



AR20475

## Revit and Dynamo for Landscape Architects

William Carney  
BSA LifeStructures

### Learning Objectives

- Learn how to use Revit for landscape architecture
- Learn how to utilize Dynamo to improve your typical Revit workflow
- Understand how a BIM database can streamline landscape-architecture design documentation
- Become comfortable using Revit

### Class Description

This class will demonstrate how you can use Revit software for landscape architecture with some assistance from Dynamo extension. Attendees will get an overview of how to use Revit software features such as families to provide typical symbols in plan while providing opportunities for visualization, scheduling to provide quantities, and tagging to annotate consistent information. Going beyond the basics, we'll use Dynamo extension to show advanced techniques to streamline the design process. This session features Revit and Dynamo Studio. AIA Approved

### Your AU Expert

William Carney is the BIM Director at BSA LifeStructures where he is a firm leader in adoption and implementation of design technology. Although he is not a Landscape Architect, as an early adopter of Revit, he has had the opportunity of leveraging BIM on a wide variety of project types and sizes using Revit site features for design, documentation, and visualization. William highly involved in the St. Louis A&E design community as committee member for the St. Louis Revit User Group and part of the BSE advisory committee with Ranken Technical College where he draws on his 10 years of architectural experience to help improve the St. Louis area where he makes his home with his wife Liz, son Hank, and Greyhound Helen.



## **Table of Contents**

<b>Learning Objectives</b> .....	<b>Page 1</b>
<b>Class Description</b> .....	<b>Page 1</b>
<b>About the Presenter</b> .....	<b>Page 1</b>
<b>Introduction to the Course</b> .....	<b>Page 4</b>
<b>How Can Revit Benefit a designer?</b> .....	<b>Page 4</b>
The Database	
Families	
Visibility Graphics	
<b>How Can Dynamo Benefit A Revit User?</b> .....	<b>Page 4</b>
Use Dynamo to do what Revit cannot	
Connect to external information	
Perform tedious tasks	
Perform actions based on calculations	
Place and modify geometry	
Annotating with Dynamo	
Drawing lines, counting trees, updating parameters, and placing tags:	
<b>Creating Topography</b> .....	<b>Page 16</b>
Topography from CAD import	
Topography from floors	
Create topography using dynamo	
<b>Modeling Curbs</b> .....	<b>Page 17</b>
Slab edges	
Working with slab edges	
Curb cuts	
Method one: Edited slab with walls	
Method Two: Void family to cut floor and curb	
<b>Working With Materials</b> .....	<b>Page 25</b>
Repeating patterns	
But what about curves?	
Using the join tool	
<b>Hardscape Strategies</b> .....	<b>Page 30</b>
My thoughts	
The differences	
<b>Hardscape with Dynamo</b> .....	<b>Page 31</b>
Making floors from sub-regions	
Make a floor follow Topography	
<b>Using Railings With Sloped Floors</b> .....	<b>Page 34</b>
Hosting the railing	
Curbs	
Balusters	



<b>Designing with Dynamo</b> .....	<b>Page 37</b>
Select a line and place plants along it	
Troubleshooting	
Splines	
Planting patterns	
Creating rows	
Annotating	
Working with Revit geometry	
Selecting edges	
Collect Sketches	
Select Faces	
<b>Planter Beds (Combining it all together)</b> .....	<b>Page 52</b>
Adding slope	
Edging	
Placing Plants	
Evenly spaced plants in a region	
Combining lines and regions	
<b>Conclusion</b> .....	<b>Page 58</b>



## Introduction to the Course

Shortly after starting at BSA LifeStructures, I was approached by our landscape architects asking about Revit. I had a meeting with the design team to discuss what we do as Landscape architects and evaluate if Revit would make sense for their process. As a group, we identified a lot of re-work through a CAD workflow, our designers layout and draft their landscape designs in CAD. Our designers were providing design documents and presentation packages. To do this they needed to:

- Color the site plan in Photoshop
- Redraw the site in Sketchup to provide perspectives of the site
- Count plants to provide a cost estimate
- Update documents as the design changed

The conversations turned toward Revit and how it could benefit the landscape architecture workflow for the design and documentation process.

## How can Revit benefit a designer?

By utilizing Revit on our landscape architecture projects, designers are able to produce construction documents with a live update to presentation graphics, plant counts, and cost. This is all possible because Revit is producing a building information model which is a 3D representation of a database. As something is drawn in Revit, that item and all of its associated data is stored in a database that is easily accessible.

A database of information backing a 3dimensional design environment is what makes Revit powerful.

## The database

A BIM model database is made up of type and instance tables for every category inside of the Revit model. The type table stores information about each family type of the category of the table while the instance table stores information about every individual instance of the type that is placed in the model for that category.

Confused?... Here's an example. An oak tree is a type of tree and is of the planting category inside of Revit. There is a table of information called Planting Types that contains data about each type of tree. For table can contain information like species, watering needs, root sizes, or even fall colors. Each oak tree that is placed in your project will be added to the instance table of the planting category. This table stores information like its location, when it's created, and if it demolished.

Because each individual plant that you place is tracked in the database, and each item knows what type of item it is, you have quantities at your fingertips. Revit allows you to combine related information. If you store the cost of each type of plant, you can add the cost for each of those plants in your project to give you the total cost of the plants needed. Using the information of the database allows for more correct documents and reduces error from manually counting. Having objects connected to information stored in the database, provides an opportunity to create more informative diagrams, details, and drawings confidently, because what is displayed is correct and updates if the information is updated in the database.

## Families

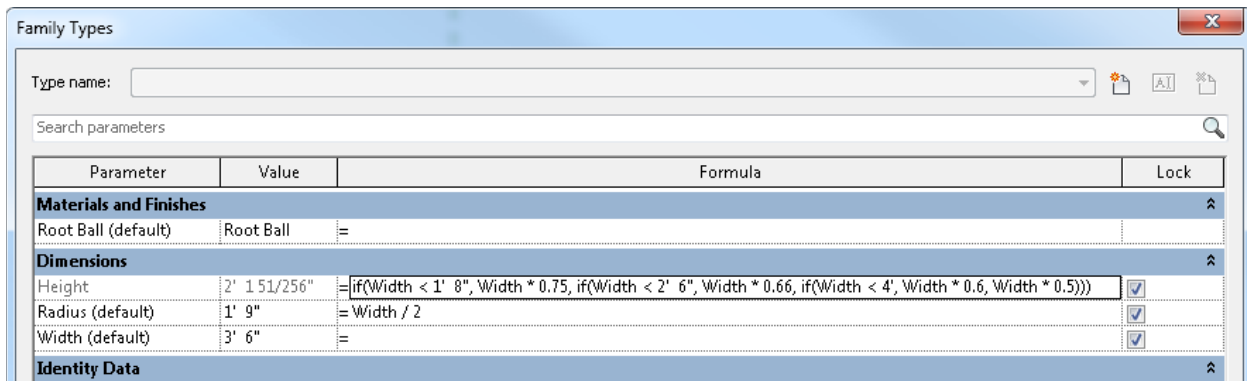
Families and categories are two major concepts to understand when working in Revit. A family is a group of elements with a common set of properties called parameters. A category is the overall grouping of families inside of Revit, defining where information is stored in the BIM database. Planting is a category in Revit a shrub would be a family of the planting category. Sagebrush, berberis, or boxwoods would be types of shrubs. Each family gives us the opportunity to define the geometric appearance of all of the types for the different view types in Revit.

Similar to AutoCAD blocks, families allow a Revit user to update the geometry inside the family in one location and have those changes applied to every instance of that family throughout the entire Revit project. Family parameters make families even more powerful. The geometry within a family can be associated to the family's parameters making it reactive to input. Dimensions and materials are the most commonly used parameters associated with family geometry. Where the parameters can take this concept even further is in the use of formulas. Formulas can be added to parameters to drive the value in the given parameter. Formulas can read the values of other parameters and use conditional statements similar to excel.

We can do simple and complex formulas. If we know a family's width is half the height of the family. The formula is height/2. What about ranges of sizes? Formulas can use if statements where you can write: if(Condition, What happens if that condition is true, what happens if that condition is false) Something more practical to look at would be the chart below. We would use a formula that says if the diameter is less than 20", diameter x .75%.

Root ball diameter (inches)	Minimum root ball depth
less than 20	not less than 75% of diameter
20-30	not less than 66% of diameter
31-48	not less than 60% of diameter

*ROOT BALL DEPTH CHART*



## Visibility Graphics

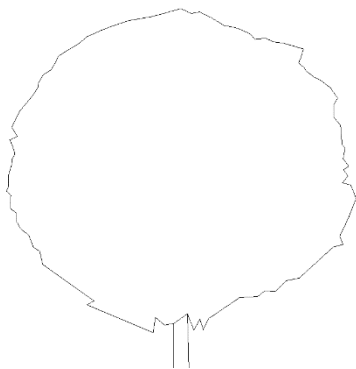
I cringe every time I hear someone say *“I hate Revit line weights.”*

In Revit, you have the ability to control the line weight, line type, and hatch pattern of each category and subcategory of object being cut by the view plane or not cut throughout every view in the project through object styles and control those same values at an individual view level using visibility graphics. View filters allow a Revit user to further control those same values by creating rules based on parameter information. As an example, you can create a view filter that any plant with a diameter less than 6” be halftone. If all of that does not suit your needs, Revit users can control the same values on an individual element level by selecting a family in a view and overriding it and if that isn’t enough, the line weight override tool will override the line weight, color, and pattern of an individual line in a view.

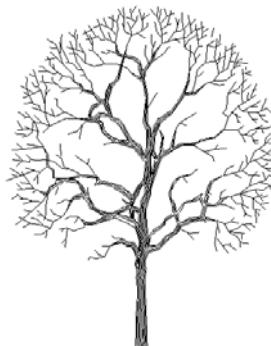
At a family level, Revit stores definitions for an element’s appearance in plan, section, elevation, and 3D. For each of these view types, Revit allows the user to define an item’s appearance at course, medium, and fine detail levels. This provides an opportunity to store pre-set appearances for the same family in three different ways for four different types of views. This ability to display information allows designers to show the same item in many different ways.

What that means is you can use a symbol for your tree in documentation, a realistic view for detailing, and a graphically pleasing image for presentations all from the same family.

### Elevation View



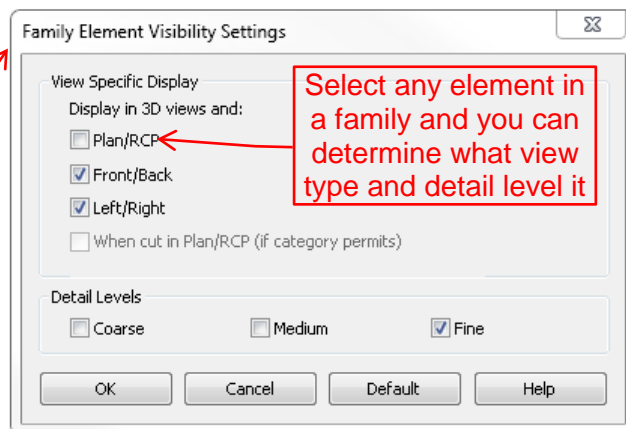
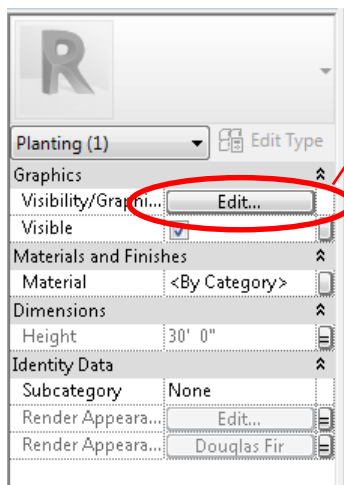
COURSE



MEDIUM



FINE



Here is a link to a great posting on Landarchbim.com covering making Revit families.

<https://landarchbim.com/2015/07/29/complex-planting-graphics-simplified/>

## How Can Dynamo Benefit A Revit User?

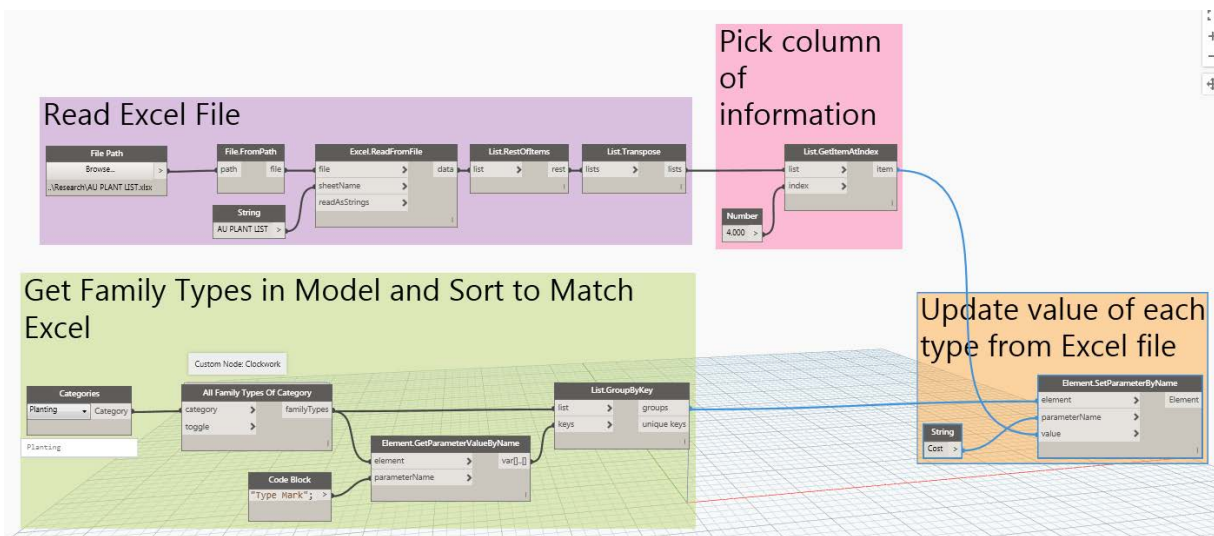
Dynamo extends the capabilities of Revit. The Dynamo extension has the ability to read, write, and process data from outside or inside of a Revit project. The great thing about Dynamo is you do not have to be a Dynamo user to use Dynamo in your workflow. I am aware that is a confusing statement but hear me out. Dynamo workspaces can be preset routines that a Revit user can run similar to lisp routines in AutoCAD. With the Dynamo Playlist, this is even more of a true statement where a Dynamo routine can now be run without even entering the Dynamo interface.

### Use Dynamo to do what Revit cannot:

Dynamo has core functions for sorting lists of information and the ability to update Revit. A Dynamo user can create a workspace that can read information in a project file and tell Revit to do something with that information. The reason that sounds vague is that by using Dynamo, a person is able to perform an extremely wide array of tasks. The main concept to grasp is have Dynamo create a list and perform a task to that list.

### Connect to external information:

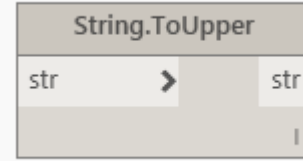
A very simple example of how Dynamo can help extend the capabilities of Revit is to update and coordinate Revit families from an Excel file. Many Landscape Architects have a list of plants they use with information about planting spacing, sizes, cost, and much more. Dynamo can read excel files and all the families in your project. If someone uses an excel file to keep information like plant cost updated, Dynamo can be used to update the families in a Revit file with the cost data in Excel.



DYNAMO WORKSPACE TO READ EXCEL FILE AND UPDATE FAMILIES

### Perform tedious tasks:

Dynamo is perfect for tedious tasks. As mentioned before you create a list of stuff and tell dynamo to perform a task to that list. It doesn't get tired or forget a step while running the same routines over and over on whatever list of items you give it.



