Dynamo for Structure

Marcello Sgambelluri
John A. Martin Structural Engineers

Learning Objectives

- Learn how to build structural beams, including roof framing, using Dynamo in Revit
- Learn how to get and set structural parameters in Revit using Dynamo
- Learn how to speed up the structural modeling process in Revit by using Dynamo
- Learn how to create complex structural shapes in Revit using Dynamo

Description

Very few classes focus on how Dynamo software could be used in the structural design office from a practical level—until now. This class will teach attendees how to use Dynamo to help model structural elements in Revit software, how to quality-control Revit model and analytical models, and even show how Dynamo could be used to help annotate structural drawings. If you model any type of Revit structural elements, then this class is for you. We will make modeling structure in Revit fun again!
Speaker

Marcello Sgambelluri currently serves as the BIM Director at John A. Martin & Associates Structural Engineers in Los Angeles. Marcello has worked on many BIM projects over the last 18 years as a project manager, design engineer, and BIM Director. Some of the BIM projects Marcello has worked on includes the Walt Disney Concert Hall in Los Angeles - CA, the Ray and Maria Stata Technology Center at MIT, Tom Bradley International Terminal Expansion at LAX. Marcello is internationally recognized at one of the top BIM leaders and contributors to the education and implementation of BIM technology in the building industry. Marcello continually speaks at Autodesk University and the Revit Technology Conference (BILT) where he has received the 1st place speaker award for a record 12 times between 2012 thru 2016 between both conferences. Marcello received his Bachelors and Master’s degrees in Civil Engineering and he is also a licensed Civil and Structural Engineer.
Introduction

What is Dynamo?

This is a very difficult question to answer simply because Dynamo does SO much. Below is my answer to this question and I hope it clarifies it for some of you.

Dynamo is a free program from Autodesk that uses visual programming (or boxes and wires). Dynamo primarily does two tasks:

1. **Creates its own geometry with parametric relationships.**
2. **Reads and writes to and from external databases.**

Since Dynamo creates its own geometry and reads and writes to external databases it is a perfect fit to interact with Revit because....isn’t Revit simply a database with parametric geometry?

Dynamo reads and writes back data to and from the Revit database via the Revit API. The data could be just about anything, parameter values, family geometry, and family placement.
Why Dynamo?
Why not use Dynamo? You don’t need to use Dynamo all the time but consider the following image and let that guide your decision on when to use dynamo and when not to.

Every Revit User Gets to Touch the API
Any Revit user could learn Dynamo (since it uses visual programming it is very simple to learn) they could access the Revit API and perform simple tasks that only could have previously been performed with writing an add-on or writing a macro with .net language for Revit.
How this Handout is organized
Each example is set up in a “picture” format meaning all you need to know on how to perform an example is completely contained within the image and not in the supporting text.

I felt that making this handout in the “picture book” format you are could easily understand what to do and not have to read many lines of text to get the “step by step” method on how to perform the examples. I call this the “DynamoCheatSheet”
See the image below to understand how the DynamoCheatSheet is structured. I hope you like this new picture method format!
Lab Exercises

Download the dataset and handout here
http://a360.co/2x3FC6g
TOTAL SURFACE AREA OF STRUCTURAL BEAMS

**Family Types**
- W Shapes:W12X26 → Family Type

**Element.Solids**
- element → Solid[]

**All Elements of Family Type**
- Family Type → Elements

**Solid.Area**
- solid → double

**Flatten**
- list → var[]..[]

**Math.Sum**
- values → sum

CONVERTS THE REВIT GEOMETRY OF BEAM TO DYNAMO SOLIDS

SELECTS ALL THE INSTANCES OF BEAM TYPE

GETS THE SURFACE AREA OF EACH SOLID

FLATTENS THE AREA LIST TO 1 LIST

SUMS THE LIST OF EACH BEAM AREA

---

STEP 1: OPEN FILE "BEAM_AREA_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: ALT. METHOD IS TO EXTRACT SOLID AND GET SURFACE AREA AND COULD BE USED ON MOST REВIT SOLID AND SURFACE ELEMENTS
ALIGN REVIT BEAMS WITH AN OFFSET WITH DYNAMO PLAYER

**Select Model Element**
- Change: 388863
- Element: 

**Element.SetLocation**
- element: void
- geometry: 

**Select Edge**
- Change: Curve
- Edge of Element Id: 815786

**Integer Slider**
- Min: -1
- Max: 1
- Step: 2

**Dynamo Nodes**
- Selects the beam to align
- Note: Selection UI nodes become input in Dynamo player

**Dynamo Player**
- Offset curve from edge of slab by "1" note: sliders become input in Dynamo Player

**Revit Geometry with Dynamo Player**
- Beam to align 1'-0" from edge

**Steps**
1. Open file "ALIGN_OFFSET_START.RVT", open new Dynamo, add nodes, and save
2. Open Dynamo Player and navigate to folder with .DYN file
3. Select the "Edit Inputs" on player, select slab edge and beam and click run
CREATE BEAMS FROM 2-PT AC LINE FAMILIES

DYNAMO NODES

SELECT ALL THE INSTANCES OF ADAPTIVE COMPONENT FAMILY TYPES

CONVERTS/EXTRACTS THE CURVES FROM THE AC FAMILY

CREATES BEAMS OF SPECIFIED TYPES AT THE AC LINE FAMILY

REVIT GEOMETRY AND STEPS

STEP 1

CREATE A 2-POINT AC FAMILY AND LOAD INTO PROJECT

STEP 2

OPEN "BEAMS_AC_LINES_START.RVT"

STEP 3

PLACE 2-POINT AC FAMILIES IN PROJECT AS ROOF FRAMING

STEP 4

PLACE DYNAMO NODES AND DYNAMO WILL PLACE BEAMS ON AC LINES

NOTES

CONSIDER USING A 2-PT AC FAMILY FOR ROOF FRAMING IN LIEU OF BEAM FRAMING. PLACE THE BEAMS ON THE AC FAMILIES VIA DYNAMO.
SETTING MIN CONCRETE FLOOR THICKNESS USING ENGINEERING LOGIC

DYNAMIC NODES

- SELECTS DIMENSION STRING AND GETS VALUES
- SELECTS SLAB ELEMENT AND ELEMENT TYPE
- SETS THE FLOOR SLAB THICKNESS TO MIN
- SETS MIN THICKNESS PER RULES OF THUMB AND GETS MAX VALUE

REVIT GEOMETRY

STEPS

STEP 1: OPEN "ONE_WAY_BeAM_SLAB_SYSTEM_START.RVT"
STEP 2: OPEN NEW DYNAMO AND ADD NODES
STEP 3: SELECT SLAB AND SELECT LOWER DIMENSION STRING W/ "SELECT" AND RUN DYNAMO
CHANGE BEAM TYPES USING ENGINEERING LOGIC

DYNA MO NODES

Categories
Structural Framing → Category

All Elements of Category
Category → Elements

Code Block
"Length";

Element.GetParameterValueByName
element → Element
parameterName → var[1].1
value →

Code Block
"Family and Type";

Switches the beam framing type (see note below)

CALCULATES THE MIN BEAM DEPTH PER ENGINEERING RULES OF THUMB

Math.Ceiling
number → integer

Code Block
(1/24)*12;

SET S THE BEAM SIZES BASED ON MIN VALUE FOR JUST 2 BEAM TYPES (NOTE: MORE BEAMS SIZES COULD BE USED BY USING MORE IF STATEMENTS)

Structural Framing Types
W12X19 → Framing Types
W24X55 → Framing Types

STEP 1: OPEN "CHANGE_BM_TYPE_LOGIC_START.RVT"
STEP 2: OPEN A NEW DYNA MO FILE AND ADD NODES AND RUN
NOTES: THE "FAMILY AND TYPE PARAMETER" IS USED TO CHANGE THE FAMILY TYPE
ALSO CONSIDER USING YOUR OWN ENGINEERING RULES OF THUMB

BEAM SIZES BEFORE CHANGE

BEAM SIZES AFTER CHANGE

STEPES & NOTES
BUILD A COMPLEX BEAM IN DYNAMO AND SEND TO REVIT

STEPS:

STEP 1: OPEN FILE “CANOPY_BEAMS_TO_REVIT_START.RVT”, OPEN NEW DYNAMO, ADD NODES

STEP 2: SELECT THE CANOPY EDGE USING THE “SELECT EDGE” NODE AS SHOWN AND RUN

THIS CUSTOM NODE FROM THE “SPRING” PACKAGE CREATES AND PLACES A REVIT LOADDABLE FAMILY FROM SOLID GEOMETRY.
ORIENT BEAMS PERP TO SURFACE

NOTE THE V,U PARAMETERS MATCH THE REVIT SURFACE
ALSO NOTE THE LIST AT LEVEL IS ACTIVATED HOWEVER "CROSS
PRODUCT LACING ALSO WORKS

THESE NODES CREATE THE LIST FOR THE POINTS AT EACH END

THESE NODES SELECT THE SURFACE AND AC FAMILY

THIS NODE HOSTS 2-PT AC BEAM ON SURFACE

REVIT FINAL GEOMETRY AND U V COORDINATES OF REVIT ROOF SURFACE

STEP 1: OPEN "BEAM_ORIENTATION_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, AND ADD NODES
STEP 3: SELECT THE SURFACE USING THE "SELECT FACE" NODE
STEP 4: SELECT THE "4X2" FAMILY TYPE AND RUN - NOTE BEAMS ARE ADAPTIVE COMPONENTS
CONVERT REVIT ANALYTICAL BMS + PTS TO SOLIDS FOR RENDERING IN A GAME ENGINE (SIMPLE METHOD)

