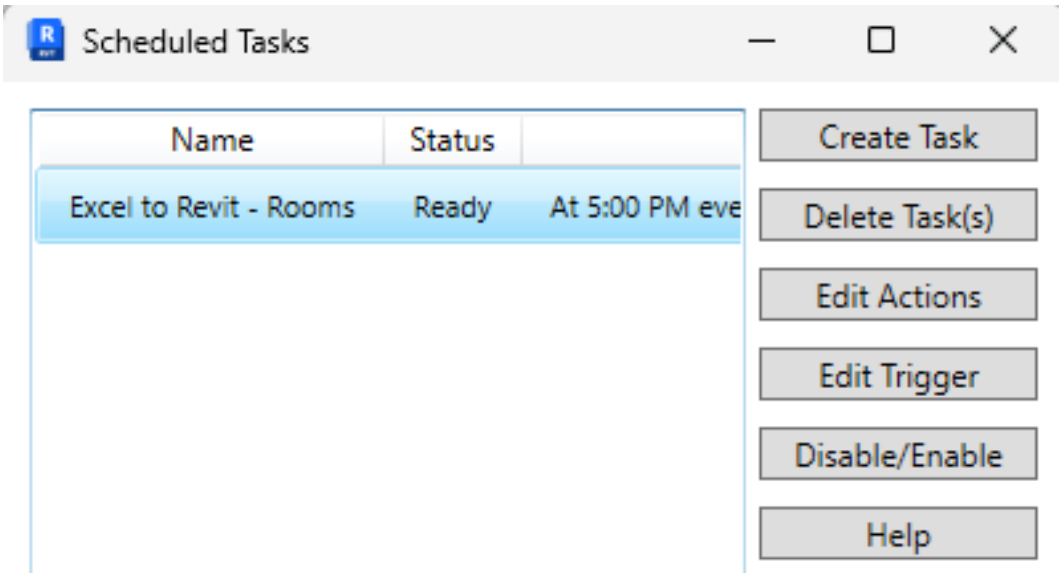


6- VERIFY AND MODIFY THE TASK SCHEDULE

You can always access the scheduled run. Click on the dropdown menu of the Dynamo Multiplayer tool.



You can see all tasks and modify them.



7- LET THE TASK RUN ITSELF

You only need your computer to be opened for the task to run. Revit will automatically open and run the task.

Still, you should verify if everything ran as expected. Once again, the Excel data is pushed to Revit! Yay.

Thanks to Bird Tools for creating this amazing free plugin.

Click here to download Dynamo Multiplayer

THANKS FOR READING!

As always, send your thoughts to nick@revitpure.com. I read and answer all emails. Let me know what theme you want explored for the next edition.