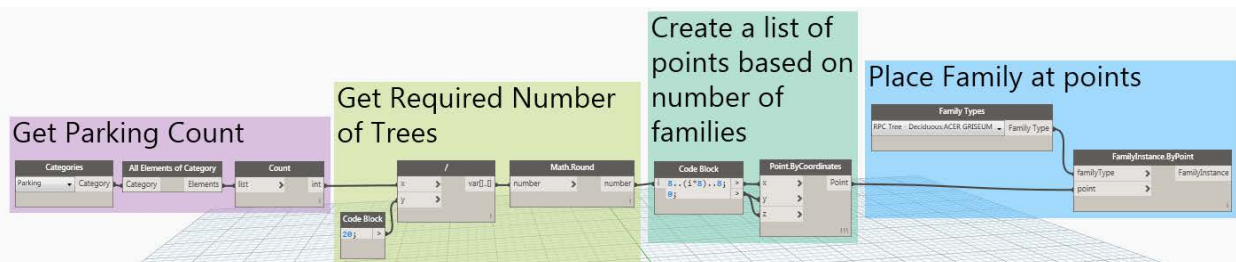
If you've never opened Revit or Dynamo, you may still appreciate this very simple example of how Dynamo can benefit a designer. Dynamo has a node called String.ToUpper that will convert any text that is passed through the node to UPPERCASE. Checking to make sure text is uppercase in a set of drawings is a very tedious thing to review and even more so to update when it is wrong. By using dynamo to make a list of all your sheet names, view names, or even all of the text notes in the drawings, you can run that information through the String.ToUpper node to capitalize any text. No checking required.

### Perform actions based on calculations:

Dynamo has math functions that can perform calculations based on the information that is in the Revit model. This can extend Revit's capabilities in many ways. Often there is a requirement for amount of greenspace and trees based on the parking count. If the parking spaces are modeled in Revit, schedules can be used to generate the numbers shown below but Dynamo can do something with that information.

INTERIOR PARKING LOT: 15 SQ FEET GREEN SPACE PER PARKING STALL X 81 SPACES	
REQ. SQ FT.	1,215
ACTUAL SQ FT	3,598
1 TREE PER EVERY 20 SPOTS	
REQ. TREES	5
ACTUAL TREES	12

In Dynamo, we can use the All Elements of Category node to get all the parking spaces. Using the Count node, Dynamo can report the number of parking spaces. The Divide node, can be used to divide the number of spaces by the required spacing. In this case 20. The Math.Round node will round the partial number to an integer. This integer is the number of trees we want to place. Dynamo can then create a list of points as locations to place the selected tree type.



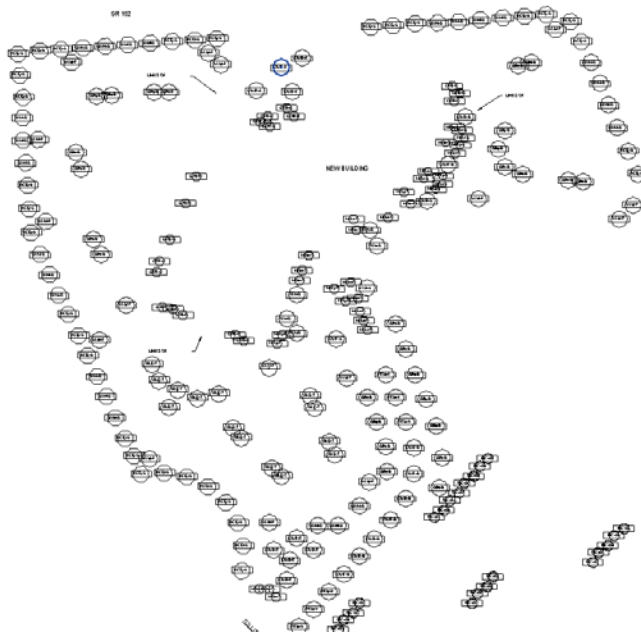


### Place and modify geometry:

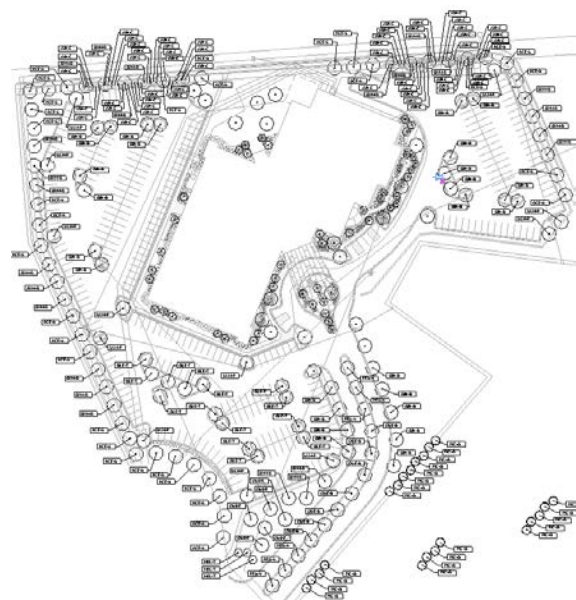
As shown in the section above, Dynamo can place geometry. When it was first introduced Dynamo seemed more like a complex modeling program similar to Grasshopper. However, it is the previously discussed features that have led Dynamo to rise in popularity. The ability to place geometry with Dynamo is one of its more powerful features and a feature that I am going to focus heavily on for Landscape Architecture because of the non-linear nature of many of the designs. This document will heavily cover how Dynamo can place and modify geometry.

### Annotating with Dynamo

The very first thing that I identified a need for Dynamo was annotating trees. Our designers easily placed trees and created schedules for cost estimates using Revit. When they used the tag all button, a tag per tree was placed which created a lot of tags to manipulate and left the drawings a bit cluttered. We did save time by not coordinating the type of tree but it's not an easy sell for me to say why someone would want to use Revit over CAD for Landscape Architecture without the help of Dynamo.



*SITE AFTER TAG ALL*



*ARRANGE EACH TAGS*

Often, landscape drawings depict where a type of tree is planted by drawing a chain of lines through the similar tree types then indicate the number of trees in the chain of lines through a number shown with a tag (shown below). In Revit, this is possible but it is a very manual task with a lot of drafting and counting. Doing this in Revit is marginally better than doing the work in CAD and that is up for debate.

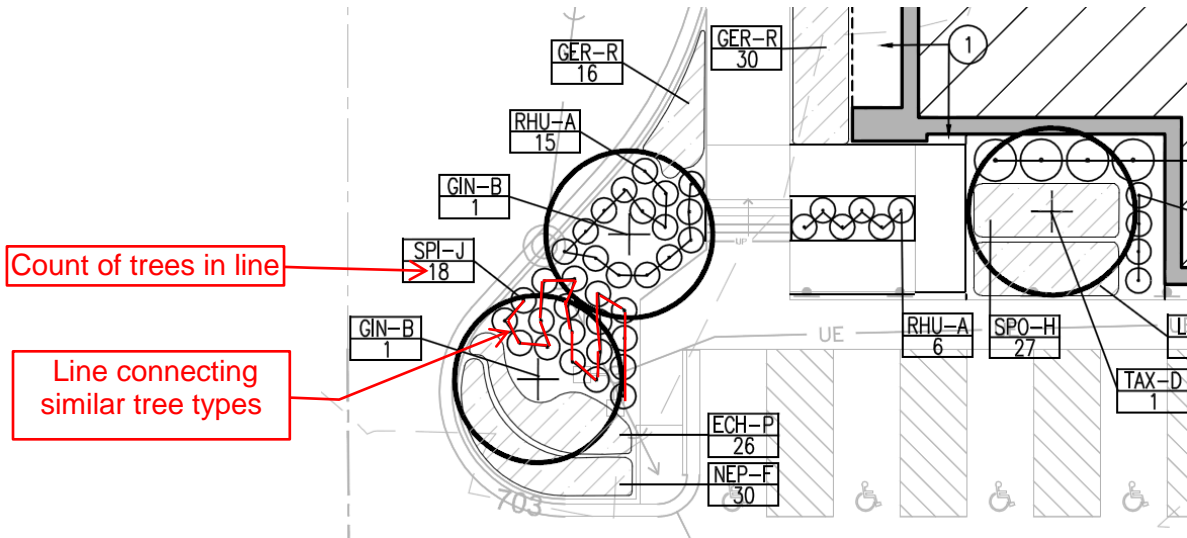
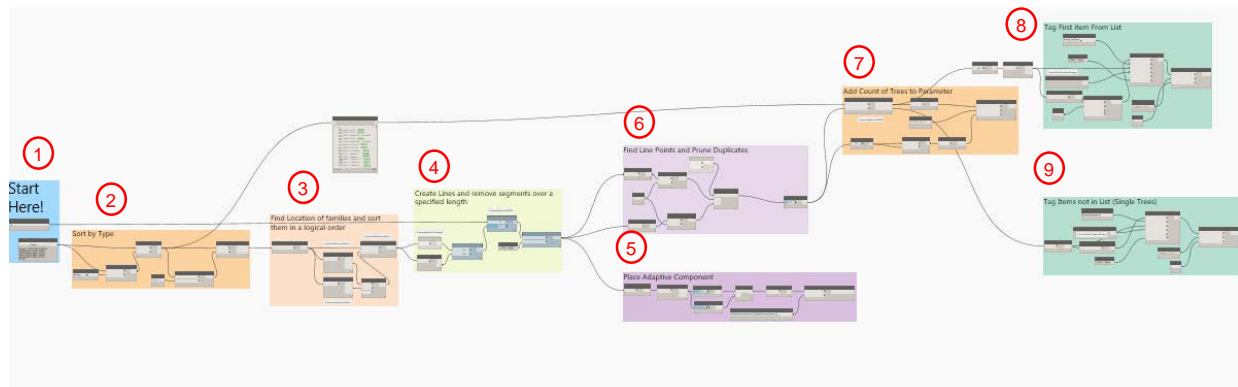


IMAGE OF CAD SITE ANNOTATION

### Drawing lines, counting trees, updating parameters, and placing tags:

Here's a look at a Dynamo Workspace where the Revit user can select everything in the view, filter out what is of the planting category, sort them by plant types, determine if they are located close enough to be considered a string of plants, draw a line between each of those plants, count how many were in the line, and finally, tag the plants.

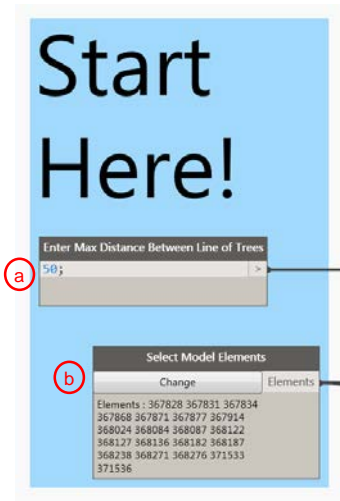


The image above is the overall workspace, we're going to walk through each step and then show the results. I have tried to do this for most workspaces. The numbers in Red # in the image are called out as sections below. Occasionally on an output or selection if I am going to Reference it I later in the section, I have added a circle with a letter. **a**

For this workspace, I used a line based adaptive component as the line for the plants because it's visible in 3D which I think can lead to some interesting 3D diagrams. I wanted to show multiple methods annotating so later I will show a method using a detail line.

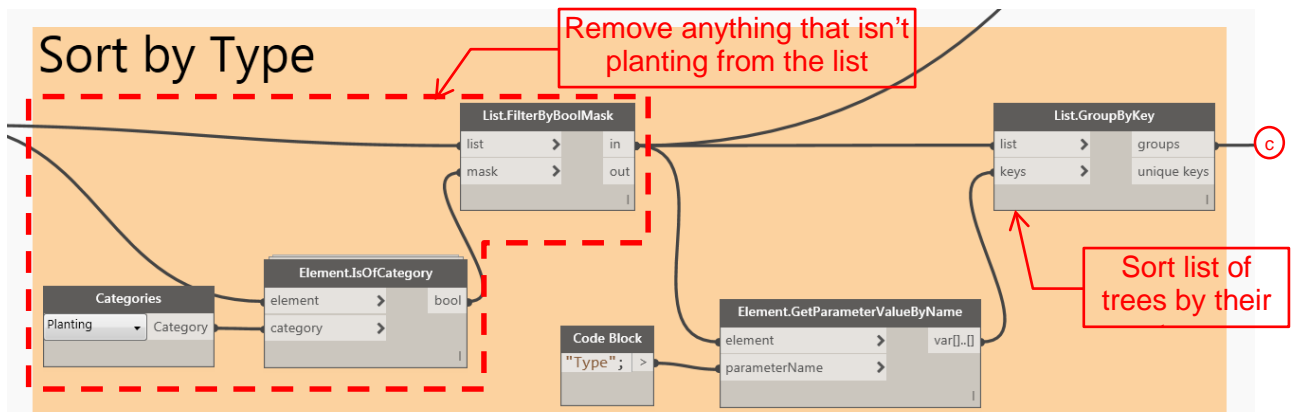
## 1) Start Here

I like to lump all of the nodes that require user input together. (I picked this tip up this summer at RTC thank you Chris Williams from AECOM). There are two nodes in this group. The first is the number in feet of maximum spacing and the second is a selection node to select the elements in the view. We could have made this workspace using the select all elements visible in view node however, I like the selection node because it allows someone to select portion of the view to process less information. This helps with speed of processing and allows for updates after the initial run. Also, if you modify an area, this will allow you to more easily update the drawing. The user can delete the tags and lines then reselect that area and update.

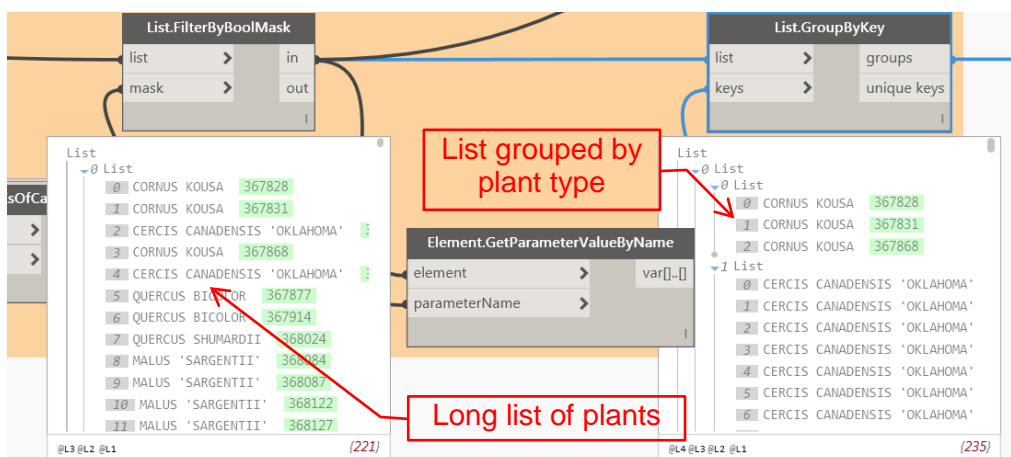


## 1) Sort by Type

This group will take all the elements from the select model Elements node (b shown above) and filter them by the selected category. This allows the user to select everything in the view and work with only the planting category families.



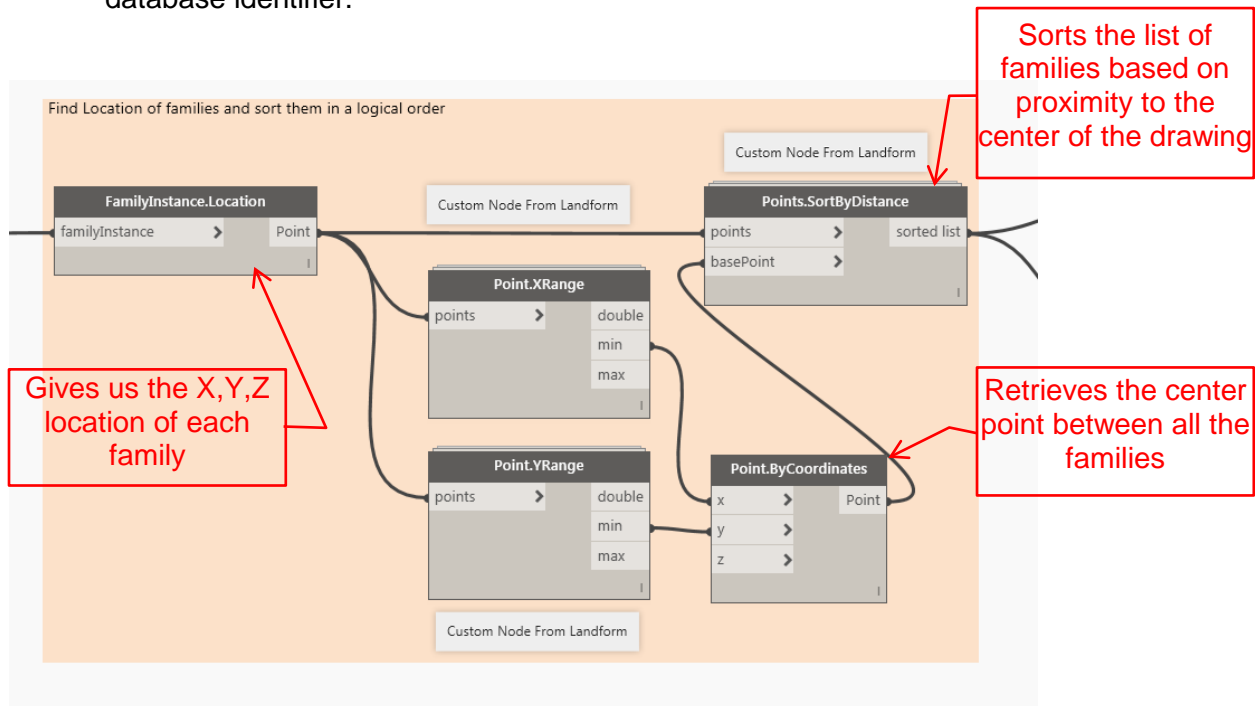
## Information expanded



## 2) Find Family Location and sort them in a logical order.

Here's some unnecessary back info, Dynamo reads the Revit database when it creates the lists of information. The way that a BIM database works is that it has a list of information (in this case plants) and whenever an item is added to the Revit model, that item is added to the list of information where it receives an element id which is a number that makes that a unique item in the database. These lists build as items are added.