ANALYTICAL NODES

SELECTS ALL ANALYTICAL NODES IN THE PROJECT
CONVERTS ANALYTICAL NODES TO DYNAMO NODES
ADD Spheres TO ALL ANALYTICAL NODES
IMPORTS ALL SOLIDS INTO REVIT VIA SAT IMPORT

ANALYTICAL BMS

SELECTS ALL ANALYTICAL BMS IN THE PROJECT
CONVERTS ANALYTICAL BMS TO DYNAMO CURVES AND JOINS THEM
ADDs CYLINDERS TO ALL ANALYTICAL LINES
IMPORTS ALL SOLIDS INTO REVIT VIA SAT IMPORT

ANALYTICAL LINES IN REVIT

ANALYTICAL MODEL CONVERTED TO SOLIDS IN REVIT

ANALYTICAL REVIT SOLID MODEL RENDERED IN A GAME ENGINE

STEP 1: OPEN “ANALYTICAL_MODEL_START.RVT”
STEP 2: OPEN A NEW DYNAMO FILE AND ADD NODES AS SHOWN AND RUN SCRIPT
STEP 3: ISOLATE AND OPEN OR EXPORT TO GAME ENGINE
NOTE: GAME ENGINES WILL NOT RENDER SINGLE LINES AND NODES THUS NEED METHOD ABOVE
Additional Examples
CHANGE LOADABLE FAMILY TYPES

1. THESE NODES SELECT ALL THE FAMILY INSTANCES IN REVIT...

2. Code Block
   "Family and type"; 
   Element.SetParamByName
   element
   parameterName
   value

3. THIS NODE SETS THE NEW FAMILY TYPE

4. THIS NODE SETS THE PARAMETER TO BE CHANGED ALSO SEE NOTES BELOW

5. THIS NODE SETS PARAMETER TO THE NEW FAMILY TYPE

STEP 1: OPEN "CHANGE_BM_TYPE_START.RVT"
STEP 2: OPEN NEW DYNAMO FILE AND ADD NODES AND RUN
NOTES: THE FAMILY AND TYPE PARAMETER SELECTS THE TYPE IS OF EACH FAMILY INSTANCE AND THUS ALLOW THE TYPE TO BE CHANGED
SURFACE AREA OF RAMPS

1. Select Model Element
   - Change Elements
   - Element: 309385
   - This node selects the ramp

2. Element.Faces
   - Get all the surfaces of the ramp
   - This node extracts all the surfaces of the ramp

3. Code Block
   - Code: `element.SetSurface(0)`
   - This node "gets" the top surface or index [3]

4. Surface.Area
   - Surface area: 107.932
   - This node extracts the surface area of the "top" surface of the ramp

---

STEP 1: OPEN FILE "RAMP_AREA_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: TO GET THE AREA OF THE SIDES AND BOTTOMS, SIMPLY DELETE THE TOP SURFACE AREA FROM THE TOTAL SURFACE AREA

---

NOTE: IF THE RAMP IS STRAIGHT USE [0]

NOTE: USE THE METHOD ABOVE TO EXTRACT OTHER AREAS FROM REVIT SOLIDS SUCH AS ROOFS, FLOORS, FOUNDATION SLABS, ETC....
SURFACE AREA OF RAMP FROM A LINKED FILE

**THIS NODE SELECTS THE RAMP**

**THESE NODES SELECT THE RAMP FROM THE LINKED FILE (THIS IS A CUSTOM NODE FROM THE STEAM NODES PACKAGE)**

**THIS NODE EXTRACTS ALL THE SURFACES OF THE RAMP**

**THIS NODE "GETS" THE TOP SURFACE OR INDEX [3]**

**RAMP GEOMETRY**

**STEP 1: OPEN FILE “RAMP_AREA_START_LINK.RVT”**

**STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT**

**NOTES:** THIS CUSTOM NODE ALLOWS DYNAMO TO ACCESS LINKED FILES IN REVIT AS "READ ONLY FILES" WHICH HAS A WIDE RANGE OF APPLICATIONS
TOTAL SURFACE AREA OF MECH EQUIPMENT

<table>
<thead>
<tr>
<th>Select Model Elements</th>
<th>Element.Faces</th>
<th>Flatten</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change</td>
<td>Elements</td>
<td></td>
</tr>
<tr>
<td>Elements:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4826 4898 4921 5019 5095</td>
<td>element</td>
<td>var[]...[]</td>
</tr>
<tr>
<td>5147 5176 5242 5261 5308 5343</td>
<td>Surface[]</td>
<td></td>
</tr>
<tr>
<td>5382 5428 5463 5486 5499 5508</td>
<td>list</td>
<td></td>
</tr>
<tr>
<td>5557 5574 5634</td>
<td>var[]...[]</td>
<td></td>
</tr>
</tbody>
</table>

**THIS NODE SELECTION:**
ALL THE ELEMENTS IN REVIT

**THIS NODE:**
CONVERTS ALL THE REVIT FAMILY GEOMETRY TO DYNAMO SURFACES

**THIS NODE FLATTENS:**
ALL THE ITEMS INTO A SINGLE LIST

---

**Dynamo Nodes**

**Dynamo Geometry**

**Revit Geometry**

**Steps & Notes**

STEPT 1: OPEN FILE "SURFACE_AREA_ROBOT_START.RVT"

STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT

NOTES: ALT. METHOD IS TO EXTRACT SOLID AND GET SURFACE AREA AND COULD BE USED ON MOST REVIT SOLID AND SURFACE ELEMENTS
CANOPY BEAM FRAMING ON GRIDS (USING PROJECTION)

SELECT THE ROOF FACE AND GRID LINES

CONVERT THE GRIDS TO DYNAMO CURVES

PROJECT THE CURVES ONTO THE ROOF SURFACE

BEAM TYPES AND LEVELS
NOTE: LEVEL IS ONLY THE REF. LEVEL

ADDS BEAMS ON CURVES ON SURFACE

STEP 1: OPEN FILE “CANOPY_BEAM.RVT”
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: IF THE STRUCTURAL FRAMING NODE FAILS TO CREATE STRUCTURAL FRAMING THEN
SIMPLIFY THE INPUT CURVE OR CHECK IF THE BEAM IS CURVING IN 2 DIRECTIONS
GET AND SET INSTANCE PARAMETERS

Select Model Element
- Change
- Element
  - Element: 551125

SELECTIONS WALL ELEMENT

Code Block
- "Base Offset":

Element.GetParameterValueByName
- element
- parameterName
  - var[0...-1]

GETS THE BASE OFFSET PARAMETER VALUE

Family Types
- Concrete-Square-Column: 30 x 30

All Elements of Family Type
- Family Type
- Elements

SELECTS ALL THE COLUMN TYPES IN THE PROJECT

Element.SetParameterByName
- element
- parameterName
- value

SETS THE BASE OFFSET PARAMETER OF THE COLUMN TO THE BASE OFFSET PARAMETER OF THE WALL

STEP 1: OPEN FILE "GET_SET_PARAMETERS_START.RVT"

STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT FOR EACH COLUMN

NOTES: THERE ARE MANY OTHER WAYS TO SELECT THE WALL AND COLUMNS
GET AND SET INSTANCE PARAMETERS WITH LINK FILE

Select Model Element
Select Element
Element: 570521

SELECTS LINKED FILE IN REVIT

Element.GetFromLinkedInstance
Link Instance Categories

GETS THE WALL ELEMENTS FROM THE LINKED FILE IN REVIT (AMAZING!) (NOTE: THIS IS A CUSTOM NODE IN THE SPRING NODE PACKAGE)

Category

Walls

SET THE WALL CAT.

Family Types
Concrete-Square-Column: 30 x 30

SELECTS ALL THE COLUMN TYPES IN THE PROJECT

All Elements of Family Type
Family Type: Elements

Code Block
"Base Offset":

DYNAMO GEOMETRY ONLY

Element.SetParameterByName
Element parameterName value

GETS THE BASE OFFSET PARAMETER OF THE COLUMN TO THE BASE OFFSET PARAMETER OF THE WALL

SET THE BASE OFFSET PARAMETER OF THE WALL

REVIT GEOMETRY

STEP 1: OPEN FILE “GET_SET_PARAMETERS_START_LINK.RVT”
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT FOR EACH COL
SELECT THE LINKED FILE USING THE "SELECT MODEL ELEMENT" UI NODE
NOTES: LINKED FILES ARE "READ ONLY"

WALL IS FROM A LINKED FILE!

