# Dynamo for Revit 2017 - Automatic Labelling of Doors with Level Reference

Using Dynamo to count doors and automatically label according to Revit Floor Level.

Revit is a big step forward over conventional CAD in terms of coordination and the amount of data that a project can contain. However, there is sometimes a need for a tool to work out an optimal design, explore design options in a more analytical way and test out a design or to simply automate repetitive tasks.

A tool for this is supplied with Revit, known as Dynamo for Revit. Dynamo is a free and open source scripting tool. Its interface is visual in nature, working from a node based concept. Some developers have utilised the Dynamo scripts to create some useful tools to aid in fabrication of parts like Dynamo Unfold and the Structural Analysis Toolkit for Structural Engineers to query and refine a structural frame.

A simple script is shown in the Dynamo help. Although this is a very simple task we can do much more impressive things, like create complex geometry, build lists, analyse parts, locations and call up a range of information on Revit Families. e.g. Door Reference and Level location.
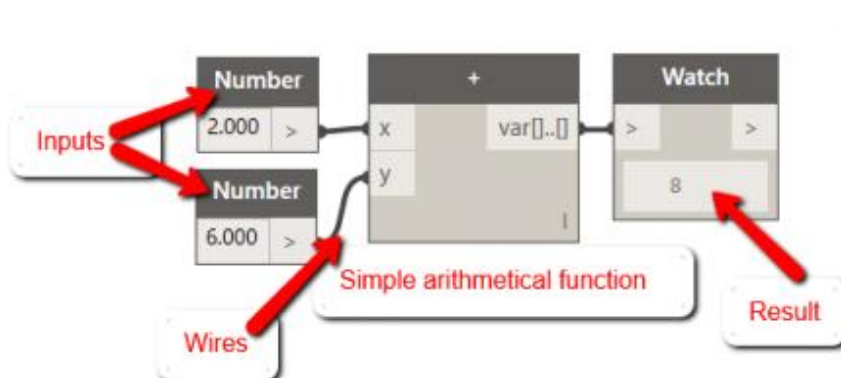

To access Dynamo, it is now located under the Manage tab in Revit, it also may be worth downloading and installing the latest version of Dynamo, which at the time of writing is 1.1.0, rather than the version supplied. (0.9x)
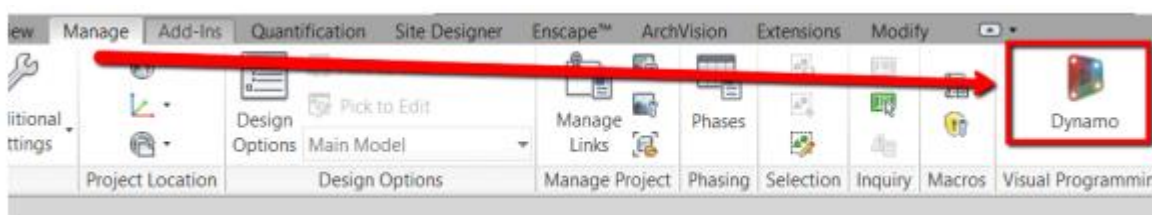
Clicking on Dynamo will fire up a second window where we can start to create a script.

One of the things we have to get our heads around is locating and understanding the application of Dynamo nodes.

These can be located from the Category and library list on the side panel. A predictive text style search box can also be used.
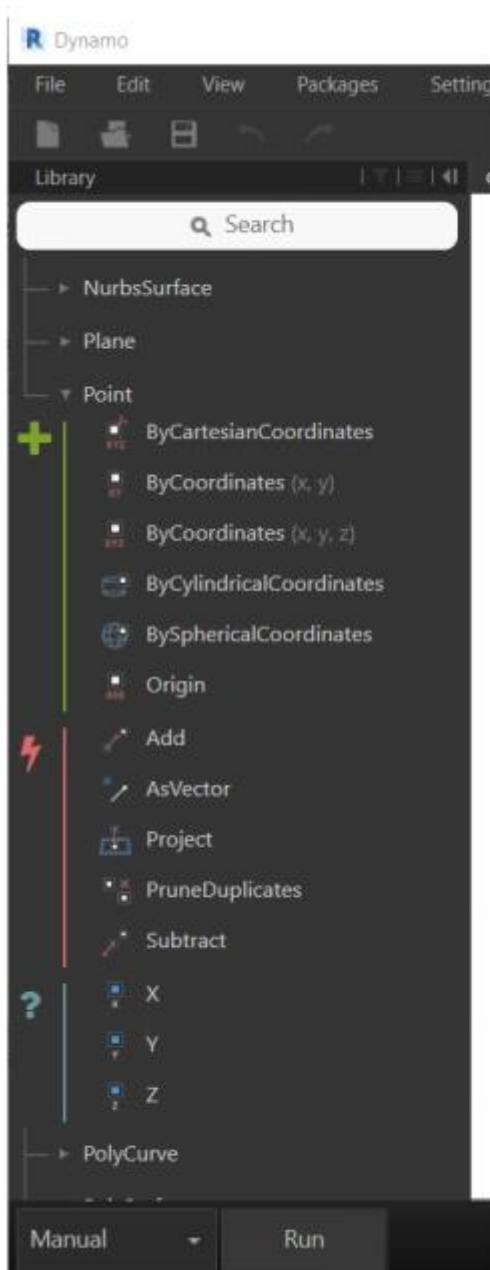
Dynamo Node Category List