The reason this is important is this group is getting the location point of each family. We're going to use that location to specify the start and end point of each line that dynamo will draw. If every plant is drawn in the exact order that you would like to draw the lines this would be fine. However, most likely the plants are not placed in the exact order that you would like to annotate them. We need to sort them by their location not database identifier.

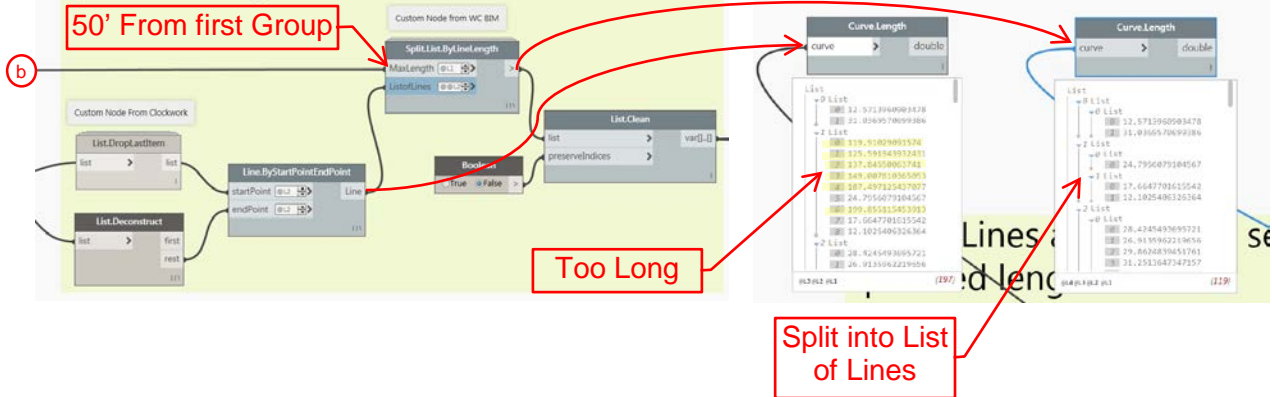


## 3) Create Lines and Remove Segments that are longer than a specified Length

This group will take the sorted list of points and make a line connecting all of the families. To do this the start point end point node is used. What I've done is taken the list of points and made two lists one list we drop the last item so we have all the start points of lines, the second list we drop the first item so we have all the end points.

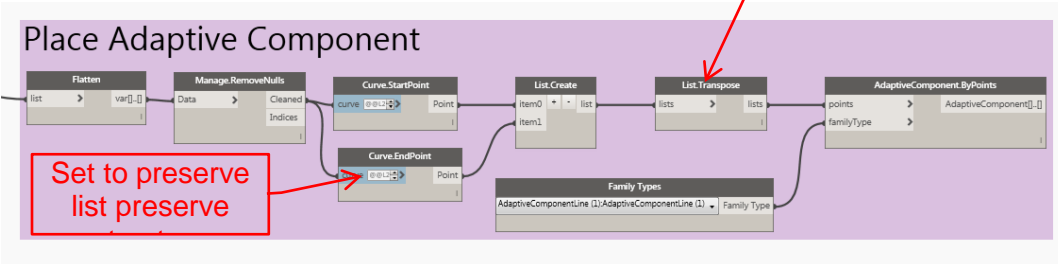
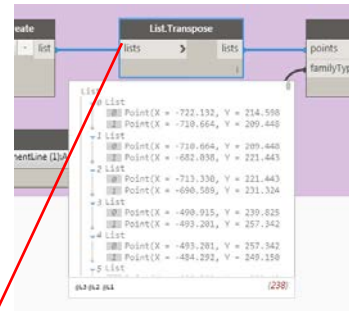
This list of lines is input to **Slit.List.ByLineLength** which is a custom node that I made (Uploaded with the datasets). It will take the input of a number and a list of lines. The number is the maximum distance of a line. For this example, we took 50'. This produces the list of lines that are connected before a line exceeds a length of 50'. When a line exceeds that distance, a new list is started with the remaining connected lines. This will produce lists of lines to connect similar plant types from our selection without having lines extend across the entire drawing.

### Create Lines and remove segments over a specified length



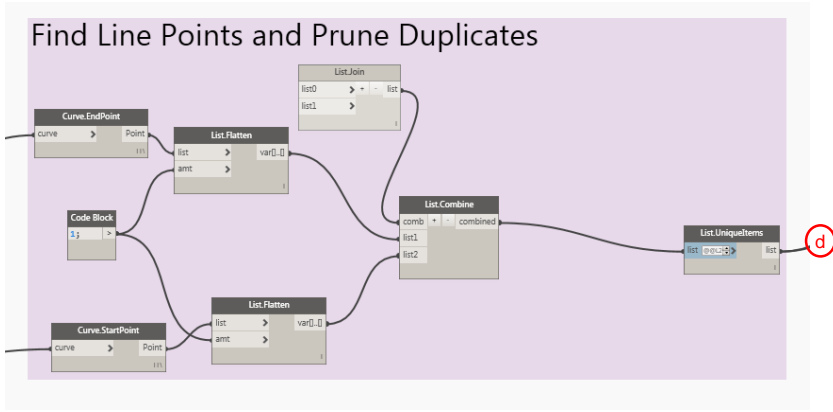
#### 4) Place the adaptive components

This group will take the list of lines that have been pruned of lines longer than 50' and place a line based adaptive component at that location. Most of this group is processing the line information to format it in a matter that can be used by the adaptive component by points node. The adaptive component by points node takes requires an input of start point and end point for each line as shown right.



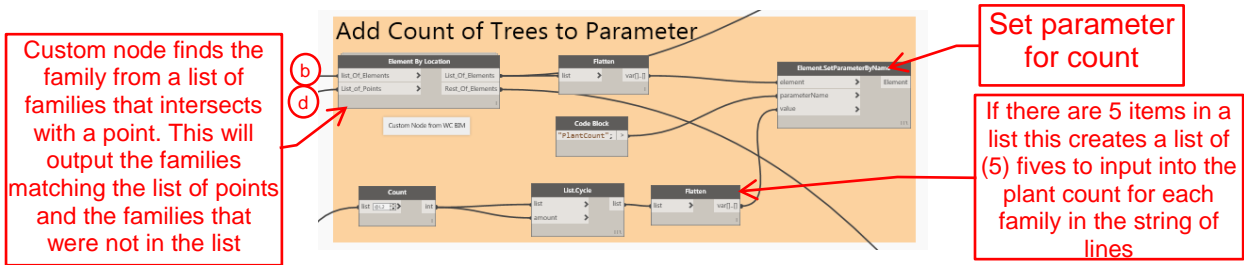
#### 5) Find Points of the lines and remove duplicates from overlapping start and end points.

These next two sections find the number of trees in each string and input that count into a shared parameter so we can tag and display the count of trees. This section takes the list of lines sorted and pruned of lines over 50' from group 4. We'll take the start and end point of each of those lines and remove duplicates. This gives us the location of families.



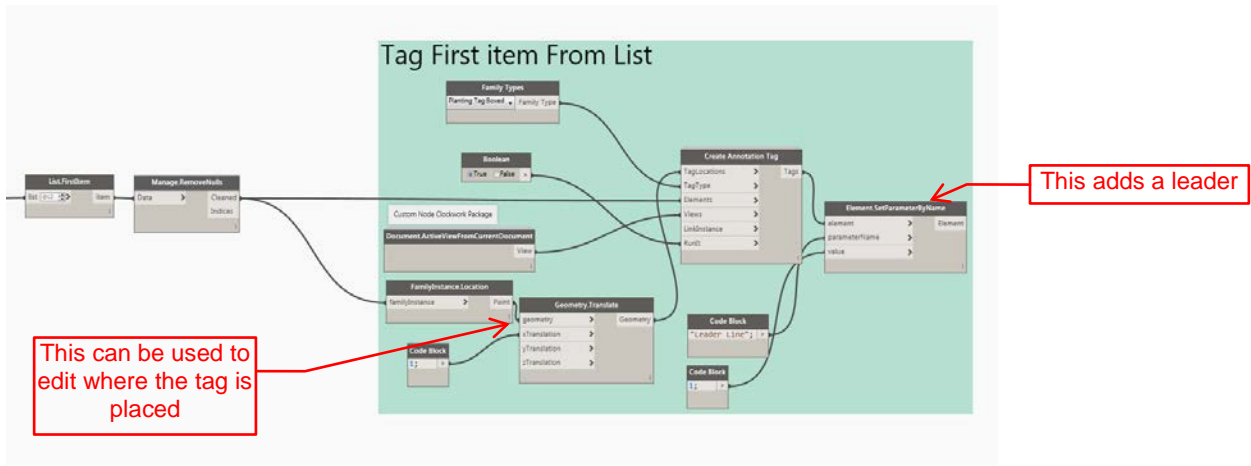
**6) Create a list of families from a list of points. Then count the number items in the list and add that to a Shared Parameter to tag**

This group uses another custom node that I made (Element by Location). This is also included in the datasets. The Element by Location node will take a point location and find the family at that location. Once we know the family at each location of each list, we can count how many items are in each list and input that into a shared parameter to display in the tag.



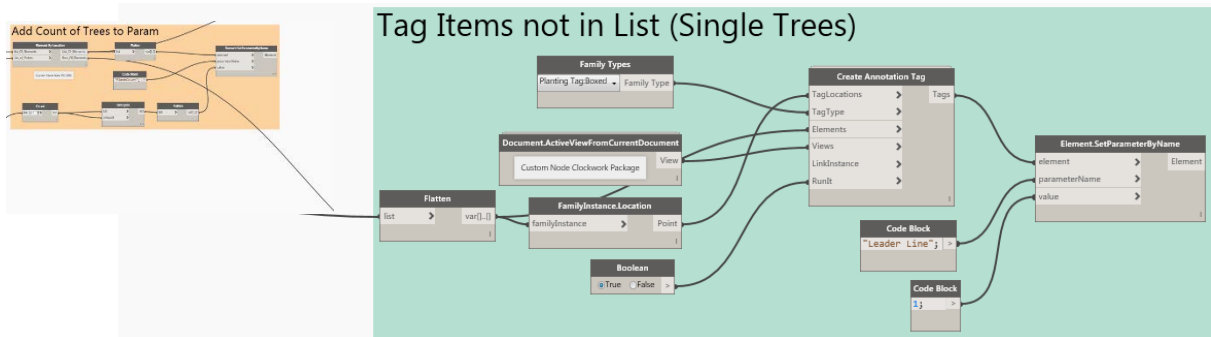
**7) Tag one item from each list**

This group will take the first item from each list of families. Since we've sorted our list of families by the list of lines that we drew, we can pull one item from that list and tag it.

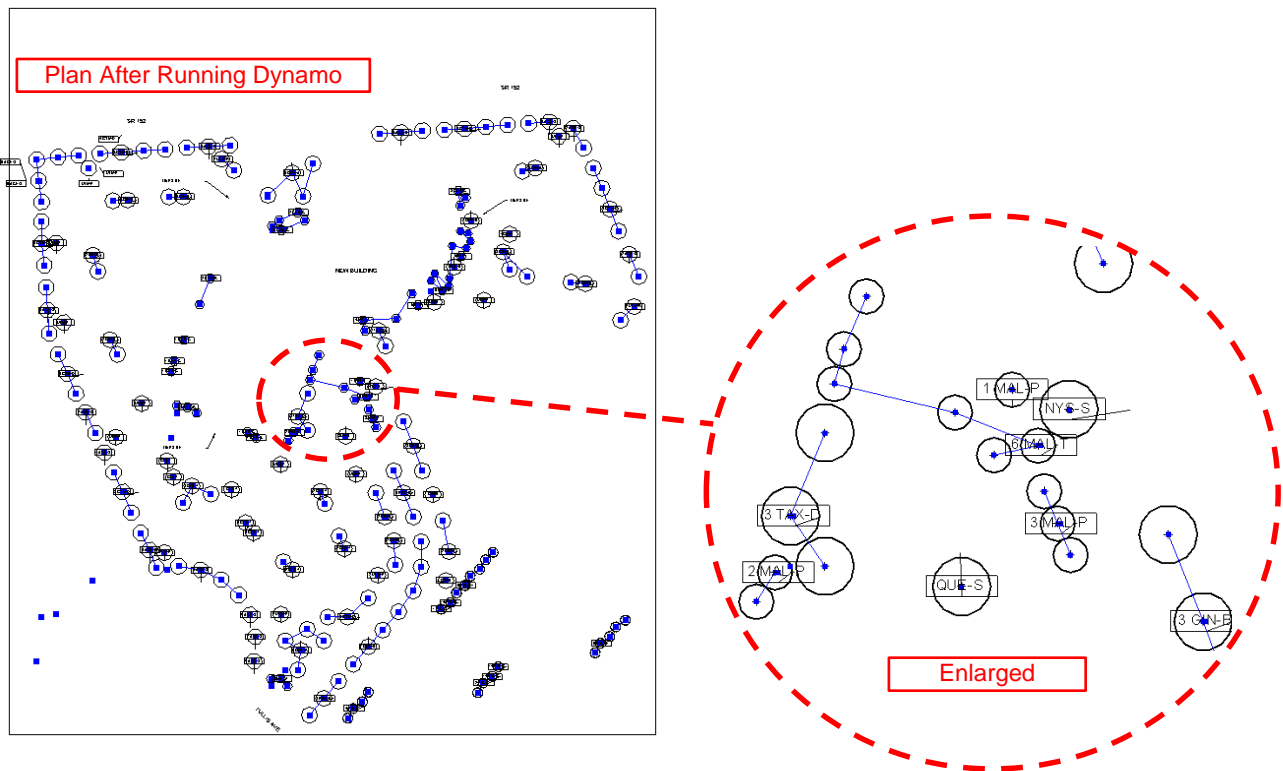


### 8) Tag the families that weren't part of a chain of lines

Back in group 7 there was a node that found the families that were part of the list of lines. This node also will output the list of families that were not associated with the list of lines. These families still need to be tagged for our plans so rather than tagging them manually, Dynamo can be used to automate tagging the families.



This takes a few minutes to run but it does save a ton of time. The results do still require some work. If you use this workspace as it stands now, your results will look similar to below. You will want to move tags to avoid notes and other items on the plans.