ORIgINAL LOCATION OF BOTTOM OF COLUMNS

NEW LOCATION OF BOTTOM OF COLUMNS

STEPS & NOTES
ADD FIREPROOFING TO WIDE FLANGE AND PIPE COLUMNS

WIDE FLANGE

PIECE COLUMN

DYNANO

REVIT

STEP 1: OPEN FILE "FIREPROOFING_START.RVT"
STEP 2: OPEN A NEW DYNANO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT FOR EACH COL
NOTES: THERE ARE MANY WAYS TO BRING DYNANO GEOMETRY INTO REVIT AND THIS METHOD SHOWS JUST ONE VIA OOTB NODE FOR DIRECT SHAPE FOR PIPE COL,
FLAT FRAMING TO CURVED (PROJECTION)

- **SELECT THE BOT ROOF SURFACE**
- **PROJECT THE FLAT BEAM CENTERLINES TO THE CURVED ROOF**
- **BUILD THE BEAMS ON THE CURVED ROOF**

**DYNAMO NODES**

**DYNAMO GEOMETRY**

**REVIT GEOMETRY**

**STEPS & NOTES**

**STEP 1:** OPEN FILE "FLAT_FRAMING.RVT"
**STEP 2:** OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
**NOTE:** ALT METHOD: CONSIDER USING THE "SET LOCATION" NODE INSTEAD OF PROJECTION
TRUE LOCATION OF CURVE OF BEAMS

These nodes select and get the solid geometry of the struct beam (note the top of beam location).

This node gets location of the beam curve (note: the geometric "top" of the beam).

This node gets the "Z" offset parameter of the beam (note: the BM curve and BM geometric top are off by this value).

This node moves the beam curve to the top of the geometric BM top center.

BM CURVE LOC

BM GEOMETRIC (SOLID) LOC

Level 01
0' - 0"

The beam curve location will not be at the beam geometric (solid) justification loc (top center in this case) if the "Z" offset value is non-zero. Therefore, you must correct the beam curve loc by the method above.

Also note that this method shown above will not work if your beam reference work plane is not horizontal. In that case you would need to change the vector input from "Z" to another vector that is perpendicular to your work plane but this case is not that common.
COLUMN CENTERLINE

1. This node selects the structural column.
2. This node changes Revit geometry to Dynamo geometry - a solid.
3. This node finds the vertices of the faces of the solid.
4. This node converts the vertices to points.
5. This node places a curve at the center of all the points, or centerline of Col!

Steps & Notes:
STEP 1: OPEN FILE "COL_CENTERLINE_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTE: THE METHOD SHOWN ONLY WORKS ON COLUMNS THAT HAVE A CONSTANT PROFILE AND IS STRAIGHT. ALSO NOTE THAT THE COL CENTERLINE IS THE GEOMETRY C.G.
CREATING SURFACES WITH PROFILE ORDER

**Dynamo Nodes**
- These nodes select the profile curves.
- This node creates a list in the correct profile order.
- This node creates the surface lofted over the correct profile order.

**Dynamo Geometry**

**Revit Geometry**

**Steps & Notes**

Step 1: Open Revit file "PROFILEORDER_START.RVT"
Step 2: Open a new Dynamo file and add nodes as shown.
Step 3: Select element 0, 1, 2 in that order using the "Select" node and run.

Note: The Revit Massing family does not allow for profiles to be selected in order.
FIND FAMILIES THAT HAVE EMBEDDED MESHES/DWG

THIS NODE SELECTS THE FAMILIES
(OR USE A DIFFERENT FAMILY SELECTION METHOD)

CONVERTS ALL REVIT FAMILY GEOMETRY TO DYNAMO GEOMETRY
(INCLUDING MESHES)

CONVERTS LIST TO STRINGS

DETERMINES WHICH LIST HAS "MESH" (USE @@L2)

FILTERS OUT WHICH FAMILIES HAVE "MESH" AND "NO MESH"

THESE FAMILIES HAVE NO MESH

THESE FAMILIES HAVE MESH!
MESH IS MOST LIKELY FROM AN IMPORTED .DWG FILE IN THE FAMILY

BUSTED!!! :

THESE DINING TABLE FAMILIES CONTAIN MESHES WHICH ARE IMPORTED DWG'S

STEP 1: OPEN FILE "MESHFAMILYEMBED_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: SOME FAMILIES ALWAYS CONTAIN MESHES SUCH AS TOPO AND RPC
METHOD WILL NOT FIND MESHES THAT ARE HIDDEN OR NOT VISIBLE
SELECT/FIND IMPORTED DWG

1. Element Type
   - Importance
   - Type

2. All Elements of Type
   - Element
   - Parameter
   - Importance
   - Type

3. This node lists all the parameters of the DWG's

4. Code Block
   - "Shared Site"

5. These nodes filter the parameters to determine if "Share Site" exists and returns a filtered list that creates a list of only imported DWG

6. List: [importance, list
   - In
   - Mask
   - Out

7. List of imported DWG's

NOTE: Since all linked DWG (show "Shared Site" parameter

IN DYNAMO, filter all DWG in REVIT such that only instances not showing the "Shared Site" parameter are selected

ALL THESE DWG LOOK IDENTICAL
3 WERE INSERTED BY "IMPORT"
4 WERE INSERTED BY "LINK"

THESE ARE IMPORTED DWG!!!

NOTE: This method selects the imported DWG's and is important since it is difficult in some cases to "find" imported DWG's in a Revit file since there is no good ways to "manage" imported DWG's. Most prefer to "link" DWG files since they could be managed

29 | Page
SET TYPE PARAMETER IN A LOADABLE FAMILY

1. Family Types
   - M_Fixed: 4ft x 4ft

2. Code Block
   - "Width";

3. Code Block
   - 4;

4. Element.SetParameterByName
   - Element
   - parameterName
   - value

Then this node SETS the Type Parameter.

This node selects the family TYPE.

STEP 1: OPEN FILE "SET_TYPE_PARAMETER_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTE: THIS METHOD WILL NOT WORK FOR A SYSTEM FAMILY
KEEP NORTH ARROW ALIGNED WITH TRUE NORTH

1. SELECT NORTH ARROW FAMILY
2. SELECT PROJECT BASE POINT
3. ENTER FAMILY PARAMETER NAME
   (NOTE: CREATE ROT PARAMETER IN NORTH ARROW FAMILY WITH THE SAME NAME)
4. GET THE ANGLE OF THE TRUE NORTH FROM BASE PT
5. SET ANGLE OF THE BASE PT TRUE NORTH ANGLE TO NORTH ARROW ANGLE

PROJECT NORTH PLAN VIEW

STEP 1: OPEN FILE "NORTH_ARROW_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTE: CREATE A NORTH ARROW FAMILY WITH A ROTATION PARAMETER
### Levels in Revit

#### DynamoBim

<table>
<thead>
<tr>
<th>Step (Level HT)</th>
<th>Number Range</th>
<th>Start (Elevation)</th>
<th>End (Elevation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100</td>
<td>19</td>
<td>Code block</td>
</tr>
</tbody>
</table>

This node creates levels by elevation.

#### Revit

<table>
<thead>
<tr>
<th>Level</th>
<th>Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 7</td>
<td>80’ - 0”</td>
</tr>
<tr>
<td>Level 6</td>
<td>50’ - 0”</td>
</tr>
<tr>
<td>Level 5</td>
<td>40’ - 0”</td>
</tr>
<tr>
<td>Level 4</td>
<td>30’ - 0”</td>
</tr>
<tr>
<td>Level 3</td>
<td>20’ - 0”</td>
</tr>
<tr>
<td>Level 2</td>
<td>10’ - 0”</td>
</tr>
<tr>
<td>Level 1</td>
<td>0’ - 0”</td>
</tr>
</tbody>
</table>

**Note:** If first level is created manually, rest created by DynamoBim will autorename in order.

---

**Steps & Notes**

1. Open file "LEVELS_START.RVT"
2. Open a new Dynamo file, place nodes as shown and run script.

**Note:** Design scripts are used to ranged expressions in method above.
GRIDS IN REVIT

DESIGN SCRIPT

```
Code Block
Point.ByCoordinates(-50...50..10,50,0);
Point.ByCoordinates(-50...50..10,-50,0);
```

GRID

CREATES GRID IN REVIT

START POINT

NUMBER RANGE

END POINT

NOTE: IF YOU CHANGE THE NAME OF THE FIRST GRID NUMBER DYNAMOBIM WILL AUTO RENUMBER GRIDS!

STEP 1: OPEN FILE "GRIDS_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT

NOTE: DESIGN SCRIPTS ARE USED TO RANGED EXPRESSIONS IN METHOD ABOVE
BOTTOM OF PIPE

These Nodes Select all the Pipes in the Project. Note: use "Select Model Element" Node if you don’t want all Pipes.

This Node Converts Revit Geometry to Dynamo Surface Geometry

This Node Isolates the Bottom Pipe Surface

This Node Flattens the Single List

This Node Adds a Point to Each End of the Pipes Bottom Surface

This Node Adds a Line Between the Bottom Points on the Pipe Bottom Surface

Added Points to Bottom of Pipe at Each End

STEP 1: OPEN FILE "BOT_OFPIPE_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT

NOTE: IF THE END POINTS OF THE PIPE IS NEEDED THEN ADD POINTS TO THE END OF THE LINE
REVIT 4 SIDED PYRAMID

These nodes create the 5 points for the tetrahedron and arranges them in lists to define vertices of faces. Note that this example uses a pyramid that is 1 unit x 1 unit x 1 unit.

These nodes create the faces from the vertices (5 total faces) and creates a list.

This node creates the solid pyramid from the 5 faces (must make a closed shape to make a solid).

These nodes take the solid tetrahedron in Dynamo and creates a Revit family (in this case in the wall category). This custom node is from the "Springs" package.