The different colours are significant as they will do different tasks according to what is desired and needed in the script.

Create, adds objects by a range of different options

Modify, copy objects, perform boolean operations, insert points along other objects etc.

Query, obtains information, coordinates, length, direction, object data etc.

library is particularly useful as we can use the create nodes to list and sort various Revit families and extract data to generate a door reference which we'll feed back into the Revit Project. The Revit/Selection is what we see below:-
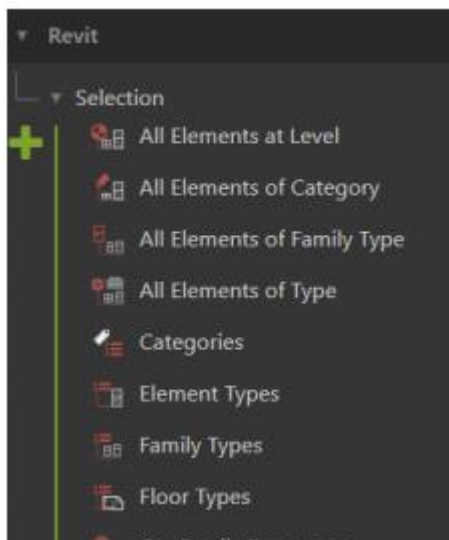
By choosing a node and clicking onscreen we can get something like below. Wiring the nodes and running the script will populate the results into a Watch node if desired.
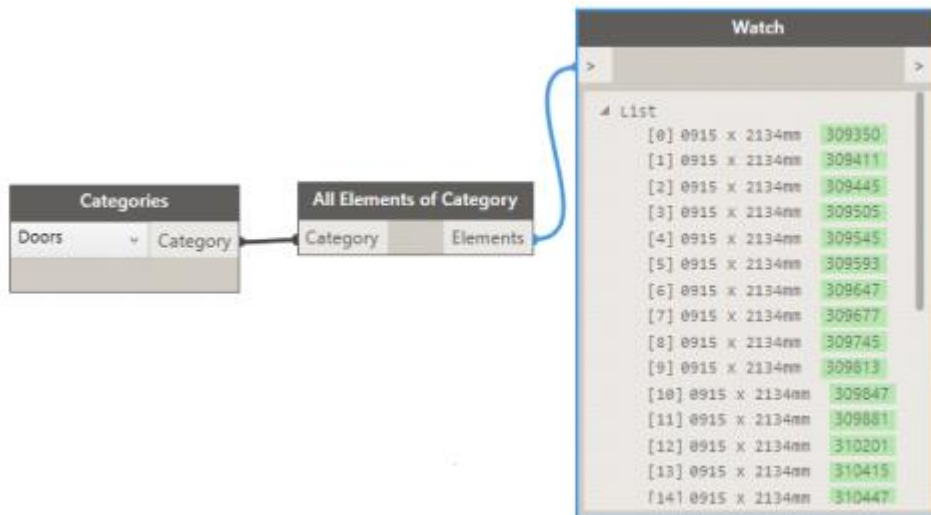
The Watch node is a valuable asset in diagnosing scripts and showing what information has been obtained and being driven so far.