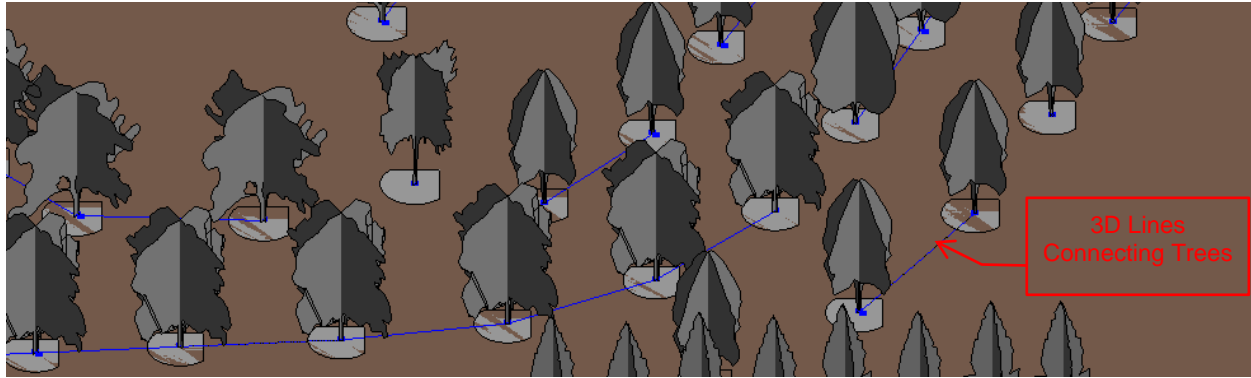


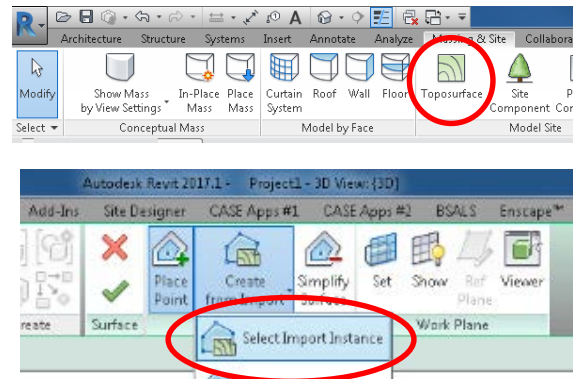
IMAGE OF 3D VIEW FROM REVIT OF LINES CONNECTING TREES

## Creating Topography

Before creating topography, take inventory of what you have and what you need to provide to determine what is the best method modeling topography for your project. Revit's topography tools can be a bit lacking and can be difficult to work with but with assistance from other programs or use of non-topography features, creating topography can be pretty easy.

### Topography from CAD import:

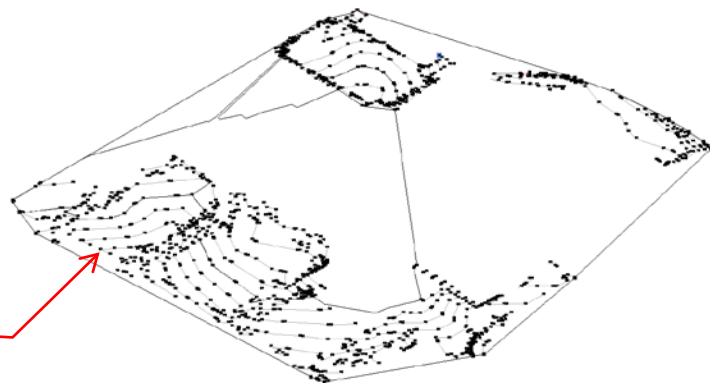
Creating a toposurface from imported .dwg file is a fairly easy process. This is one of my preferred methods because it is easy to do and can be made to update with changes throughout the project. Simply link a CAD file into your model, from the Massing and Site tab, select Toposurface. Then pick Create from Import and select "Select Import Instance" from the drop down. You will be given a list of CAD layers to choose from. Choose only the layers that your contours are on.



This method only works if your contours in the CAD file have elevation to them.

If the CAD file is flat, a quick fix is to move the contour lines to their elevation in the CAD file then link into Revit to create topography.

Revit will place points along the selected line work in the CAD file



You can read more about this process here:

<https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Revit-Model/files/GUID-D7C6E4AF-D902-478D-92C2-CD2DCE338426-htm.html>

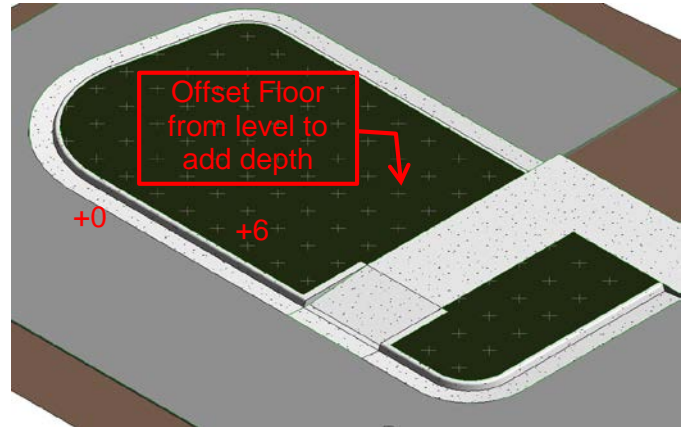




### Topography from floors:

If a site is mostly flat and you are using the site for visualization purposes or not required to provide elevation coordination from the Landscape drawings, using floors to create a site plan is a quick easy method.

Floors have a thickness and material properties. The offset parameter of floors can be used to quickly offset one material from another. As shown in the image to the right, the grass is 6" higher than the asphalt.



### Create topography using dynamo:

Dynamo can be used to create topography by importing the osm data from mapping sites using the ELK package thanks to Timothy Logan. I'm not going to write on this much other than it's pretty amazing.

Here's a link to a video that Timothy posted on Vimeo showing how this can be done.

<https://vimeo.com/143624038>

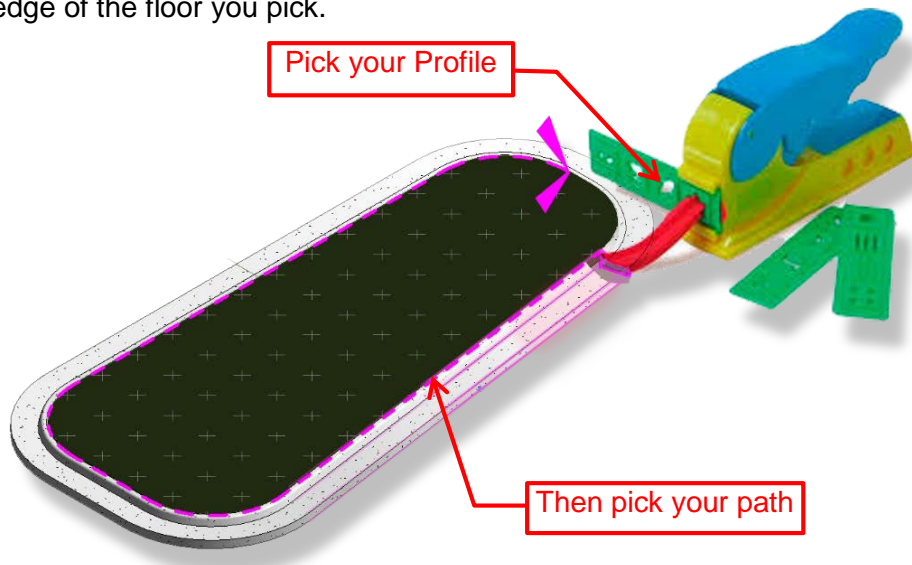
I've tried this myself and it works very well. I have found the site to be heavy in the model but it can produce some amazing fast results.'

## Modeling Curbs

### Slab edges:

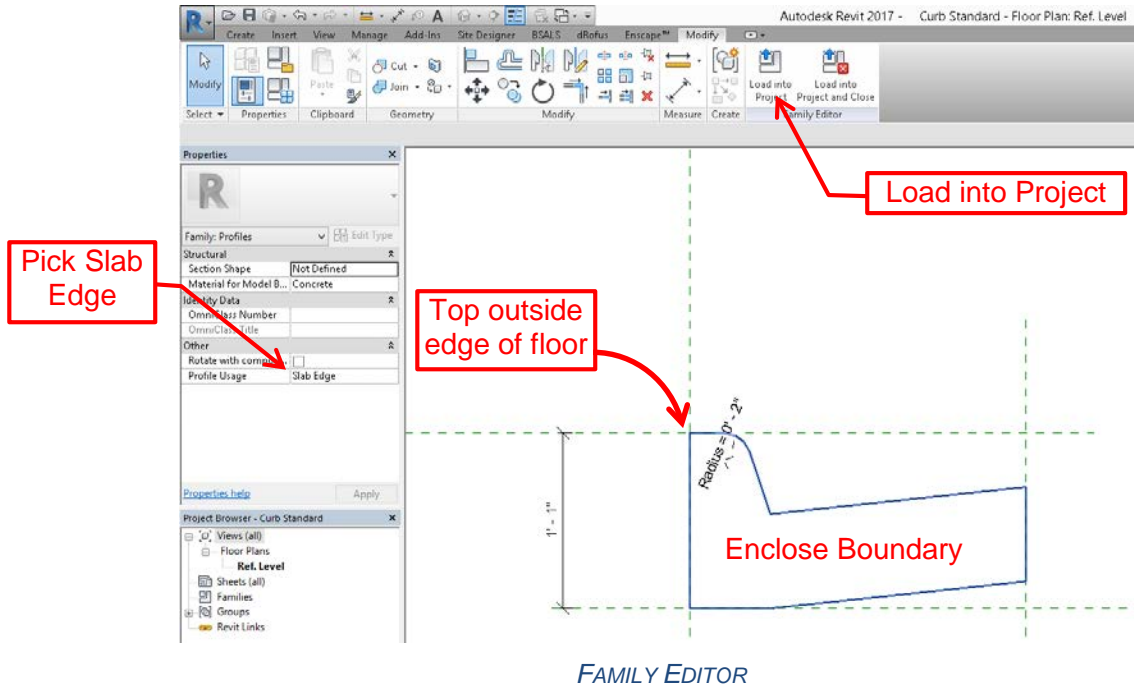
Floor Slab Edges are an easy way to create curbs. A slab edge is made using a sweep where Revit will extrude a profile along a path. You will need a profile family loaded into your project.

The simplest analogy for working with profiles and sweeps is that it's like using a Play-Doh press. The profile is the shape you want the extruded material to look like. If you draw a cross section of your curb, Revit will extrude that shape along a path. In the case of a slab edge, that path is the edge of the floor you pick.

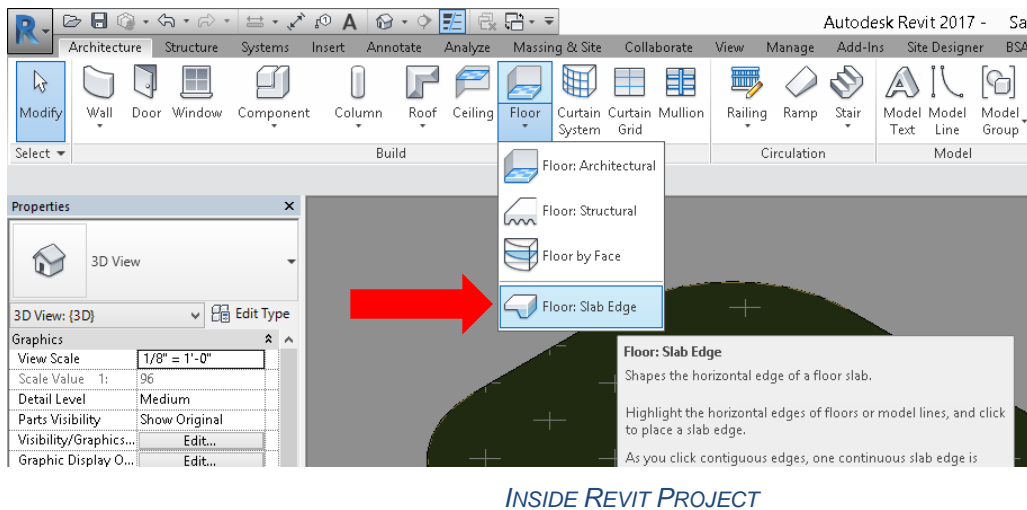


### Working with slab edges:

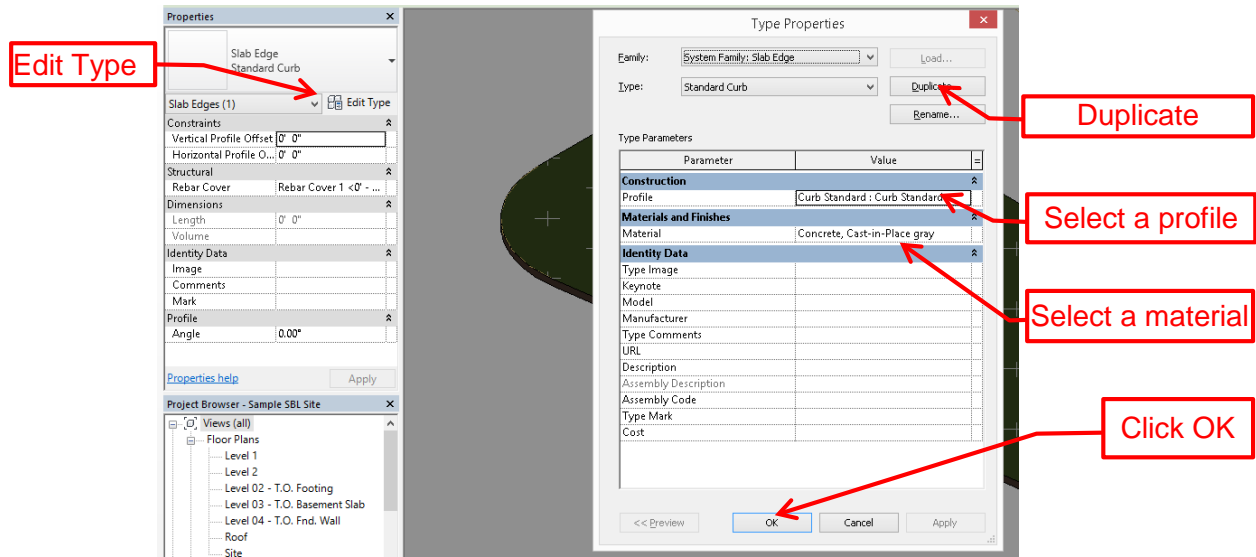
If you do not already have a slab edge in your project, you will need to create one. You need to have a profile loaded into the project. If you don't have a profile family, click on the Application Menu (The R), select New then family. Browse to your family templates and select profile. Draw the outline of the curb you hope to use. The lines need to be an enclosed loop. Save and name the family then load it into your project.



To create your slab edge, go to the Architecture Tab on the Build Panel under Floor, click the drop down arrow and select *Floor Slab Edge*.

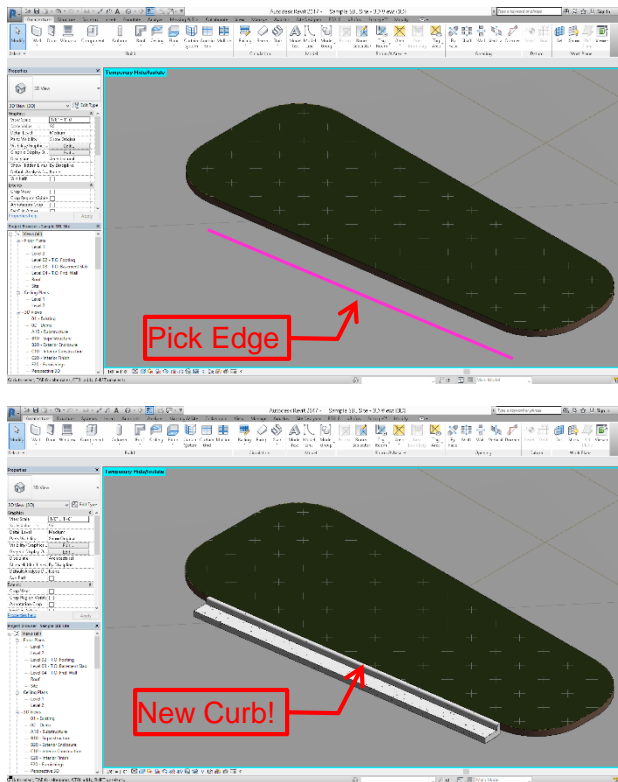


Click Edit Type in properties, then click duplicate and name your new slab edge.