NOTE:
This creates a pyramid in Dynamo then in Revit at 1X1X1.
DYNAMO AND REVIT POINTS
W/ SCALE VIA DESIGN SCRIPT

Code Block

```dynamo
p1=Autodesk.Point.ByCoordinates(0,0);
p2=Autodesk.Point.ByCoordinates(0,25);
p3=Autodesk.Point.ByCoordinates(25,25);
p4=Autodesk.Point.ByCoordinates(25,0);
points={p1,p2,p3,p4};
scaledpoints=points.Scale(3);
ReferencePoint.ByPoint(scaledpoints);
```

Steps & Notes

Step 1: Open a new Mass Family Template
Step 2: Open a new Dynamo Graph and add nodes and run
Note: This creates Dynamo Points and scales 3x and creates Revit Reference Points in the Massing Editor at 3x scale
REVIT 3D ROOMS VIA DIRECT SHAPE

DYNAMO NODES

DYNAMO GEOMETRY

REVIT GEOMETRY

STEP 1: OPEN REVIT SAMPLE PROJECT “rac_advanced_sample_project.rvt”
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: ROOMS IN REVIT ARE 2D ELEMENTS, THIS METHOD CONVERTS THE ROOMS TO 3D
ALSO THERE ARE MANY OTHER METHODS TO GET DYNAMO GEOMETRY INTO REVIT
STEP 1: OPEN REVIT SAMPLE PROJECT "rac_advanced_sample_project.rvt"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: ROOMS IN REVIT ARE 2D ELEMENTS, THIS METHOD CONVERTS THE ROOMS TO 3D
ALSO THERE ARE MANY OTHER METHODS TO GET DYNAMO GEOMETRY INTO REVIT
VIEW REVIT 3D MEP SPACES IN DYNAMO

STEP 1: OPEN REVIT SAMPLE PROJECT “rac_advanced_sample_project.rvt”
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT

NOTES: MEP SPACES IN REVIT ARE 2D ELEMENTS, THIS METHOD CONVERTS THE ROOMS TO 3D ELEMENTS IN DYNAMO ONLY
REVIT 3D MEP SPACES VIA IMPORT INSTANCE

STEP 1: OPEN REVIT SAMPLE PROJECT “rac_advanced_sample_project.rvt”
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: MEP SPACES IN REVIT ARE 2D ELEMENTS, THIS METHOD CONVERTS THE ROOMS TO 3D
ALSO THERE ARE MANY OTHER METHODS TO GET DYNAMO GEOMETRY INTO REVIT
STACKING DIAGRAMS USING 3D ROOMS

STEPS & NOTES

STEP 1: OPEN FILE "STACKING_DIAGRAMS_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT

NOTES: COLO/MATERIALS WILL SOMETIMES NEED TO BE ADJUSTED OR MAPPED
DWG TO GRID

1. Select Model Element
   Change: Element
   Element: 308812
   This node selects the DWG

2. Element.Curves
   This node changes the DWG into Dynamo Curves

3. CurveEndPoint
   curve: Points
   These nodes extract the start and the end points of the Dynamo Curves

4. Curve.StartPoint
   curve: Points
   This node creates grids in REVIT based on start and end points

5. Grid.ByStartPointEndPoint
   start: Grid
   end: Grid
   REVIT GEOMETRY

GRID GENERATED IN REVIT

STEP 1: OPEN FILE "DWG_TO_GRID_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN
STEP 3: LINK "DWG_GRID.DWG" INTO REVIT AND SELECT WITH "SELECT" NODE

NOTES: ISOLATE LAYERS IN DWG BEFORE LINKING REVIT AND TRY TO CREATE OTHER ELEMENTS
TEXT TO UPPERCASE PLAN VIEW NAMES START

STEP 1: OPEN REVIT SAMPLE PROJECT "TEXT_TO_UPPERCASE_PLAN_VIEW_NAMES_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT

NOTES: THE METHOD ABOVE WORKS WELL WHEN CHANGING MANY VIEW NAMES TO UPER CASE
TEXT TO UPPERCASE SHEET NAMES

1. THIS NODE SELECTS THE SHEET CATEGORY
2. THIS NODE SELECTIONS ALL THE SHEETS IN THE REVIT PROJECT
3. THIS NODE GETS THE SHEET NAME'S
4. THIS NODE CHANGES THE TEXT IN THE SHEET NAME TO UPPERCASE
5. THESE NODES SET ALL THE SHEET NAMES TO UPPERCASE

STEP 1: OPEN REVIT SAMPLE PROJECT “TEXT_TO_UPPERCASE_SHEET_NAMES_START.RVT”
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: THE METHOD ABOVE WORKS WELL WHEN CHANGING MANY SHEET NAMES TO UPPERCASE
TEXT TO UPPERCASE SHEET NAMES

1. This node selects the room category.
2. This node selects all the instances of rooms in the Revit project.
4. Element.GetParameterByName.
5. String From Object.
6. String To Upper.
7. Element.SetParameterByName.

Steps & Notes:

Step 1: Open Revit sample project "TEXT_TO_UPPERCASE_ROOM_TEXT_START.RVT"
Step 2: Open a new Dynamo file, place nodes as shown and run script.
Notes: The method above works well when changing many room text notes to upper case.
CREATE PROPERTY LINES ON TOPO IN 3D

STEP 1: OPEN REVIT SAMPLE PROJECT "PROPERTYLINES_3D_START_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: PROPERTY LINES IN REVIT ARE 2D AND THIS METHOD TURNS THEM INTO
3D ELEMENTS WITH MODEL LINES AND CREATES THEM TO FOLLOW TOPO
CREATE PIPING ON TOPO IN 3D

STEP 1: OPEN REVIT SAMPLE PROJECT "PIPE_TOPO_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: THIS METHOD PLACES MODEL LINES ON TOPO AND CREATES MEP PIPING
REPEAT A LOADABLE FAMILY NOT AC

1. Select Model Element
2. Dynamic Curves
3. Curve Points/Parameter
4. Family Types
5. Family Instance by Points
6. This Node Selects the Curve/Path in Revit

This Node Converts the Revit Curve to Dynamo Geometry
This Node Defines the Number of Divisions Using a "Number Series"
This Node Divides the Curve/Path by a number of Points defined in Node 3 Sim. to "Divide"
This Node Selects ANY loadable family type in the project
This Node assigns the Selected Family Type to the Curve/Path at each Point Sim. to a "Repeater"

Note: Do not confuse this Parameter with Family Parameter. This "Parameter" means "location" along a curve/path as a normalized length... meaning 0 is start and 1 is end and... 0.5 is middle... etc...

55 Planting Families (not AC Families) Repeated along Curve/Path in Revit

STEP 1: OPEN REVIT SAMPLE PROJECT "REPEATER_LIMIT_START.RVT"
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTES: DYNAMO ALLOWS LOADABLE FAMILIES TO BE PLACED IN REPEATER PATTERNS
DIVIDE A PATH GREATER THAN 200 LIMIT FROM REVIT

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>This Node Selects the Curve/Path in Revit</td>
<td>This Node Converts the Revit Curve to Dynamo Geometry</td>
<td>This Node Defines the Curve/Path by a number of Points defined in Node 3 Sim. to &quot;Divide&quot;</td>
<td>This Node Selects the Adaptive Component Family Type</td>
<td>This Node assigns the Adaptive Component to the Curve/Path at each Point Sim. to a &quot;Repeater&quot;</td>
</tr>
</tbody>
</table>

**Dynamo Nodes**

**Revit Geometry**

**300 Adaptive Component "posts" "Repeated" along a Curve/Path in Revit**

**Road In Revit**

**Steps & Notes**

STEP 1: OPEN REVIT SAMPLE PROJECT "REPEATER LIMIT 300 START.RVT"

STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT

NOTES: DYNAMO ALLOWS LOADABLE FAMILIES TO BE PLACED IN REPEATER PATTERNS GREATER THAN THE LIMIT OF REVIT DIVIDED PATH AND ARRAY
TEXT NOTES

DYNAMO NODES

- Views
  - Level 1
- Point Origin
  - Point
- Code Block
  - Text "TEXT_OOTB"

DYNAMO

- TextNote By Location
  - View
  - Location
  - Text
  - Alignment
  - Type
  - Keep Rotated Text Readable
  - Rotation

REVIEW

- Horizontal Text Alignment: Left

NOTE

This amazing OOTB Dynamo node creates a text note object in Revit. Note that the input ports "Type", "Keep", and rotation are using default values (see notes).