Notice the counting of the elements, the family type name and unique ID. All of this can be utilised and sorted. In this case we want to break down all the doors by Level. So to do that we need another list!
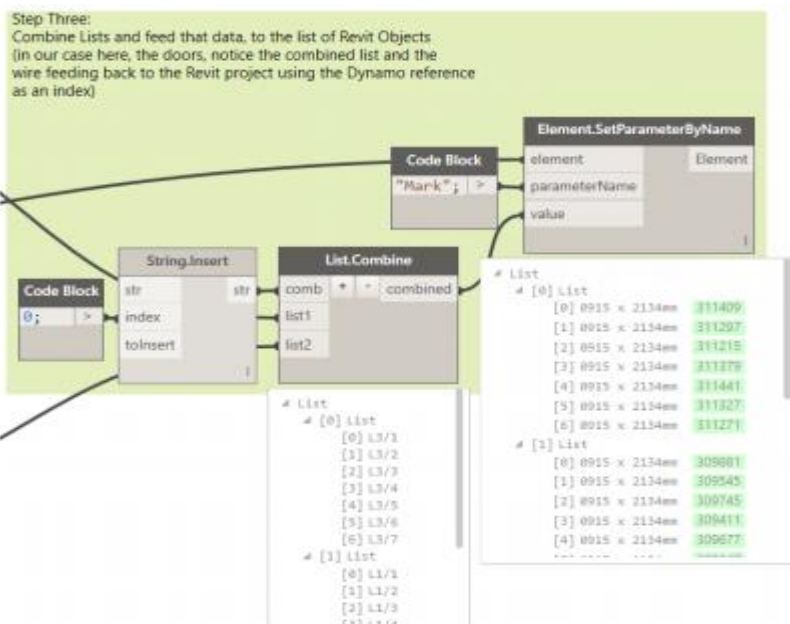
We use a Codeblock (another Dynamo essential) to obtain the level name within Revit to sort the doors into a more useful list.
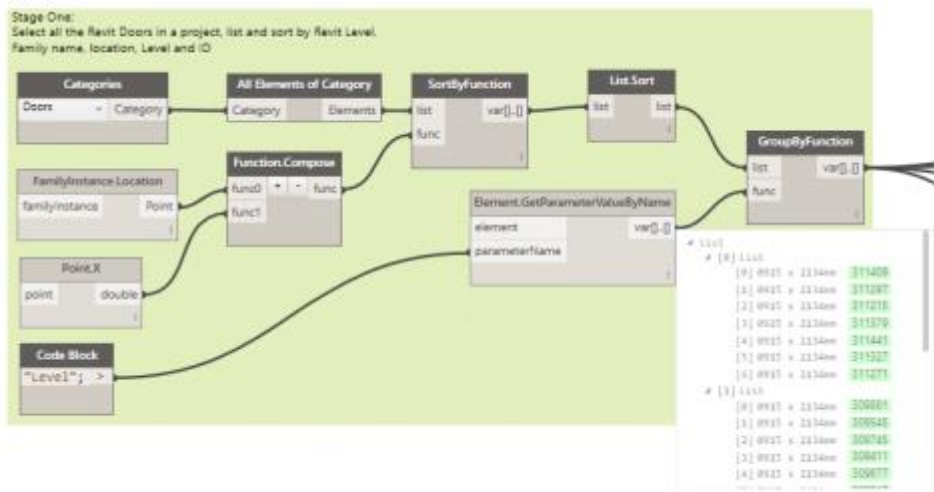
With the next step we can use Dynamo to obtain two lists of our door objects. The first is an index starting from a number and incrementing by one per level. The second list obtains the Level name and separator character which will be assigned per door.

Stage Two:
Build lists for Level Location , separator and a number for each door.
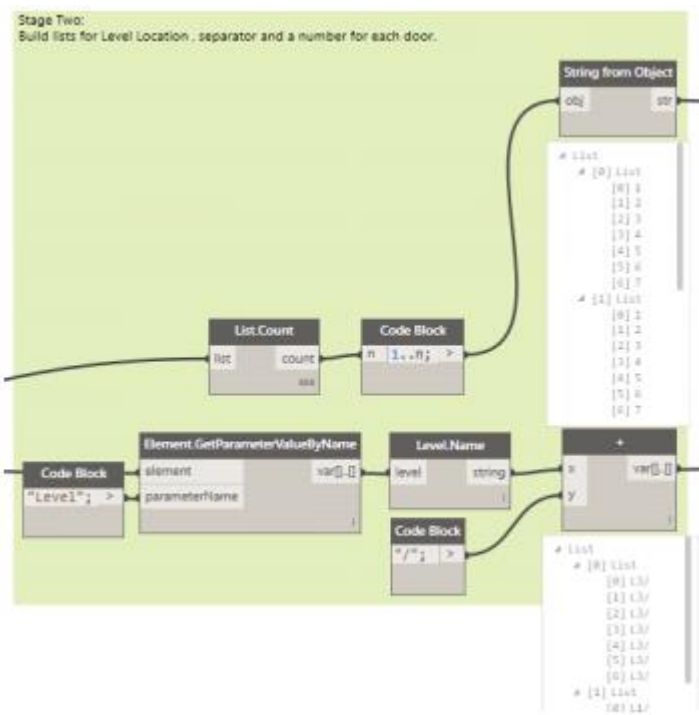
The third and next step in the script is to combine the two lists and feed those combined values back to our door objects in our Revit project.