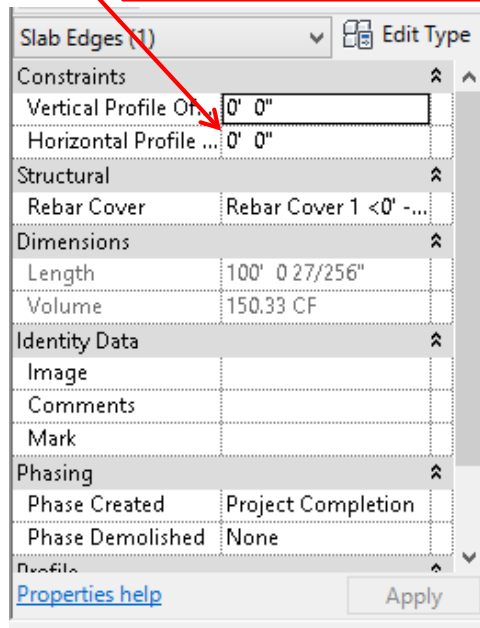


*SLAB EDGE TYPE PROPERTIES*

Then pick the edges of the floor where you want to have a curb. Revit will extrude a solid shape with a cross section of the profile you made along the path you picked. Hit tab to select multiple contiguous edges. Select the new curb and use the properties to adjust the location seen below.



Adjust the curb's vertical & horizontal position here. A positive number will move the curb up. A negative number will move the curb down.



**Modeling curb cuts:**

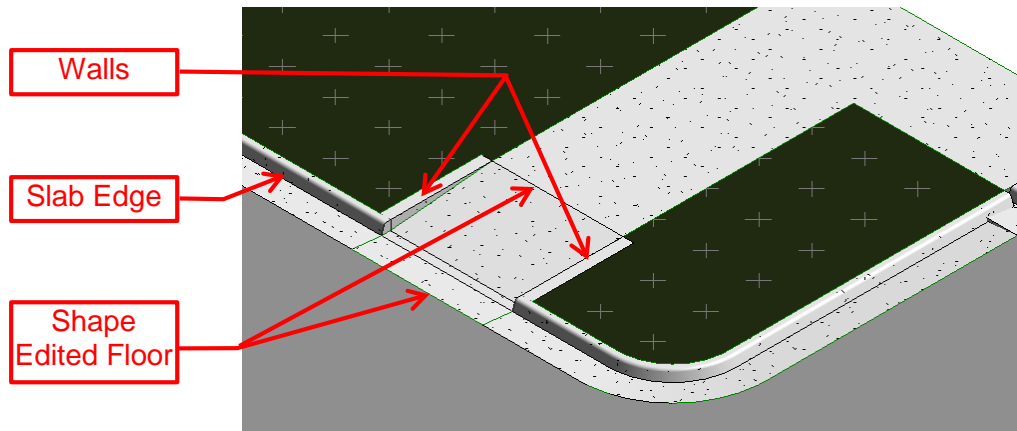
Curb cuts can be tricky to model in Revit because they often vary having differing slopes coming from multiple directions. I primarily use two methods to create curb cuts. One method is to use the [Shape Editing Tools](#) which allow a Revit user to manipulate the surface of a flat

floor by editing the elevation of control points. Points or lines can be added as control points to adjust the height of a floor giving it a non-uniform sloping surface. Another method is to place a surface hosted family that will cut the floor or slab edge curb.

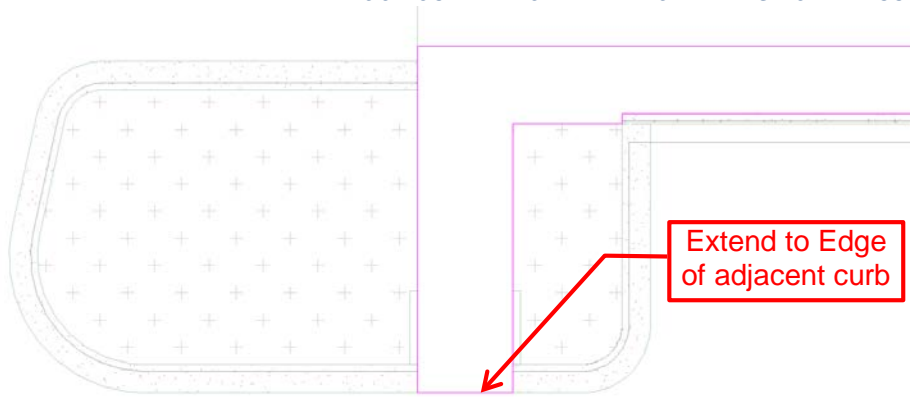
A good and a bad thing about Revit is there are many ways to handle the same situation. My suggestion is determine what makes the most sense for you and your firm. The condition below shows a straight sidewalk intersecting with a straight curve. I'm going to illustrate two different methods of modeling this condition.

### Method One: Edited slab with walls

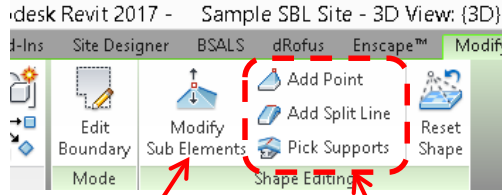
This is an example of using the shape editing tools to adjust the floor to create the slope. This method is slightly more time consuming than placing a surface hosted family. It offers two major advantages over placing families. The first is that it gives the Revit user a ton of control and allows them to manipulate the slab for just about every situation. The other reason is that it will maintain the thickness of the slab in section creating more accurate detailing and quantities in your model.



*DRAW THE FLOOR OUTLINE TO EXTEND TO THE EDGE OF THE CURB.*



*SKETCH MODE IN REVIT*

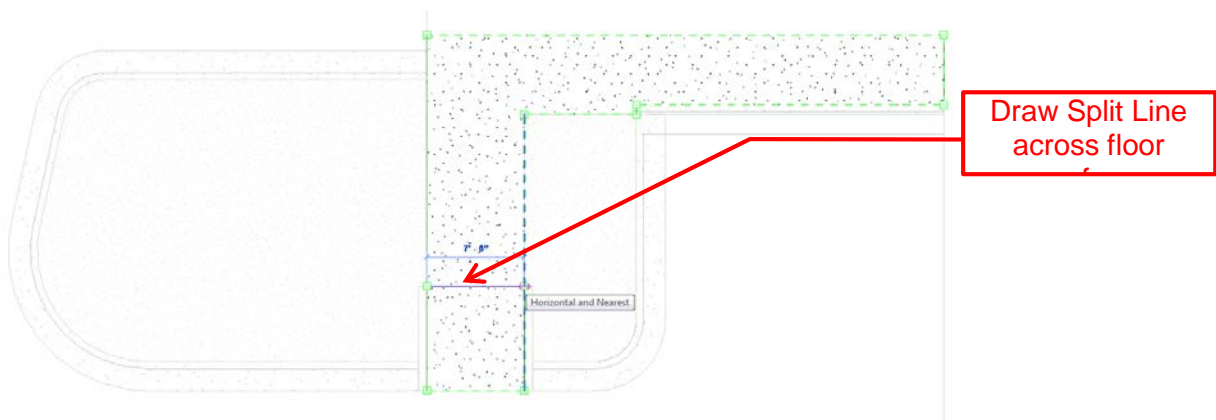


This button will allow you to select and edit any of the features added on the right

These buttons add sub elements that can be used to manipulate the floor surface

SHAPE EDITING PANEL IN REVIT

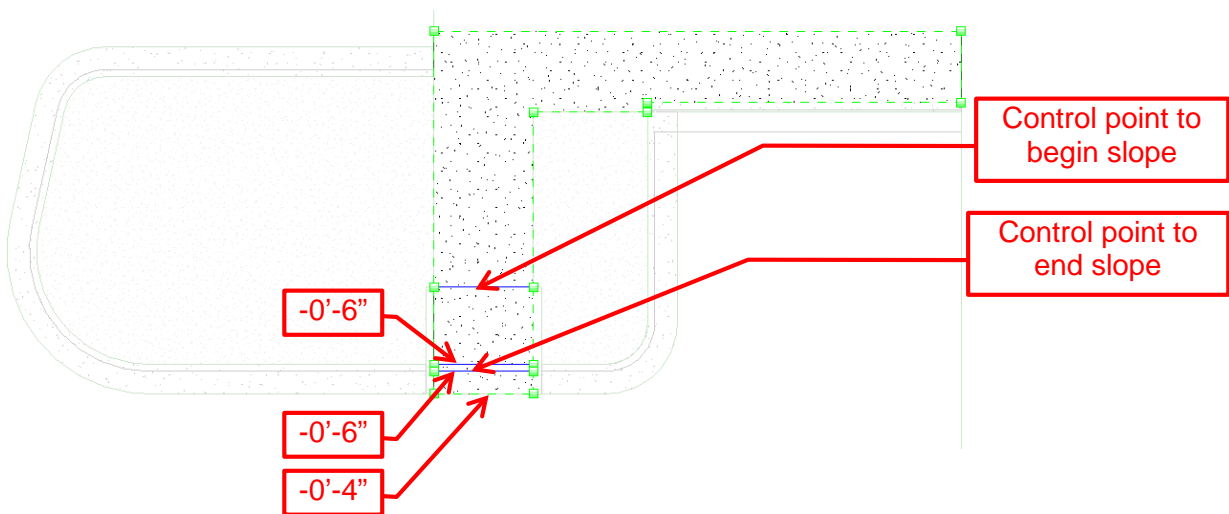
Select the floor then click Add Split Line from the shape editing panel on the modify tab. and draw lines at boundaries where you want to control the elevation of the floor slab.



Draw Split Line across floor

IN SHAPE EDITING MODE (CREATE CONTROL POINTS)

Add Split Lines where slopes start and stop. Then click Modify Sub Elements and pick the lines that need to be adjusted. Input the desired distance from the current surface for each line. For my example I used the numbers below.



Control point to begin slope

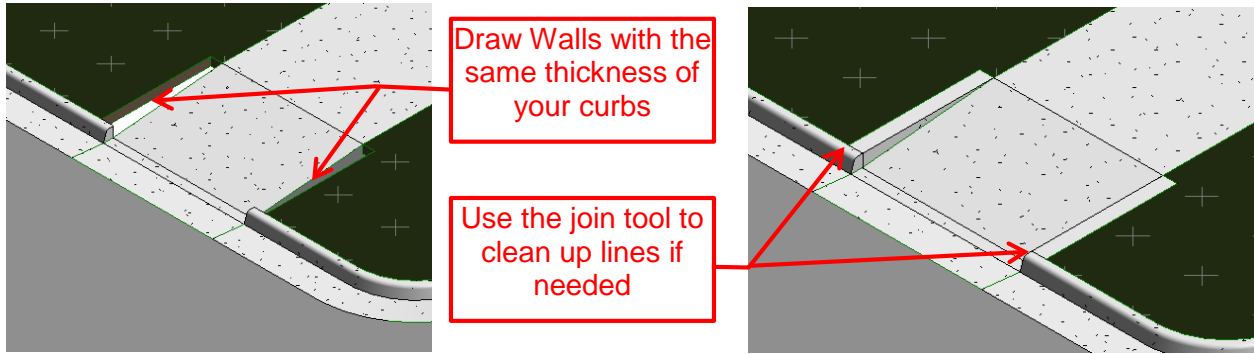
Control point to end slope

-0'-6"

-0'-6"

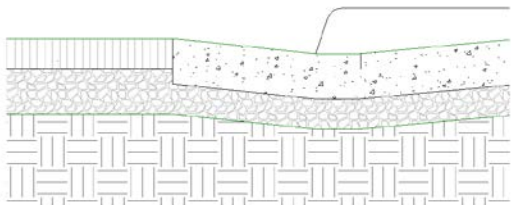
-0'-4"

IN SHAPE EDITING MODE (MODIFY ELEVATION OF CONTROL POINTS)



Looking at a section through the curb cut you'll see the sloping side walk. The images below illustrate an important step in making the structure of floors for sites. The variable thickness option on the floor structure properties determines if a floor tapers when it is added or if it remains a set thickness. The top images below illustrate the difference between having a variable layer for a floor. When the concrete is variable, the floor gets thinner when sloped. This is an important feature to know about and utilize especially if you are using floors for site features. There are times where you would want a layer to vary and times where you would want a layer to be a consistent or minimum thickness.

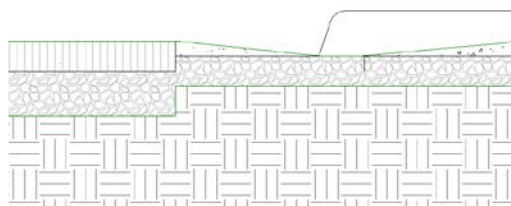
**Variable thickness on the floor slab**



	Function	Material	Thickness	Wraps	Structural Material	Variable
1	Core Bounda	Layers Above	0' 0"			
2	Structure [1]	Concrete, C	0' 6"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Structure [1]	Gravel	0' 4"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Core Bounda	Layers Below	0' 0"			

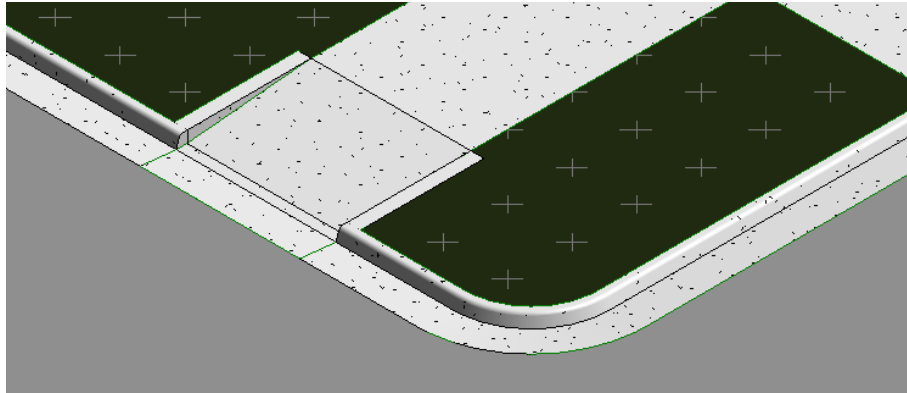
*SECTION AND PROPERTIES WITHOUT VARIABLE THICKNESS*

**Variable thickness on the floor slab**



	Function	Material	Thickness	Wraps	Structural Material	Variable
1	Core Bounda	Layers Above	0' 0"			
2	Structure [1]	Concrete, C	0' 6"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Structure [1]	Gravel	0' 4"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Core Bounda	Layers Below	0' 0"			

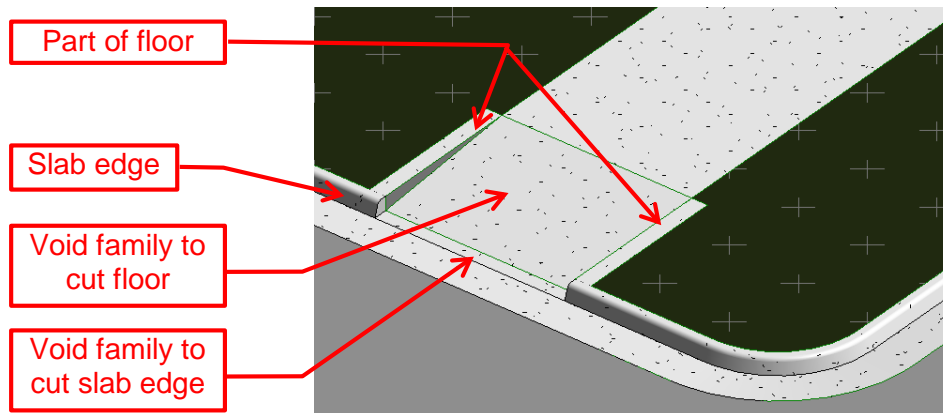
*SECTION AND PROPERTIES WITH VARIABLE THICKNESS*



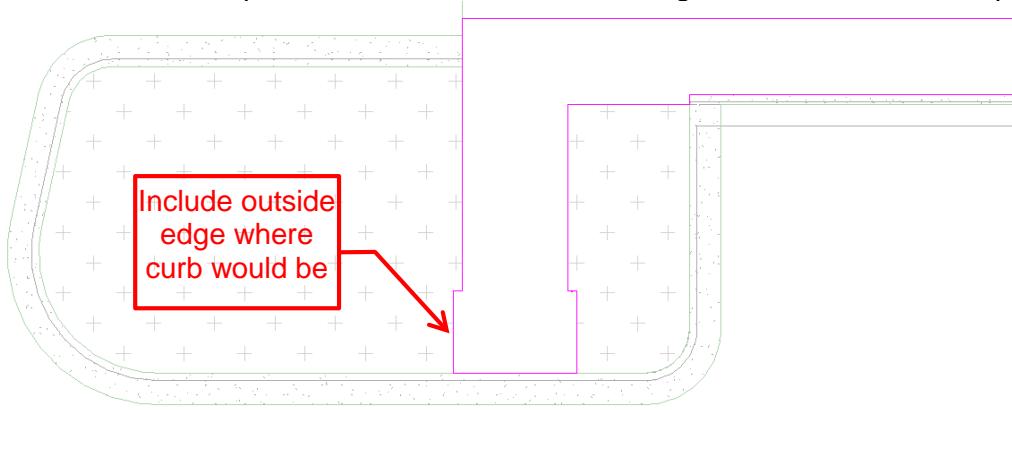
FINAL PRODUCT

### Method Two: Void family to cut floor and curb

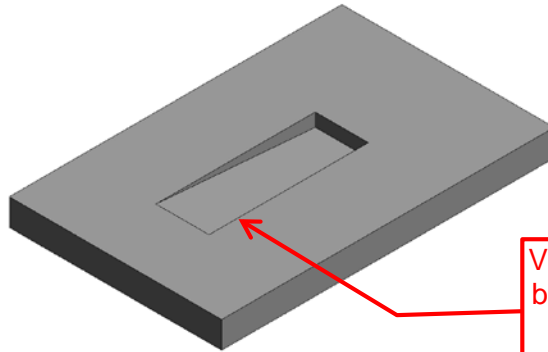
This method can be faster and more easy to update when needing to make changes. This method utilizes face based families to cut the sidewalk and the curb. The issue that I have with this method is that by cutting the floor slab and not editing it, the thickness of the floor varies. This leads to inaccurate details and material quantities (if those features are being used). If you are using the site for imagery only or modeling context, this method is a little quicker than sloping the floor by using the shape editing tools and offers consistency in appearance and may be a good choice for your project. Because the same family is used in multiple locations to make the curb cut, a change can be made in the family and updated throughout the project.