Dynamo Text Preview

- TEXT_OOTB

(DAY DYNAMO DOES TEXT PREVIEWS NOW!)
HOW DYNAMO SELECTS REVIT ELEMENTS

When you select a Revit Element using DynamoBIM it displays the Element ID

Select Model Element

Select Elements

Element: 1324093

When you select a Revit Element using Revit it turns it BLUE

NOTE: THIS SHOWS HOW DYNAMO SELECTS REVIT ELEMENTS, ONCE THE REVIT ID IS DISPLAYED IN DYNAMO THEN DYNAMO HAS SELECTED THE REVIT ELEMENT
GET WORKSHARING STATUS OF REVIT

STEP 1: LOCATE THE REVIT FILE USING THE "FILE PATH" NODE
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTE: THIS METHOD WILL EXTRACT THE WORKSHARING STATUS OF THE REVIT PROJECT FILE USING THE "TEXT" WITHIN THE .RVT FILE

1. **File Path**
   - This Node Selects the Revit File
   - Path: E:floor_0002.rvt

2. **File.FromPath**
   - Read the Text from the Revit File
   - (Sim to opening up .rvt in notepad)

3. **File.ReadText**
   - These nodes read the text from the Revit file

4. **Code Block**
   - These nodes tell DynamoBIM to only collect text between the words "Worksharing and User" ("10" Designate Spaces)

5. **Code Block**
   - This node splits the text based on nodes 4 and 5

6. **String.Split**
   - This node extracts (1) item on the list and will display Worksharing Status as:

7. **Code Block**
   - A: Not Worksharing
   - B: Central
   - C: Local
GET REVIT VERSION AND BUILD

1. This Node Selects the Revit File
   - File Path Browse... >
   - E:\floor-0002.rvt

2. These Nodes Read the Text From the Revit File
   (Sim to opening up .rvt in notepad)
   - File.FromPath path file
   - File.ReadText file string

3. These nodes Tell DynamoBIM to only collect Text between The Words "Revit Build and Last" ("\0" Designate Spaces)
   - Code Block

4. This node Splits the Text based on Nodes 4 and 5
   - String.Split
   - str
   - separator0
   - separator1

5. This node extracts (1) item on the list
   - Code Block
   - Autodesk Revit 2013
     Build: 201103_2115

STEPS & NOTES

STEP 1: LOCATE THE REVIT FILE USING THE "FILE PATH" NODE
STEP 2: OPEN A NEW DYNAMO FILE, PLACE NODES AS SHOWN AND RUN SCRIPT
NOTE: THIS METHOD WILL EXTRACT THE REVIT VERSION AND BUILD NUMBER OR THE REVIT PROJECT FILE USING THE "TEXT" WITHIN THE .RVT FILE
DYNAMO POINT

Code Block

ADD A "DOUBLE" OR NUMBER TO THE INPUT PORT

Point.ByCoordinates

CREATES A DYNAMO POINT IN DYNAMO AT 1,1,1

NOTE
THIS CREATES A DYNAMO POINT AT X,Y,Z BY DEFAULT (NO DATA ENTERED INTO THE INPUT PORTS) THEN THE POINT WOULD BE CREATED AT 0,0,0 ALSO NOTE THIS IS NOT!!! REVIT GEOMETRY (NO REVIT POINTS WERE CREATED!!)
DYNAMO LINE BY START AND END PTS

ADD A "DOUBLE" OR NUMBER TO THE INPUT PORT

CREATES A DYNAMO POINT IN DYNAMO AT 1,1,1

CREATES A DYNAMO LINE BY START AND END PTS

CREATES A DYNAMO POINT IN DYNAMO AT 0,0,0

NOTE
THIS CREATES A DYNAMO LINE BY END POINTS, AND THERE ARE OTHER WAYS TO CREATE A LINE DYNAMO REFERS TO STRAIGHT CURVES AS LINES (THEREFORE LINES ARE CURVES)
ALSO NOTE THIS IS NOT REVIT GEOMETRY
CURVES: START AND END POINTS

Point.ByCoordinates
x > Point
y > Point
z > Point

Line.ByStartPointEndPoint
startPoint > Line
endPoint > Line

Curve.StartPoint
curve > Point

Curve.EndPoint
curve > Point

CREATES POINTS AT THE START AND END OF THE LINE

NOTE
A LINE IS A CURVE AND THEREFORE ALL "LINE" OUTPUT PORTS COULD BE CONNECTED INTO A "CURVE" INPUT PORT
CURVES: START AND END POINTS USING PARAMETERS

**Dynamo Nodes**

- **Point.ByCoordinates**
  - \( x \rightarrow \) Point
  - \( y \rightarrow \)
  - \( z \rightarrow \)

- **Code Block**
  - \( 0; \rightarrow \)

- **Line.ByStartPointEndPoint**
  - \( startPoint \rightarrow \) Line
  - \( endPoint \rightarrow \)

- **Curve.PointAtParameter**
  - \( curve \rightarrow \) Point
  - \( param \rightarrow \)

**Dynamo Geometry**

- **Start Point**
  - Parameter = 0

- **Start Point**
  - Parameter = 1

**Note**

A line is a curve and therefore all "line" output ports could be connected into a "curve" input port. Parameter is a value from 0 to 1 (start and end) 0 = end (for example).
CURVES: OFFSET

CREATE A LINE AT 0,0,0 TO 1,0,0

CREATE OFFSET OF THE CURVE IN THE DEFAULT DIR

NOTE
A LINE IS A CURVE AND THEREFORE ALL "LINE" OUTPUT PORTS COULD BE CONNECTED INTO A "CURVE" INPUT PORT.
OFFSET FOR A CURVE ONLY OFFSETS IN THE NORMAL VECTOR (IN THIS CASE (0,-1,0))
CURVES: START AND END POINTS USING PARAMETERS

Point.ById

Line.ById

Curve.ById

Code Block

List

CREATE A LINE AT 0,0,0 TO 1,0,0

CREATE MULTIPLE POINTS ON THE CURVE AT 0.1 PARAMETER STEPS

NOTE

A LINE IS A CURVE AND THEREFORE ALL "LINE" OUTPUT PORTS COULD BE CONNECTED INTO A "CURVE" INPUT PORT. PARAMETER IS A VALUE FROM 0 TO 1 (START AND END) 0 = END (FOR EXAMPLE).
DYNAMO SURFACE BY CURVE EXTRUDE

ADD A “DOUBLE” OR NUMBER TO THE INPUT PORT

CREATE A DYNAMO POINT IN DYNAMO AT 1,1,1

CREATE A DYNAMO POINT IN DYNAMO AT 0,0,0

CREATE A DYNAMO LINE BY START AND END PTS

CREATE A DYNAMO SURFACE BY EXTRUDING A LINE

NOTE
THIS CREATES A DYNAMO SURFACE BY EXTRUDING THE LINE (CURVE). THERE ARE OTHER WAYS TO CREATE SURFACES. ALSO, THE CURVE IS EXTRUDED IN THE POSITIVE DIRECTION OF THE CURVES LOCAL AXIS WITH A DISTANCE DEFAULT OF 1. ALSO TRY NEG VALUES.
DYNAMO SURFACE BY PATCH FROM CLOSED CURVES

**Dynamo Nodes**

- **Rectangle.** ByWidthLength
  - width
  - length

- **Surface.** ByPatch
  - closedCurve

**Dynamo Geometry**

- Creates a rectangle at width and length = 1
- Creates a surface by "filling in" closed perimeter curves

**Revit Geometry**

**Note**

This creates a Dynamo surface by collecting the perimeter curves of the rectangle and "filling in" the the curves with a surface. Also note that the curves must be closed for "patch" to work and a rectangle is a collection of closed curves.
DYNAMO SURFACE SINGLE PT AT PARAMETER U=0.5 V=0.5

- **Surface.ByPatch**
  - `closedCurve` ➔ `Surface`

- **Rectangle.ByWidthLength**
  - `width` ➔ `Rectangle`
  - `length` ➔ `Rectangle`

- **Code Block**
  - `0.5;` ➔ `u`
  - `0.5;` ➔ `v`

- **Surface.PointAtParameter**
  - `surface` ➔ `Point`
  - `u` ➔ `Point`
  - `v` ➔ `Point`

**DYNAMO NODES**

**DYNAMO GEOMETRY**

**NOTE**

This creates a Dynamo point(s) at point at parameter(s) on a surface. Also note that parameters for surfaces are defined by u and v coordinates and range from 0 to 1.
DYNAMO SURFACE SINGLE PT AT PARAMETER U=1 V=1

**DYNAMO NODES**

**Surface.ByPatch**
- `closedCurve` ➔ `Surface`

**Rectangle.ByWidthLength**
- `width` ➔ `Rectangle`
- `length` ➔ `Rectangle`

**Code Block**
- `1;` ➔ `u`
- `1;` ➔ `v`

**Surface.PointAtParameter**
- `surface` ➔ `Point`
- `u` ➔ `u`
- `v` ➔ `v`

 creators a surface at width and length = 1 origin (0,0)

 creAses a point at parameter u = 1 v = 1

**DYNAMO GEOMETRY**

**NOTE**

This creates a Dynamo point(s) at point at parameter(s) on a surface. Also note that parameters for surfaces are defined by u and v coordinates and range from 0 to 1.
DYNAMO SURFACE SINGLE PT AT PARAMETER U=1 V=0

**Surface.ByPatch**
- closedCurve
  - Surface

**Surface.PointAtParameter**
- surface
  - Point
  - U
  - V

**Rectangle.ByWidthLength**
- width
  - Rectangle
- length

**Code Block**
- \[ 1; \]
- \[ 0; \]

**Dynamo Nodes**

**Dynamo Geometry**

**Create a Surface at**
Width and Length = 1
Origin \((0,0)\)

**Create a Point at**
Parameter
\(U = 1\)
\(V = 0\)

**Note**
This creates a Dynamo Point(s) at point at parameter(s) on a surface.
Also note that parameters for surfaces are defined by U and V coordinates and range from 0 to 1.
Dynamo Surface Single PT at Parameter U=0 V=1