Step Three:
Combine Lists and feed that data, to the list of Revit Objects
(in our case here, the doors, notice the combined list and the
wire feeding back to the Revit project using the Dynamo reference
as an index)

Key nodes here are List.Combine and Element.SetParameterByName, which will merge our two lists together and feed the project with a combined result. By going back to our Revit Project we can see below that Dynamo has been busy!
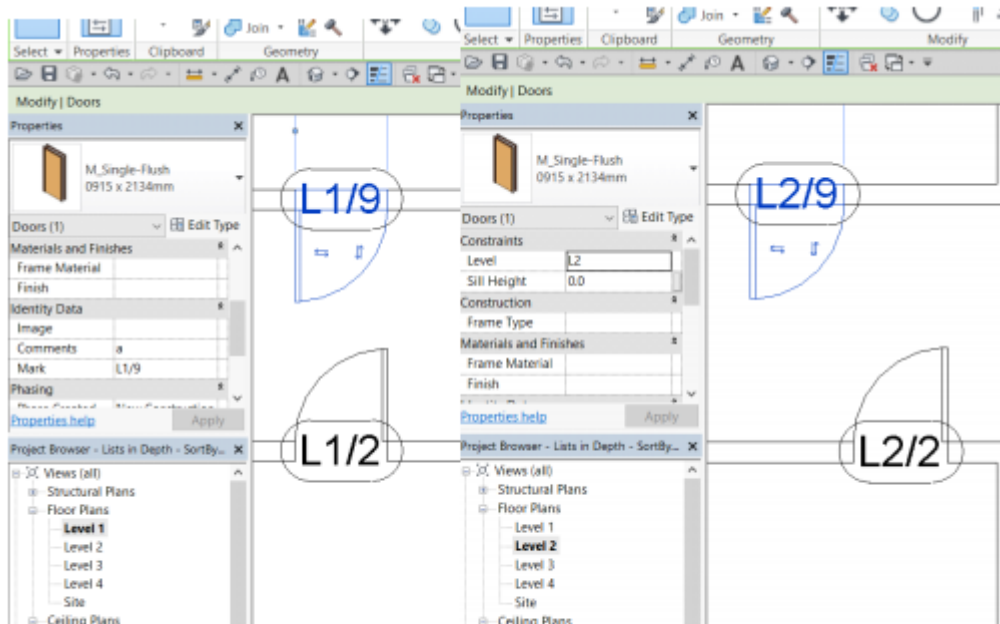
Mark parameter for all the doors in a Revit Project have been updated by Dynamo

Upon inspection of the floor plan views, we can see that all the doors have Mark values which contain the level name, a separator '/' and a unique number. This has been done for all floors; Level 1 has L1/*, Level 2 has L2/* and so on.

The number and indeed the parameter data can still be tweaked as desired. The tags were added manually by using the Revit Tag All command, but that too could be automated and added to the Dynamo script.

Using Dynamo to automatically place adaptive components on a mass.

Image Credit: Johan Cantryn

www.cadline.co.uk   01784 419 922   sales@cadline.co.uk   cadline
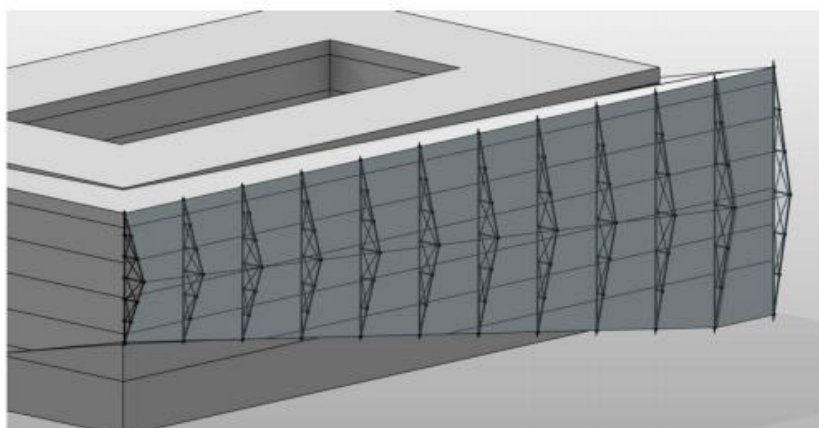
Other examples of using Dynamo could be the obtaining of coordinates for foundations schedules. As it ties in with the Revit API and such information can be easily obtained and be fed into schedules. Colouring evacuation lines linked to time for fire drawings and placing multiple views on multiple sheets are other commonly used examples, as well as driving an optimised conceptual design model. The possibilities are endless.

Dynamo driving a piling schedule to MS Excel

Image Credit: Johan Cantryn - Datech

In a way this is very reminiscent of using AutoLISP with AutoCAD. For tedious and repetitive tasks, using scripting in any project becomes an essential part of any project and to maximise profitability and ensure compliance with data standards in BIM execution plans.