Make sure the floor perimeter includes the outside edge of the curb of the ramp

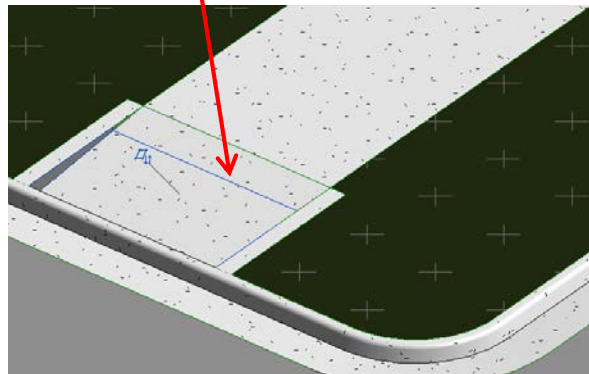
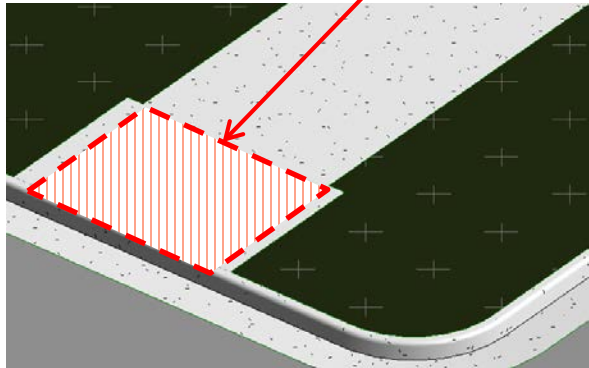


Make a surface hosted generic model family. Model a void extrusion to cut the surface of the floor. In this case I modeled a wedge shaped to create a sloping surface.

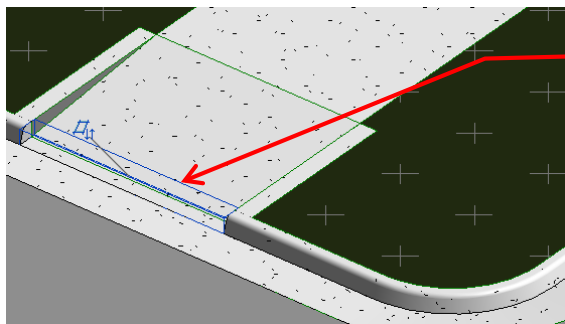


Void Extrusion. Use the cut button on the modify tab to cut the surface

Use surface hosted family to cut slab

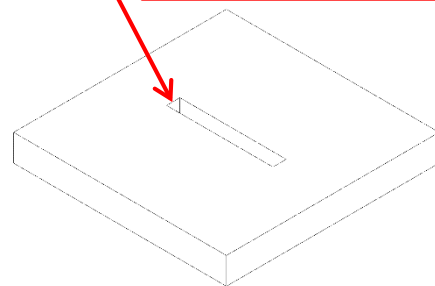


Use another surface hosted void family to cut the curb.

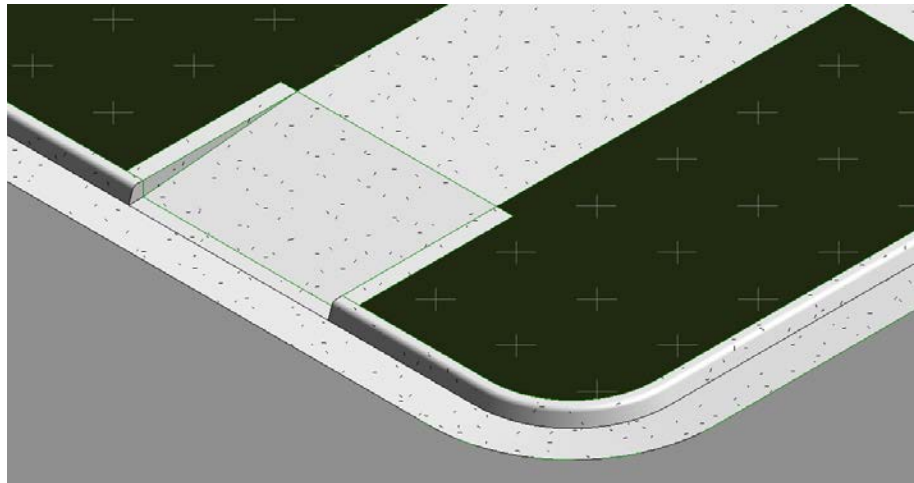


Use surface hosted family to cut curb

Cut Surface in Family







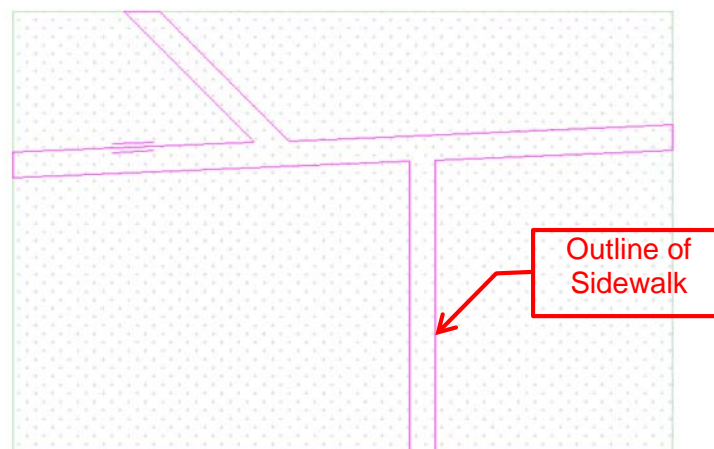
*FINAL PRODUCT (LOOKS SIMILAR TO THE SHAPE EDITED FLOOR)*

## Working With Materials

### Repeating patterns:

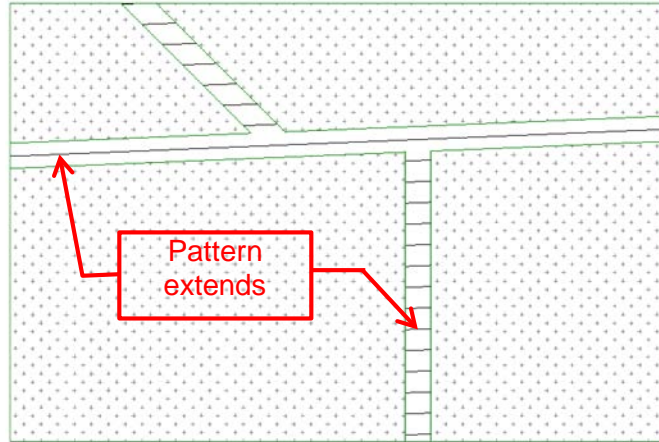
Showing materials in perspective is an advantage of using Revit to design and document your building. A common difficulty that Revit users experience when working with materials is when a repeating pattern is required to adjust to a break its pattern.

As an example a sidewalk has control joints at set intervals. Sidewalks created using the floor tool are usually drawn as a continuous loop which produces a single material as shown below.

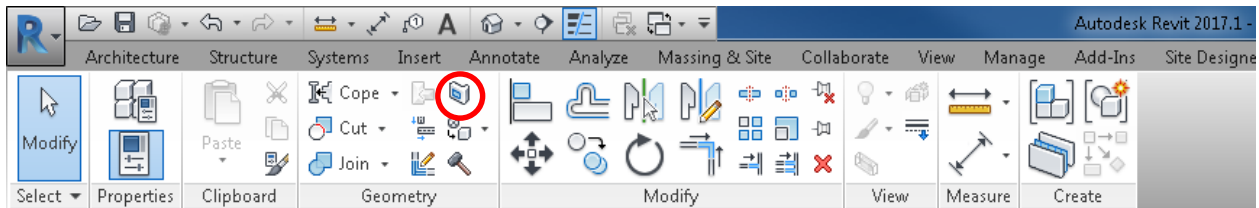


*SKETCH MODE OF A FLOOR*

When finished, this shows as a continuous material pattern. I'm using a model pattern to show the reveal lines at 60" increments. Because this is a continuous sidewalk, the sidewalk pattern shows the pattern running in one direction even though in real life, the control joints would run perpendicular to the length of the sidewalk.

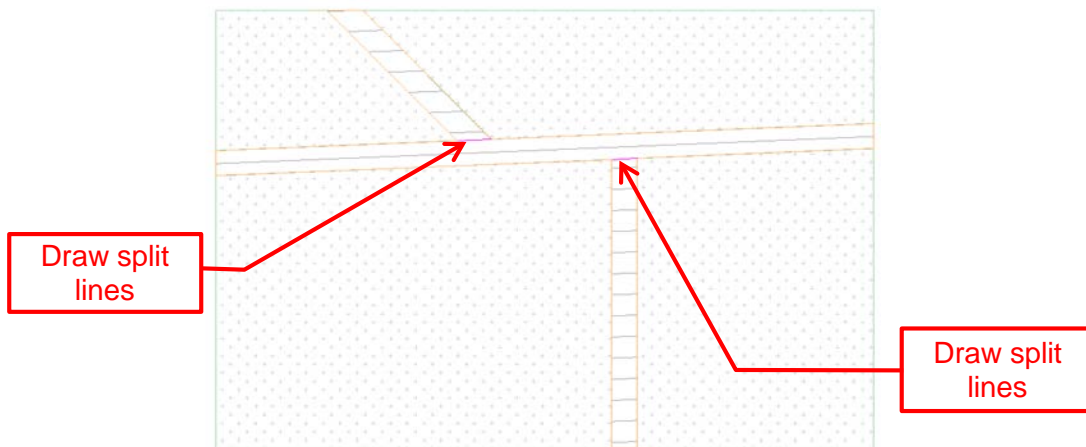


In-order to make the continuous surface pattern break, we need to make it not continuous.



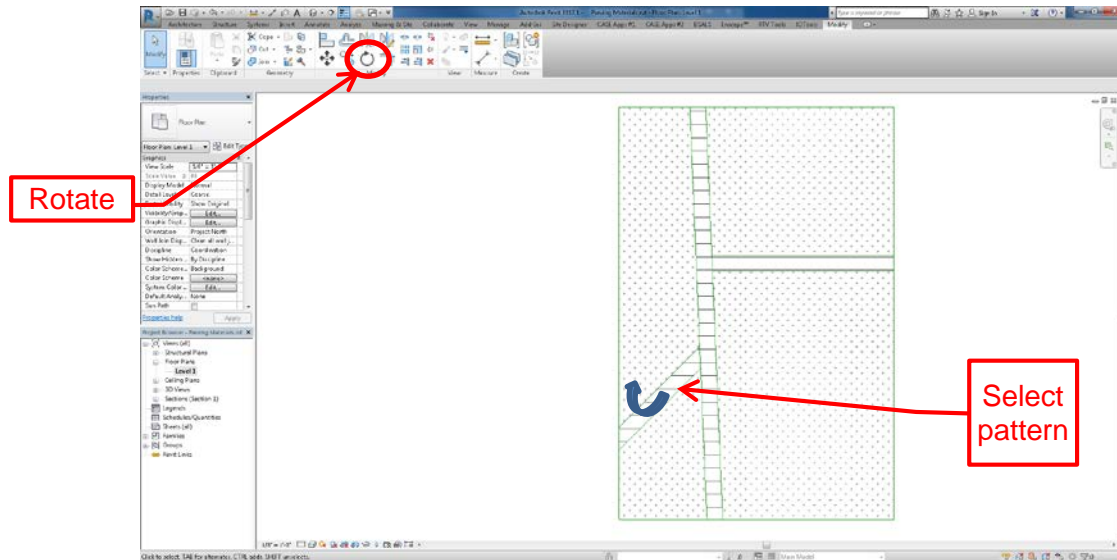
*MODIFY TAB*

This is easy to do using the Split Face tool. You can find this on the Modify tab, Geometry panel, select the Split Face Button. Then select the face of the sidewalk and draw lines where the pattern should split. In the image below, I have drawn lines at the intersections of what would be intersecting sidewalks.

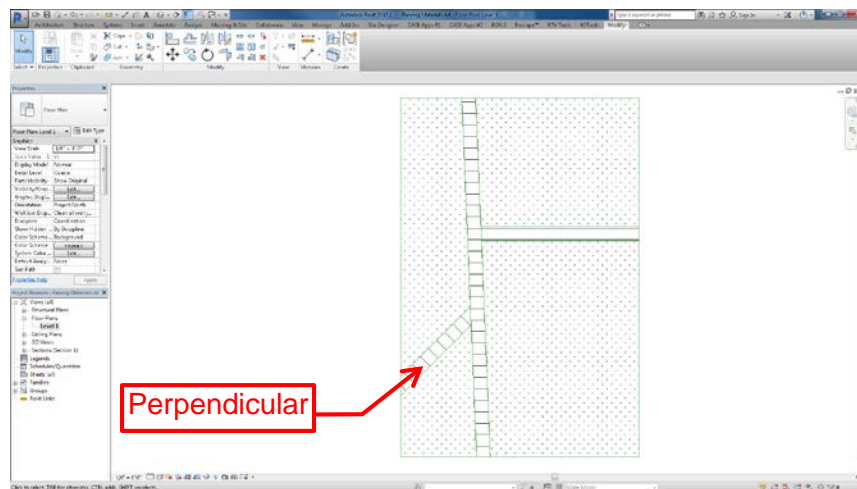


*IN EDITING MODE*

From the Modify Tab, Geometry Panel, the Paint Bucket tool can be used to apply a separate material to the individual surface areas. Once you've done that, you will be able to select the surface pattern and rotate the lines to be perpendicular to the direction of each sidewalk.