**Dynamo Nodes**

**Surface.ByPatch**

- closedCurve ➔ Surface

**Rectangle.ByWidthLength**

- width ➔ Rectangle
- length ➔ Rectangle

**Code Block**

```
θ;
1;
```

**Surface.PointAtParameter**

- surface ➔ Point
- u ➔ Point
- v ➔ Point

**Dynamo Geometry**

Creates a surface at width and length = 1 origin (0,0)

Creates a point at parameter

\[ U = 0 \]
\[ V = 1 \]

**Note**

This creates a Dynamo point(s) at point at parameter(s) on a surface. Also note that parameters for surfaces are defined by u and v coordinates and range from 0 to 1.
DYNAMO SURFACE SINGLE PT AT PARAMETER U=0 V=0

Surface.ByPatch
  closedCurve ➔ Surface

Rectangle.ByWidthLength
  width ➔ Rectangle
  length ➔

Surface.PointAtParameter
  surface ➔ Point
  u ➔
  v ➔

Code Block
  Ø;

CREATE A SURFACE AT WIDTH AND LENGTH = 1 ORIGIN (0,0)

CREATE A POINT AT PARAMETER U = 0 V = 0

NOTE
THIS CREATES A DYNAMO POINT(S) AT POINT AT PARAMETER(S) ON A SURFACE.
ALSO NOTE THAT PARAMETERS FOR SURFACES ARE DEFINED BY U AND V COORDINATES AND RANGE FROM 0 TO 1
DYNAMO SURFACE UV PARAMETER AT POINT

- **Surface.ByPatch**
  - `closedCurve` ➔ `Surface`
  - **Rectangle.ByWidthLength**
    - `width` ➔ `Rectangle`
    - `length` ➔ `Rectangle`

- **Surface.UVParameterAtPoint**
  - `surface` ➔ `UV`
  - `point` ➔ `UV`
  - **Point.ByCoordinates**
    - `x` ➔ `Point`
    - `y` ➔ `Point`
    - `z` ➔ `Point`

Point($X = 0.000$, $Y = 0.000$, $Z = 0.000$)

UV($U = 0.500$, $V = 0.500$)

Create a surface at width and length = 1 origin (0,0)

Create a point at 0,0,0

Get the UV parameter at specific point

**NOTE**
This gets the U V coordinates at a specific point on a surface
Also note that the point must be on the surface
DYNAMO SURFACE NORMAL AT POINT (0,0,0)

- **Surface.ByPatch**
  - Creates a surface at width and length = 1 origin (0,0)

- **Point.ByCoordinates**
  - Creates a point at 0,0,0

- **Surface.NormalAtPoint**
  - Gets the normal (unit vector) at specific point

UNIT VECTOR (0,0,1) (POINTS UP IN "Z" DIRECTION)

**NOTE**
- This gets the normal unit vector at a specific point on a surface
- Also note that the point must be on the surface
DYNAMO SURFACE NORMAL AT PARAMETER U=.5 V=.5

**DYNAMO NODES**

- **Surface.ByPatch**
  - closedCurve > Surface

- **Rectangle.ByWidthLength**
  - width > Rectangle
  - length >

- **Code Block**
  - 0.5; >
  - 0.5; >

**DYNAMO GEOMETRY**

- **Vector(X = 0.000, Y = 0.000, Z = 1.000,)**

**CREATES A SURFACE AT WIDTH AND LENGTH = 1 ORIGIN (0,0)**

**GETS THE NORMAL (UNIT VECTOR) ON A SURFACE AT U = 0.5 V = 0.5**

**UNIT VECTOR (0,0,1) (POINTS UP IN "Z" DIRECTION)**

**NOTE**

This gets the normal unit vector at a specific U V parameter on a surface. Also note that the U V parameter may be different with a Revit surface.
SURFACE MULTIPLE PTS AT PARAMETERS LIST AT LEVEL V

CREATE A SURFACE AT WIDTH AND LENGTH = 1 ORIGIN (0,0)

CREATE MULTIPLE POINTS AT PARAMETERS LACING = SHORT
LIST AT LEVEL @L1 FOR V

NOTE
THIS CREATE MULTIPLE POINTS AT PARAMETERS LACING = SHORT AND LIST AT LEVEL @L1 AT V
SURFACE MULTIPLE PTS AT PARAMETERS LIST AT LEVEL U

Creates a surface at width and length = 1 origin (0,0)

Note:
This creates multiple points at parameters lacing = short and list at level @L1 at U.
Also note that list at level yields the same result at lacing cross product.
SURFACE MULTIPLE PTS AT PARAMETERS LACING SHORT

CREATES A SURFACE AT WIDTH AND LENGTH = 1 ORIGIN (0,0)

NOTE
THIS CREATES MULTIPLE POINTS AT PARAMETERS LACING = SHORT OR LONG
SURFACE MULTIPLE PTS AT PARAMETERS LACING CROSS

CREATE A SURFACE AT WIDTH AND LENGTH = 1 ORIGIN (0,0)

NOTE
THIS CREATES MULTIPLE POINTS AT PARAMETERS LACING = CROSS PRODUCT

CREATE MULTIPLE POINTS AT PARAMETERS LACING = CROSS
CUBIOD LOW AND HIGH PT

`Cuboid.ByCorners`

<table>
<thead>
<tr>
<th>lowPoint</th>
<th>Cuboid</th>
</tr>
</thead>
<tbody>
<tr>
<td>highPoint</td>
<td></td>
</tr>
</tbody>
</table>

creates a CUBIOD at
LOW PT (0,0,0)
HIGH PT (1,1,1)

NOTE
This creates a Dynamo CUBIOD (CUBE) Low Point is Default PT (0,0,0)
High Point is Default (1,1,1)
DYNAMO SOLID BY THICKEN - 1 SIDE - NEG THICKNESS

DYNAMO NODES

- Point.ByCoordinates
  - x
  - y
  - z
  CREATES DYNAMO POINTS

- Line.ByStartPointEndPoint
  - startPoint
  - endPoint
  CREATES A DYNAMO LINE BY START AND END PTS

- Curve.Extrude
  - curve
  - distance
  CREATES A DYNAMO SURFACE BY EXTRUDING A LINE

- Surface.Thicken
  - surface
  - thickness
  - bool
  CREATE A DYNAMO SOLID BY THICKENING A SURFACE ON ONE SIDE VIA BOOL = FALSE AND NEGATIVE THICKNESS

DYNAMO GEOMETRY

REVIT GEOMETRY

NOTE
THIS CREATES A DYNAMO SOLID. THERE ARE OTHER WAYS TO CREATE SOLIDS. ALSO, THE SURFACE IS THICKENED IN THE POSITIVE DIRECTION WITH A THICKNESS DEFAULT OF 1. ALSO TRY NEG VALUES OR CHANGING FROM TWO SIDED THICKEN TO ONE SIDED THICKEN
# Dynamo Solid by Thicken - 1 Side

**Dynamo Nodes**

- **Point.ByCoordinates**
  - Creates Dynamo Points

- **Line.ByStartEndPoint**
  - Creates a Dynamo Line by Start and End Pts

- **Curve.Extrude**
  - Creates a Dynamo Surface by Extruding a Line

- **Surface.Thicken**
  - Creates a Dynamo Solid by Thickening a Surface on one side via Bool = False

**Dynamo Geometry**

**Revit Geometry**

**Note**

This creates a Dynamo Solid. There are other ways to create solids. Also, the surface is thickened in the positive direction with a thickness default of 1. Also try neg values or changing from two sided thicken to one sided thicken.
## Dynamo Solid by Thicken

<table>
<thead>
<tr>
<th>Dynamo Nodes</th>
<th>Dynamo Geometry</th>
<th>Revit Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create Dynamo Points</strong></td>
<td><strong>Create a Dynamo Line by Start and End Pts</strong></td>
<td><strong>NOTE</strong></td>
</tr>
<tr>
<td><strong>Line.ByStartEndPoint</strong></td>
<td><strong>Curve.Extrude</strong></td>
<td><strong>This creates a Dynamo Solid. There are other ways to create solids. Also, the surface is thickened in the positive direction with a thickness default of 1. Also try negative values or changing from two sided thicken to one sided thicken</strong></td>
</tr>
<tr>
<td><strong>Surface.Thicken</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

The surface is thickened in the positive direction with a thickness default of 1. Also try negative values or changing from two sided thicken to one sided thicken.
VOLUME FROM CUBIOD

CUBIOD AT
LOW PT (0,0,0)
HIGH PT (1,1,1)

GETS THE VOLUME FROM THE CUBIOD

NOTE
THIS EXTRACTS THE VOLUME FROM A CUBIOD AND COULD BE APPLIED TO MOST SOLIDS.
SURFACES FROM CUBIOD

CREATE A CUBIOD AT
LOW PT (0,0,0)
HIGH PT (1,1,1)

GETS THE SURFACES
FROM THE CUBIOD

NOTE
THIS EXTRACTS THE SURFACES FROM A CUBIOD AND COULD BE APPLIED TO MOST SOLIDS. ALSO
NOTE THAT IN MOST CASES THE ONLY WAY TO GET SURFACES FROM A SOLID IS TO "EXPLODE" THE SOLID
SURFACE AREAS FROM CUBIOD