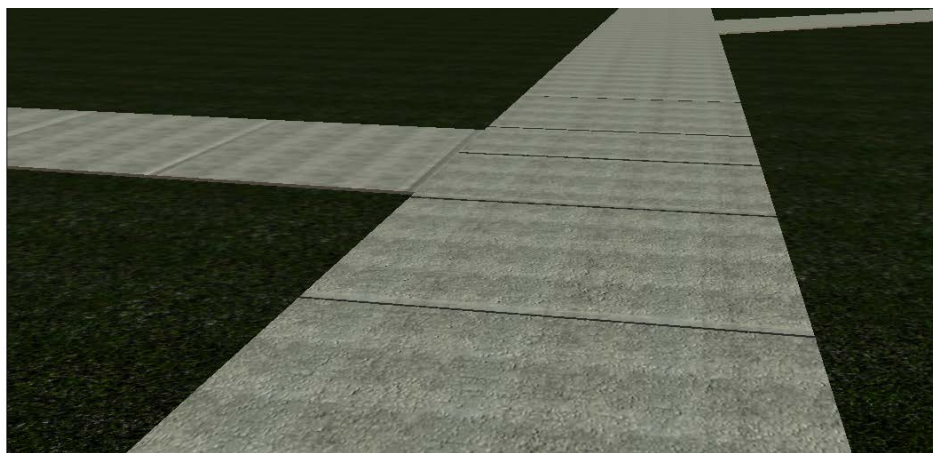
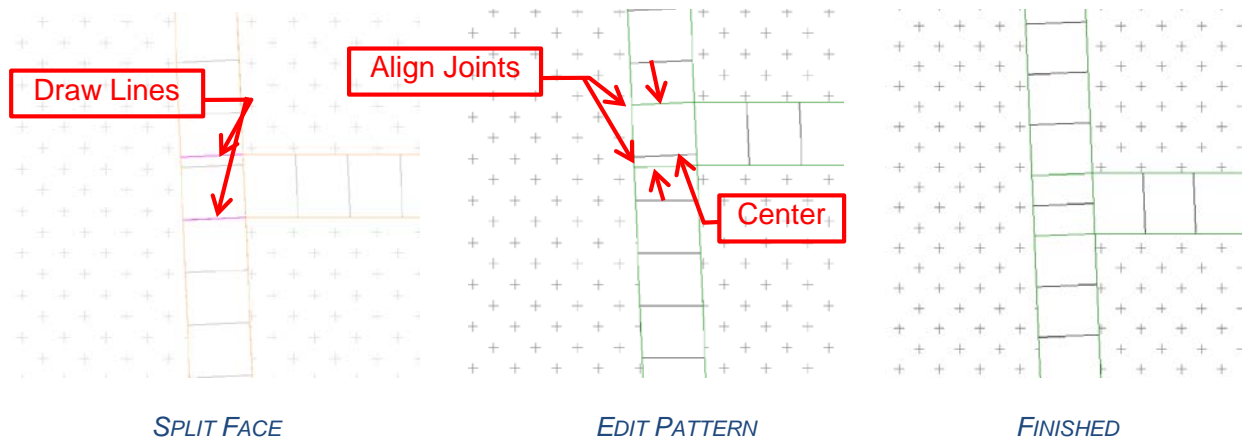


*HIT TAB TO SELECT LINE OF PATTERN*



*USE THE ALIGN AND MOVE COMMANDS TO ADJUST PATTERN*

The images that I am showing have 6' wide sidewalks. Where the horizontal running sidewalk intersects with the vertical running segment I wanted to have control joint. However, the pattern is only 5' wide so it cannot align with both intersection points. In this case, I used the split face tool again to split the surface of the sidewalk material to add control joints at the intersecting edges of the sidewalks.

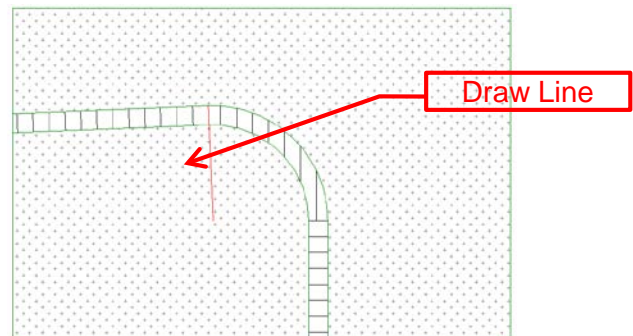
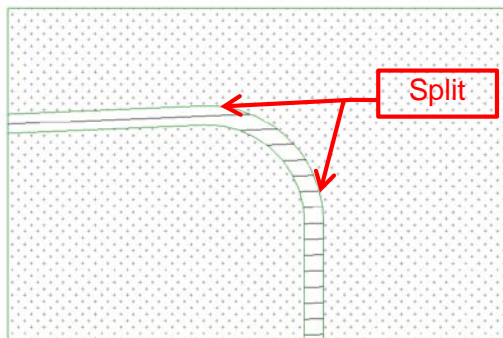


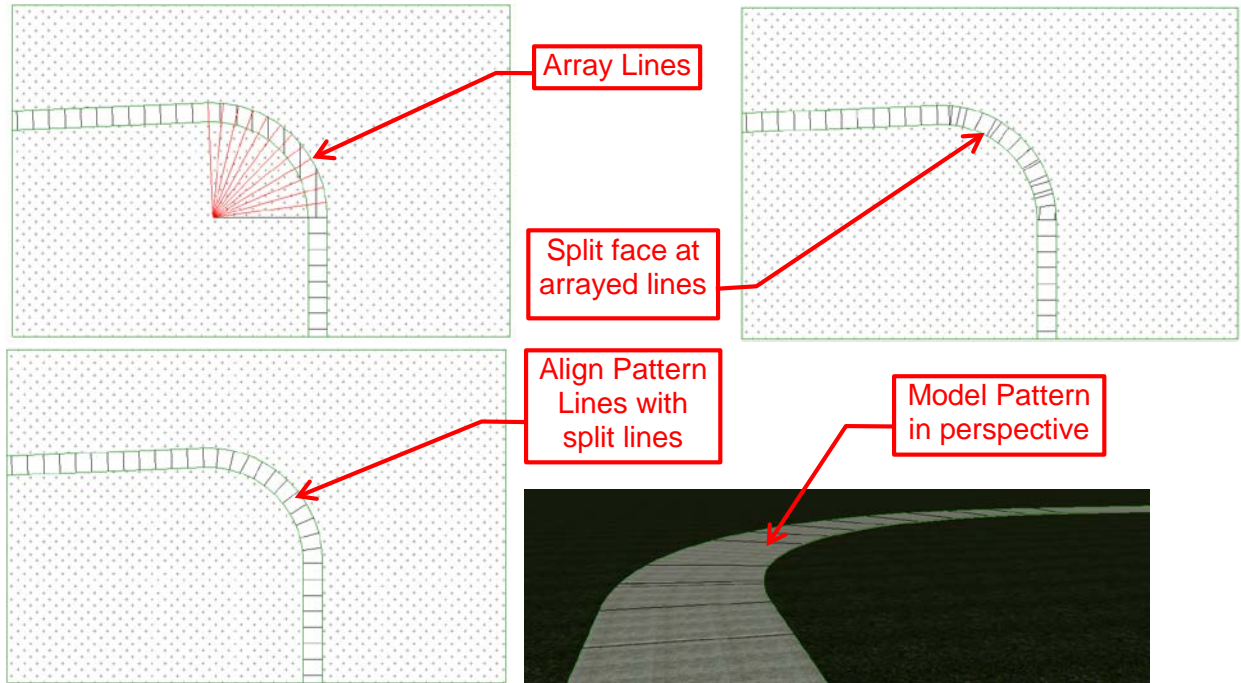
*REALISTIC VIEW: (CONTROL JOINTS ALIGN WITH HATCH PATTERN)*

**But what about Curves?:**

Similar to the previous example, when a repeating surface pattern extends across a continuous surface pattern we encounter issues

Split the face at the start and stop of the curve. Draw a detail line from the center point of the radius to the start of the curve.





### Using the join tool:

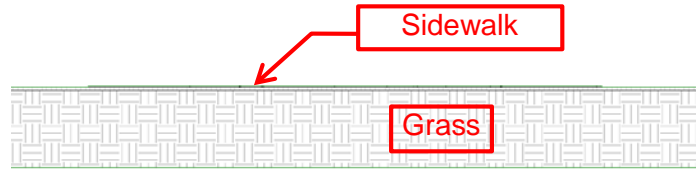
If you are using floors to make your site, this is a tip that you'll find helpful. In the name of saving time and complexity on your project the join tool can be used to separate overlapping geometry. I know this sounds confusing so I'm going to elaborate on this concept a little deeper.

Often, when drawing the site, a person will draw enclosed regions for each material. The example shown to the right shows a grass region drawn to not include the area of the sidewalk. Some Revit users will do this and think they're saving some time by locking lines to the sidewalk to have the adjacent material update if the sidewalk updates.

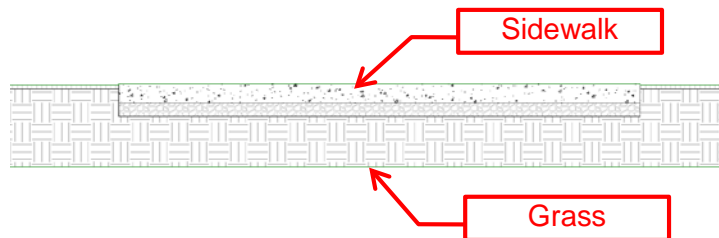
There's a better way. Rather than draw the individual areas of grass, draw the entire rectangle and use the join tool to connect the sidewalk and the grass material. If we look at this in section, (shown below) you can see the grass line through the sidewalk.



If I use the Join tool from Modify Tab, on the Geometry Panel to pick the geometry that I want to join. In this case, the sidewalk and the grass. **The order that you click matters.** If you select the grass first, then the floor slab, the grass will trim through the floor as shown below.

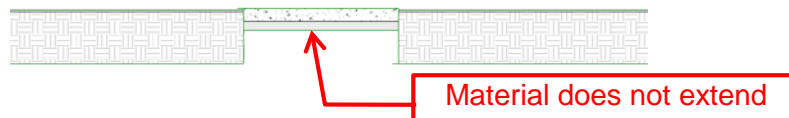


*SELECTED GRASS BEFORE SIDEWALK (INCORRECT):*



*SELECTED SIDEWALK BEFORE GRASS (CORRECT):*

When floors are joined, if the outline of the sidewalk gets modified, the floor will still trim the grass area wherever updates occur. The image below shows what a segmented floor site looks like in section. You can see the ground layer is shown only where it is drawn. Not only is it a quicker method to join the floors but it also produces better looking sections.



*SIDEWALK AND GRASS DRAWN AS SEPARATE REGIONS*

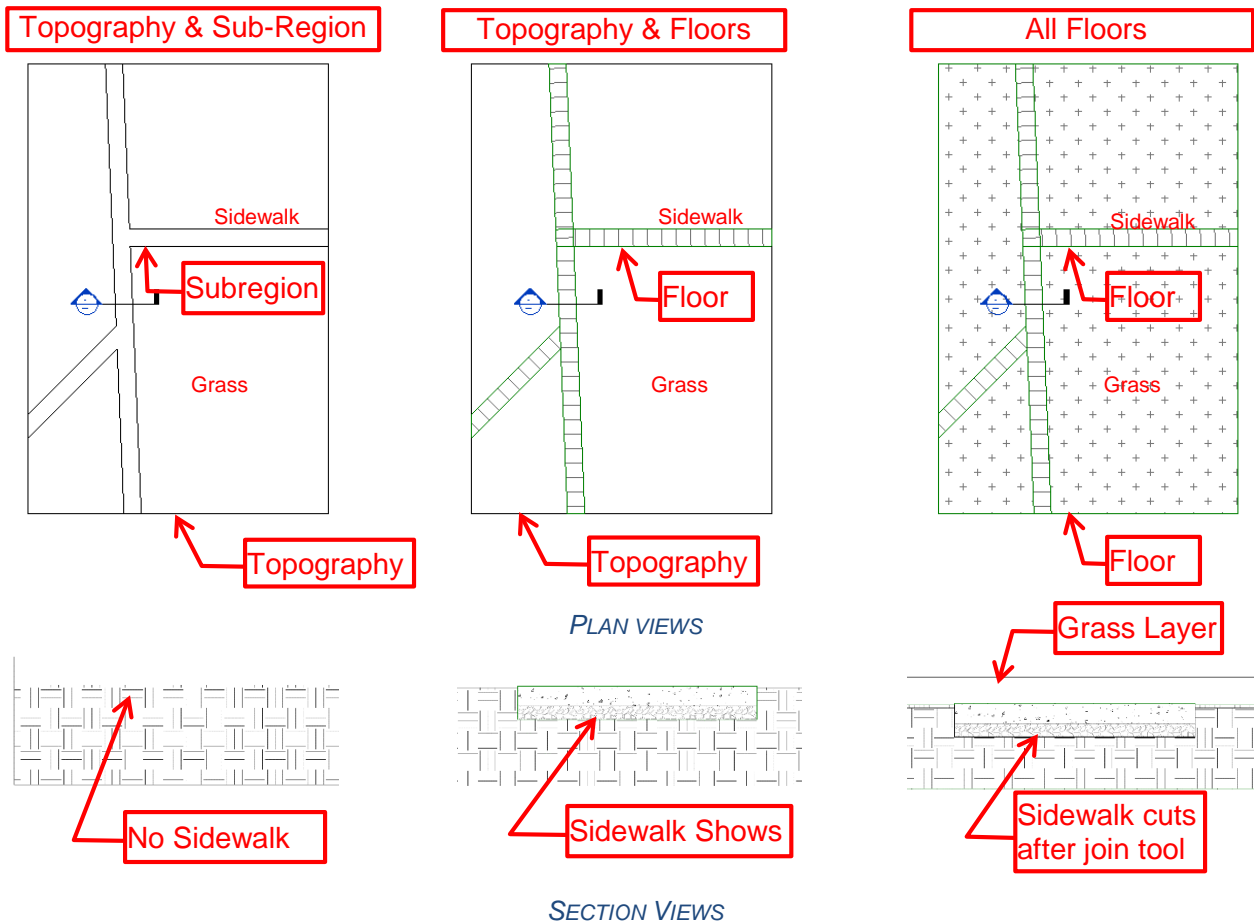
## Hardscape Strategies

### My thoughts:

I've already touched a little on hardscape with the materials and curbs mentioned above but I did want to cover some typical methods of creating hardscape. I prefer to use a combination of Revit topography and floors to build my site. I do this for a few reasons.

- Floors allow me to add structure to the hardscape which is great for detailing and sets me up for more accurate quantity take offs when need be.
- Railings can now host to sloping floors and not topography
- Topography does not support surface patterns. I still do not understand why but here's some information. <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/Revit-Model/files/GUID-5DE536EE-9F9F-4775-9237-16DD4C0BBCA5-htm.html>
- I find editing floors easier (Personal Preference)

The differences:



Topography with a sub-region for hardscape does not show the thickness of the subregion material in section and does not show surface patterns. This is difficult for landscape architecture drawings because patterns are used to differentiate the site. The combination of a floor and topography will show the thickness and structure of the floor but does not show patterns for the topography. Using all floors is difficult when there is sloping topography.

## Hardscape with Dynamo

Dynamo can be used for a number of wonderful things and working with topography is one of my favorite. There have been several posts and presentations on how Dynamo can be used to make things in Revit follow topography, so I won't go in-depth into them but I did want to share some references for anyone reading this and wanting to know how and what they can do.