Cuboid.ByCorners
lowPoint ➔ Cuboid
highPoint ➔

Geometry.Explode
geometry ➔ Geometry[]

Surface.Area
surface ➔ double

GETS THE SURFACES FROM THE CUBIOD

NOTE
THIS EXTRACTS THE SURFACE AREAS FROM A CUBIOD AND COULD BE APPLIED TO MOST SOLIDS. ALSO NOTE THAT IN MOST CASES THE ONLY WAY TO GET SURFACES FROM A SOLID IS TO "EXPLODE" THE SOLID
POINTS AT VERTICES OF CUBIOD

**CONVERTS THE VERTICES TO POINTS**

**DYNAMO NODES**

**DYNAMO GEOMETRY**

**NOTE**

THIS EXTRACTS THE POINTS AT VERTICES ON A CUBIOD

**CREATES A CUBIOD AT LOW PT (0,0,0) HIGH PT (1,1,1)**

**GETS THE VERTICES FROM THE CUBIOD**
EDGE CURVES ON CUBIOD

Cuboid By Corners
- lowPoint → Cuboid
- highPoint → Cuboid

Face.Edges
- face → Edge[]

Edge.CurveGeometry
- edge → Curve

DYNAMO NODES

CREATES A CUBIOD AT
LOW PT (0,0,0)
HIGH PT (1,1,1)

GETS THE EDGES FROM THE CUBIOD

CONVERTS THE EDGES TO CURVES

DYNAMO GEOMETRY

NOTE
THIS EXTRACTS THE EDGES FROM A CUBIOD AND COULD BE APPLIED TO MOST SOLIDS
CENTER OF MASS FROM CUBIOD

Cuboid.ByCorners

<table>
<thead>
<tr>
<th>lowPoint</th>
<th>Cuboid</th>
</tr>
</thead>
<tbody>
<tr>
<td>highPoint</td>
<td></td>
</tr>
</tbody>
</table>

Solid.Centroid

<table>
<thead>
<tr>
<th>solid</th>
<th>Point</th>
</tr>
</thead>
</table>

Point(X = 0.500, Y = 0.500, Z = 0.500)

CREATE A CUBIOD AT
LOW PT (0,0,0)
HIGH PT (1,1,1)

GETS THE CENTROID
THE CUBIOD

NOTE
THIS EXTRACTS THE CENTROID FROM A CUBIOD AND COULD BE APPLIED TO MOST SOLIDS.
SOLID: 4 SIDED PYRAMID

These nodes create the 5 points for the tetrahedron and arranges them in lists to define vertices of faces. Note that this example uses a pyramid that is 1 unit x 1 unit x 1 unit.

These nodes create the faces from the vertices (5 total faces) and creates a list.

This node creates the solid pyramid from the 5 faces (must make a closed shape to make a solid).

NOTE
This creates a pyramid in Dynamo at 1x1x1.

DYNAMO NODES

DYNAMO GEOMETRY

NOTE
DYNAMO SOLID: SWEEP PROFILE (CIRCULAR)

Select Model Element
Select Element: Element = 32161

Element Curves
element ➔ Curve[]

ADD S A N P L E A T S S T A R T O F C U R V E

Curve PlaneAtParameter
Curve ➔ Plane
param ➔ Plane

GETS THE LOCAL COORDINATE SYSTEM OF THE PLANE ON CURVE

Coordinate System By Plane
plane ➔ Coordinate System

SWEEPS THE CIRCLE "PROFILE" ALONG THE SELECTED CURVE

Circle By Center Point Radius
center Point ➔ Circle
radius ➔ Circle

CREATES THE "PROFILE" CIRCLE AT THE GLOBAL COORDINATE 0,0,0

Geometry Transform
geometry ➔ Geometry

MOVES THE "PROFILE" CIRCLE FROM 0,0,0 IN GLOBAL COORDINATES TO THE LOCAL COORDINATES OF PLANE ON CURVE

Solid By Sweep
profile ➔ Solid
path ➔ Solid

REVIT DYNAMO GEOMETRY

REVIT MASS CURVE GEOMETRY

DYNAMO GEOMETRY

NOTES
CREATES A SOLID BY SWEEPING A PROFILE ALONG A CURVE. ANY CURVE WILL WORK AND ANY CLOSED PROFILE COULD BE USED. CIRCLE WAS USED IN THIS CASE
SCALE BY DESIGN SCRIPT

INSERT ANY DYNAMOBIM GEOMETRY YOU WANT (IN THIS CASE AN ARCH COLUMN)

THIS IS A VARIABLE (CALL IT WHATEVER YOU WANT)

"Scale" IS DESIGN SCRIPT TO TELL DYNAMO TO SCALE GEOMETRY

NOTE: WITH THIS DESIGNSCRIPT GEOMETRY SCALES ON GEOMETRY ORIGIN

NOTE: THE METHOD SHOWS HOW TO SCALE ANY DYNAMO GEOMETRY USING 1 SIMPLE LINE OF DESIGN SCRIPT CODE. REVIT IS NOT INVOLVED WITH THIS EXAMPLE
MIRROR DYNAMO GEOMETRY BY DESIGN SCRIPT

Code Block

cube1 = Cuboid.ByLengths(Point.ByCoordinates(0, 5, 5), 1, 1, 1);
mirroredCube = cube1.Mirror(Plane.XZ());

TO ONLY CREATE MIRRORED GEOMETRY SIMPLY MOVE THE MIRROR COMMAND TO THE END OF THE CREATED OBJECT WITH A "." AS SHOWN HERE.
WHY? BECAUSE ONLY ONE OBJECT IS CREATED (THE MIRRORED OBJECT)

Code Block

ube = Cuboid.ByLengths(Point.ByCoordinates(0, 5, 0), 1, 1, 1).Mirror(Plane.XZ());

Cuboid(Length = 1.000, Width = 1.000, Height = 1.000)

NOTE: THIS METHOD COULD BE USED FOR OTHER GEOMETRY MODIFICATIONS SUCH AS SCALE, EXPLODE, ETC. IT COULD ALSO BE USED ON OTHER GEOMETRY SUCH AS CUBIODS. TO DISCOVER WHAT OTHER MODIFICATIONS COULD BE MADE TO THE DYNAMO GEOMETRY SIMPLY ADD A "." AFTER THE GEOMETRY

ORIGINAL (NOT CREATED)

ONLY MIRRORED WAS CREATED
HOW DYNAMO SELECTS REVIT ELEMENTS

Create separate lines
Model and Select them from Revit or
Use Design Script is shown here

0

DYNAMO NODES

1

Place this node to
JOIN the lines into a
"polycurve"

2

DYNAMO GEOMETRY

3

Add Radius
This node defines
fillet as inside

4

This node creates a new
polycurve (or joined curve)
with the fillets

NOTES
NOTE: THE NODES SHOWN HAVE BEEN RENAMED FROM THE ORIGINAL NAMES OF THE OOTB SELECTION NODES TO HELP CLARIFY WHICH NODES SELECT WHICH REVIT FAMILIES OR TYPES OR CATEGORIES
CREATE LIST OR LISTS USING DESIGN SCRIPT

\{item1, item2, item3, etc...\}

items separated by commas (,)
and closed with curly braces (\{\ldots\}) as shown

\{List 1, List 2\}

Nested List? Simply add more Curly Braces

NOTE: THE METHOD SHOWS HOW TO CREATE A LIST OR NESTED LIST USING DESIGN SCRIPT AND "{\ldots}" CURLY BRACES
GET ITEM FROM LIST BY DESIGN SCRIPT

NEW METHOD

4 NODE METHOD

DYNAMO NODES

1 NODE METHOD

DYNAMO GEOMETRY

HOW DO YOU GET THIS ITEM?

5 ITEMS IN LIST

ITEM 5 IS ALSO A LIST WITH 3 ITEMS (NESTED LIST)

NOTE: THE METHOD SHOWS HOW GET A SINGLE ITEM FROM A NESTED LIST
GET ITEMS FROM LIST BY DESIGN SCRIPT

HOW DO YOU GET ITEM 1 THAT IS IN A LIST WITHIN A LIST?

Using GetItemAtIndex takes 6 Nodes

Using Design Script takes 1 Node!

Just list the indexes as shown..... Yes thats it!

NOTE:
Using Design Script to get an item from a single list is just as easy and it takes 1 Node!

Want item 10? Type this
Yes thats it!

NOTE: THE METHOD SHOWS HOW GET MULTIPLE ITEMS FROM A NESTED LIST
CREATE RANGED EXPRESSION
(LISTS OF NUMBERS IN A SEQUENCE)

\[ A..B..\alpha C \]

(3 entries separated by two periods "..")

A  START of the Number RANGE
B  END of the Number RANGE
C  Step or Amount depending on \( \alpha \)

\( \alpha \)

Blank = Step (Not to exceed end range value)

\#  = Amount (equal steps from start to end result)

\~  = Step (Estimate such that end range value results)

---

NOTES:
- In the first list [7] was not generated B/C it exceeds "B" with is the max value
- Common format is to use "A..B (NO 'C'), if C is blank then it defaults to 1
- Ex: A..B..1
- Also, not all ranged expressions are shown
(NO LIST AT LEVEL USED) OPERATES ON THE "HIGHEST LIST LEVEL (@L3 IN THIS CASE)"

NOTE: THE METHOD SHOWS THE BASIC UNDERSTANDING OF LIST AT LEVEL
REORDER LISTS OF POINTS BY ORDINATE

SELECTS ALL THE REVIT POINTS IN NO PARTICULAR ORDER

CONVERTS REVIT POINTS TO DYNAMO POINTS AND FLATTENS LIST

SETS THE X COORDINATE OF POINTS AS THE SORT FUNCTION

SORTS OR REORGANIZES THE LIST BY "FUNCTION" IN THIS CASE "X" COORDINATE

ORIGINAL ORDER OF POINTS
NOTICE THE POINTS ARE "OUT OF ORDER"

NEW ORDER OF POINTS
NOTICE THE POINTS ARE "IN ORDER" IN THE X DIRECTION

NOTE
FUNCTIONS ARE CREATED WHEN NOT ALL THE INPUT PORTS CONTAIN INPUT WIRES SUCH AS THE POINT.X NODE IN THIS CASE. THE POINT.X IS USED AS A "CRITERIA" TO REORDER THE LIST. LIST BY FUNCTION ONLY WORKS IF THE FUNCTION IS RELATED TO THE LIST.
SIMPLE IF STATEMENTS

1. TWO "NODES" TO BE COMPARED
   - Code Block: `1; >`
   - Code Block: `2; >`

   **NOTE:** THERE ARE OTHER "OPERATORS" AS SHOWN HERE THAT COULD BE USED TO COMPARE

2. OPERATOR NODE
   - Operator: `==`
   - `x var[][[]]` and `y`

   **IN THIS CASE "==" MEANS X AND Y MUST BE EQUAL**

3. TWO NODES FOR IF TRUE OR IF FALSE
   - Code Block: "IF TRUE RUN THIS NODE"
   - Code Block: "IF FALSE RUN THIS NODE"

   **NOTE:** THESE COULD BE ANY NODE (NOT JUST TEXT)

4. IF STATEMENT NODE
   - If:
     - test
     - result
     - true
     - false

   (THIS NODE DISPLAYS THE RESULT IF TRUE OR IF FALSE)

**NOTE:**
THIS CREATES IF STATEMENT LOGIC. IF STATEMENTS COULD RUN ANY DYNAMO NODE AND ALSO COULD BE "NESTED" FOR MORE COMPLICATED LOGIC
# Math Variables and Equations

**Diagram:**

- Code Block 1: `1; >` connected to Code Block 2: `2; >` resulting in `3`.

**Text:**

- **This is called the "operator" see chart below for common ones.**
- **Any non number "string" that is not defined will be placed as an input port on the code block.**

## Common Math Variables

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs (number)</td>
<td>Absolute value of a number.</td>
</tr>
<tr>
<td>Abs (integer)</td>
<td>Absolute value of an integer.</td>
</tr>
<tr>
<td>Acof</td>
<td>Cosine of an angle.</td>
</tr>
<tr>
<td>Asin</td>
<td>Sine of an angle.</td>
</tr>
<tr>
<td>Atan</td>
<td>Tangent of an angle.</td>
</tr>
<tr>
<td>Atan2</td>
<td>Arctangent of a number.</td>
</tr>
<tr>
<td>Average</td>
<td>Average of numbers.</td>
</tr>
<tr>
<td>Ceiling</td>
<td>Ceiling of a number.</td>
</tr>
<tr>
<td>Cos</td>
<td>Cosine of an angle.</td>
</tr>
<tr>
<td>Cosh</td>
<td>Hyperbolic cosine of a number.</td>
</tr>
<tr>
<td>DegreesToRadians</td>
<td>Degrees to radians.</td>
</tr>
<tr>
<td>DivRem</td>
<td>Division and remainder.</td>
</tr>
<tr>
<td>Exp</td>
<td>Exponential function.</td>
</tr>
<tr>
<td>Factorial</td>
<td>Factorial of a number.</td>
</tr>
<tr>
<td>Floor</td>
<td>Floor of a number.</td>
</tr>
<tr>
<td>Log (number)</td>
<td>Natural logarithm of a number.</td>
</tr>
<tr>
<td>Log (number, logBase)</td>
<td>Logarithm of a number.</td>
</tr>
<tr>
<td>Log10</td>
<td>Common logarithm (base 10).</td>
</tr>
<tr>
<td>Max (value1, value2)</td>
<td>Maximum of two values.</td>
</tr>
<tr>
<td>Min (value1, value2)</td>
<td>Minimum of two values.</td>
</tr>
<tr>
<td>Pow</td>
<td>Power of a number.</td>
</tr>
<tr>
<td>RadiansToDegrees</td>
<td>Radians to degrees.</td>
</tr>
<tr>
<td>Random</td>
<td>Random number.</td>
</tr>
<tr>
<td>RandomList</td>
<td>Generate a list of random numbers.</td>
</tr>
<tr>
<td>RemapRange</td>
<td>Remap range.</td>
</tr>
<tr>
<td>Round (number, digits)</td>
<td>Round a number to a specified number of decimal places.</td>
</tr>
<tr>
<td>Square</td>
<td>Square of a number.</td>
</tr>
<tr>
<td>Sqrt</td>
<td>Square root of a number.</td>
</tr>
<tr>
<td>Sum</td>
<td>Sum of a list of numbers.</td>
</tr>
<tr>
<td>Tan</td>
<td>Tangent of an angle.</td>
</tr>
<tr>
<td>Tanh</td>
<td>Hyperbolic tangent of a number.</td>
</tr>
<tr>
<td>E</td>
<td>The mathematical constant e.</td>
</tr>
<tr>
<td>GoldenRatio</td>
<td>Golden ratio.</td>
</tr>
<tr>
<td>PI</td>
<td>The mathematical constant π.</td>
</tr>
<tr>
<td>PiTimes2</td>
<td>Pi times 2.</td>
</tr>
</tbody>
</table>

**Note:**

This creates an equation using the code block. Any variable that is not defined is placed in the input port and most operators could be used as well as the functions shown.
POINTS COORDINATES AT VERTICES TO EXCEL

CUBIOD

- Creates a cuboid at low pt (0,0,0) high pt (1,1,1) and gets pts at vertices.

X, Y, Z

- Gets x, y, z from points and combines into 1 list that contains 3 lists and transposes that list.

File Path

- Writes the coordinate lists to excel.

Note: What is shown in excel in blue was existing.

DYNAMO CUBIOD

DYNAMO GEOMETRY + EXCEL

NOTE

The excel spreadsheet was existing and what is shown in blue was what was preset in excel and what is shown in orange is what was written in by dynamo.
READ RAW DATA FROM EXCEL

**NOTE**
"FILE FROMPATH" NODE MUST BE USED WHAT READING DATA FROM EXCEL AND DATA FROM EXCEL IS READ AS ROW THEN COLUMNS HENCE THE USE OF THE TRANSFORM NODE.
CREATE DYNAMO PTS FROM RAW EXCEL DATA

SELECTS THE EXISTING EXCEL FILE VIA FILE PATH

READS THE RAW DATA FROM EXISTING EXCEL SPREADSHEET AND TRANPOSES THE RAW DATA LISTS FROM EXCEL

EXTRACTS EACH LIST FOR X,Y,Z AND CREATES DYNAMO POINTS

NOTE
"FILE FROMPATH" NODE MUST BE USED WHAT READING DATA FROM EXCEL AND DATA FROM EXCEL IS READ AS ROW THEN COLUMNS HENCE THE USE OF THE TRANSFORM NODE

RAW EXCEL SPREADSHEET DATA

DYNAMO NODES

EXCEL

DYNAMO GEOMETRY

NOTE
COPY TYPES FROM UNOPENED REVIT PROJECT TO ACTIVE REVIT PROJECT USING ZT DYNAMO NODES

These nodes designate the location of the unopened Revit project file and select the category.

This node shows all family types by category from unopened Revit project.

This node copies the selected type(s) to the active Revit project.

Step 1: Open the Revit sample file “rst_advanced_sample_project.rvt”
Step 2: Start and new Dynamo file and add node shown.
Step 3: Select the sample file “ rst_advanced_sample_project.rvt” from the “Path” node
Note: Custom nodes could be found in the “Simplex” Dynamo package.
READ TEXT FROM MS WORD DOCUMENTS (.DOCX)

**SIMPLEX PACKAGE**

**DYNAMO NODES**

**MS WORD**

**Note:** This workflow was intended for non-complex formatted word documents.

**STEPS & NOTES**

STEP 1: OPEN THE REVIT SAMPLE FILE “rst_advanced_sample_project.rvt”

STEP 2: START AND NEW DYNAMO FILE AND ADD NODE SHOWN

STEP 3: SELECT THE SAMPLE FILE “READ_TEXT_FROM_WORD.DOCX” FROM THE “PATH” NODE

NOTE: CUSTOM NODES COULD BE FOUND IN THE “SIMPLEX” DYNAMO PACKAGE
READ TEXT FROM MS WORD DOCUMENTS (.DOCX)

STEP 1: OPEN THE REVIT SAMPLE FILE “rst_advanced_sample_project.rvt”
STEP 2: START A NEW DYNAMO FILE AND ADD THE NODE SHOWN AND RUN

NOTE: CUSTOM NODES COULD BE FOUND IN THE "SIMPLEX" DYNAMO PACKAGE