Make Fences Follow Topography:

<http://therevitkid.blogspot.com/2015/11/practical-dynamo-following-topography.html>

Make Floors Follow Topography:

<https://landarchbim.com/2015/10/21/make-hardscape-follow-topo/>

Make Piles Follow Topography:

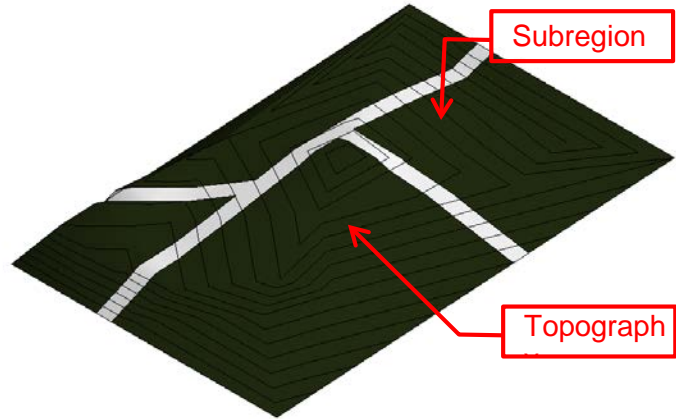
<http://jbdynamo.blogspot.no/2014/11/last-post-about-topos-and-piles-forever.html?m=1>

Make Walls Follow Topography:

<http://therevitcomplex.blogspot.com/2012/07/creating-walls-that-follow-site.html>

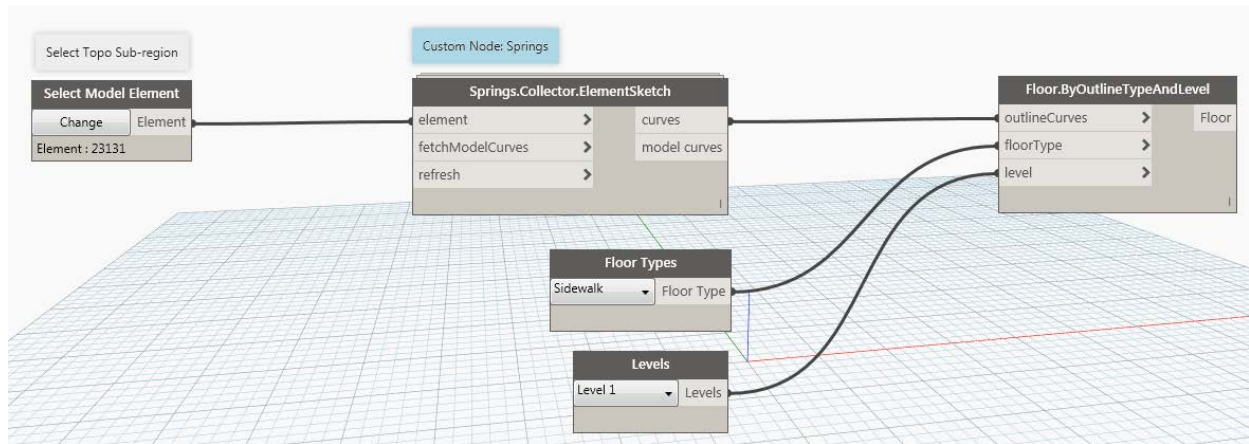
## Making Floors from Subregions:

Topography subregions are great during schematic design for showing hardscape. They follow the contours of the site, are easy to draw, and keep the model light. Where they lack is showing definition against their adjacent surface. Subregions are flat. Because of this I often started my site plans using floors. Thanks to Dynamo I have the ability to transfer my efforts from sub-regions to floors.



ISOMETRIC VIEW

This is a surprisingly simple process with Dynamo. The workspace below is all that is required to make a floor from topography. The select Model element node is used to select the subregion in Revit. The Springs.Collector.ElementSketch node will collect the lines of a Revit subregion sketch. Using the floor.ByOutlineTypeAndLevel node, Dynamo can take the sketch lines, a floor type, and level to make the floor.

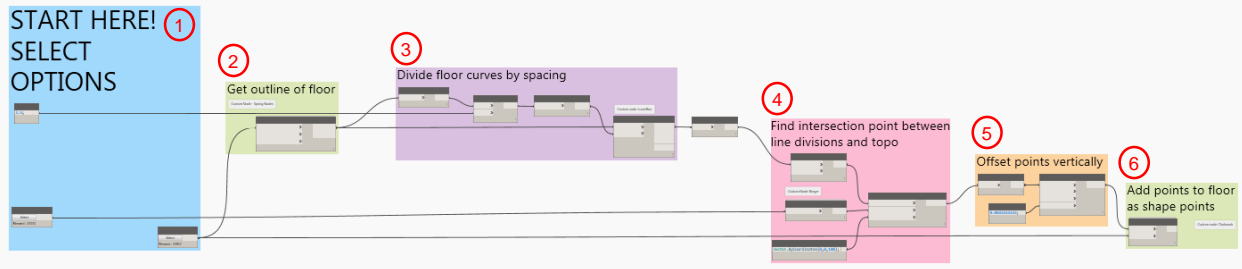


WORKSPACE TO MAKE FLOOR FROM SUBREGION

## Make A Floor Follow Topography:

Once the floor is made, Dynamo can be used to make the floor follow topography. The workspace shown below will have the Revit user select a floor and topography and decide spacing, Dynamo will find the sketch of the floor, divide each line segment of the sketch by the decided spacing. It will then make each points a control point for the floor like shape editing tools to modify the floor slab elevation to follow the selected topography.

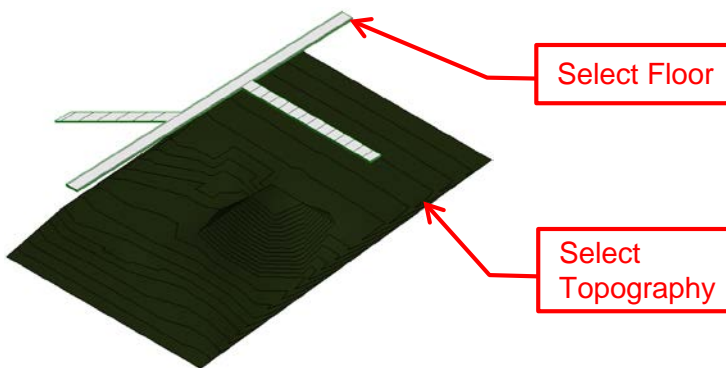




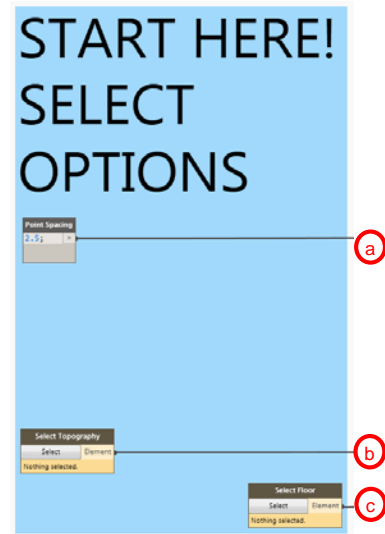
WORKSPACE TO MAKE FLOOR FOLLOW HARDSCAPE

**1) Start Here!**

This group, the select model elements node is used to select topography and the floor that will be edited to follow the selected topography.

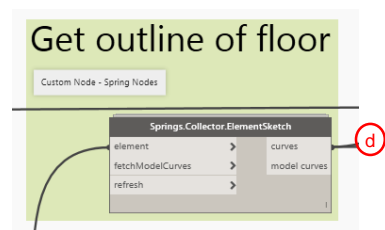


ISOMETRIC VIEW IN REVIT



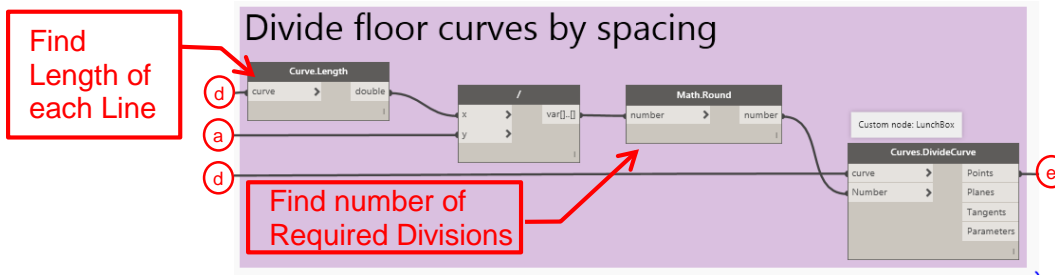
**2) Get outline of floor**

The Springs.Collector.ElementSketch node from the Springs Package will retrieve the sketch used to create the floor.



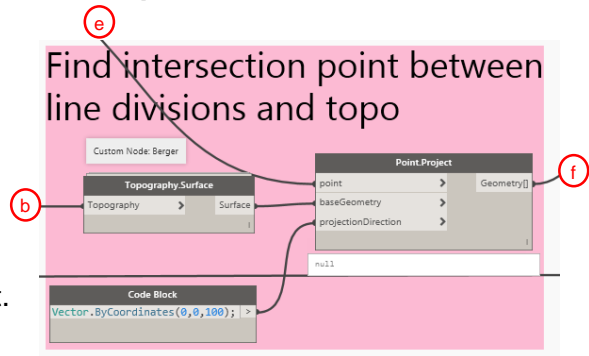
**3) Divide curves by spacing**

This group will take each line and divide the length by the spacing to determine how many times the line needs to be divided. Then the Curve.DivideCurve node from the LunchBox package will create a list of points evenly spaced along each edge of the floor.



**4) Find intersecting point between line divisions and topo**

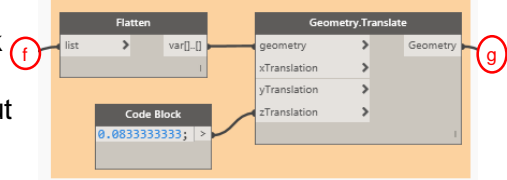
This group will take the list of points and project them until they intersect with the topography and return the list of point intersections.



**5) Offset points vertically**

This group will take the list of intersecting points and raise them by the number we input. The input is based on feet. For this example, we are raising the points by 1" so the sidewalk is above the adjacent grass.

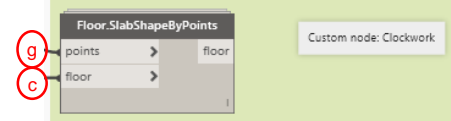
**Offset points vertically**



**6) Add points to floor as shape points.**

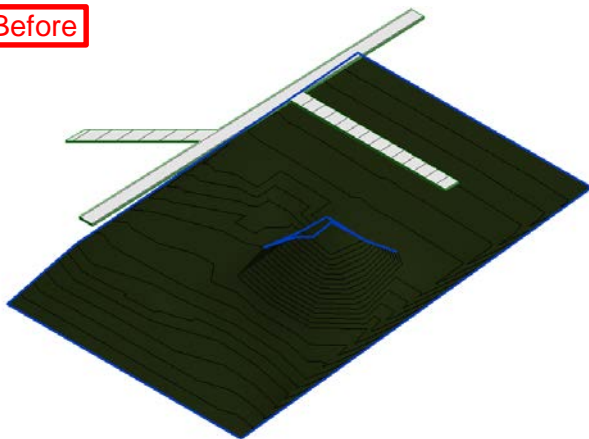
The Floor.SlabShapeByPoints node from Clockwork will take a list of points and add them to a floor similar to the curb cut example earlier in this handout to adjust the floor slab's elevation

**Add points to floor as shape points**

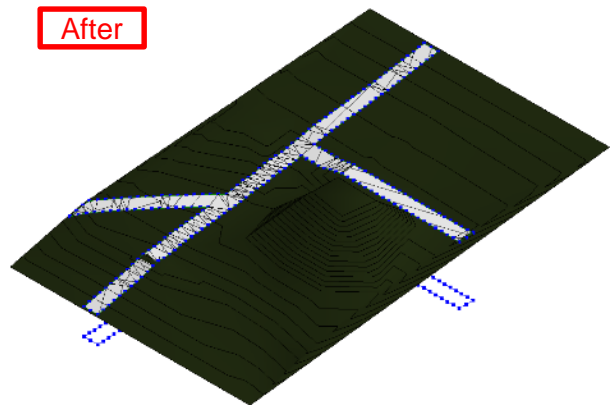


You can see there are some points where the grass is poking through the sidewalk but this is a quick way to have a floor follow a site. Thanks to the improvements in the Revit 2017 release, we can now use Railings with the sloping floor. This opens up a lot of opportunities.

**Before**



**After**

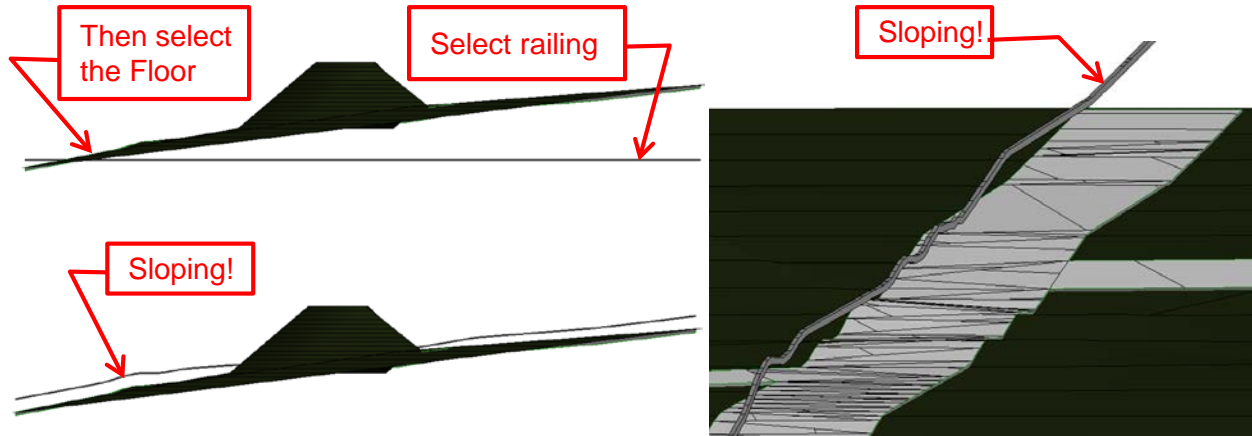
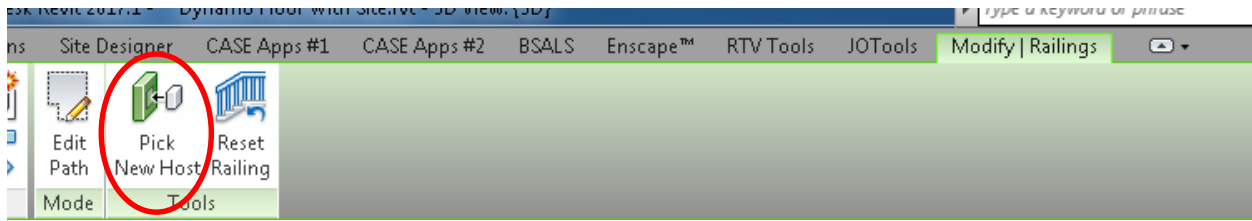


**Using Railings With Sloped Floors:**

This section is going to look at some uses for railings that follow along sloping topography. This technique is great for creating visualization but probably isn't the best for your typical workflows.

## Hosting the railing

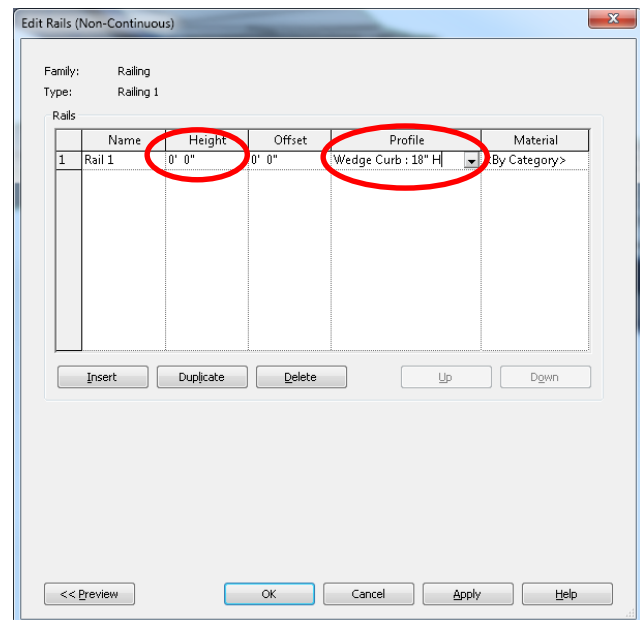
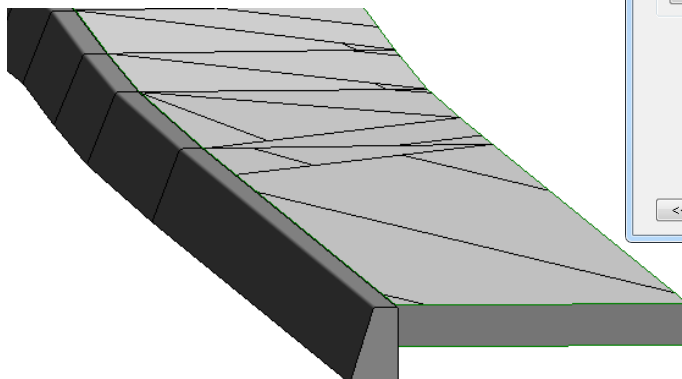
Draw a railing in plan, then select the railing, from the modify tab, select “Pick new Host”, then select the sloping floor. The railing will update and follow the slope of the path.



This opens up a lot of different opportunities. Railings extend a shape along a path similar to a slab edge. Railings also have the ability to place families in an array at a specified distance offset from the drawn path allowing railings to be used to place light ballusters, curbs, trees, and I guess even railings.

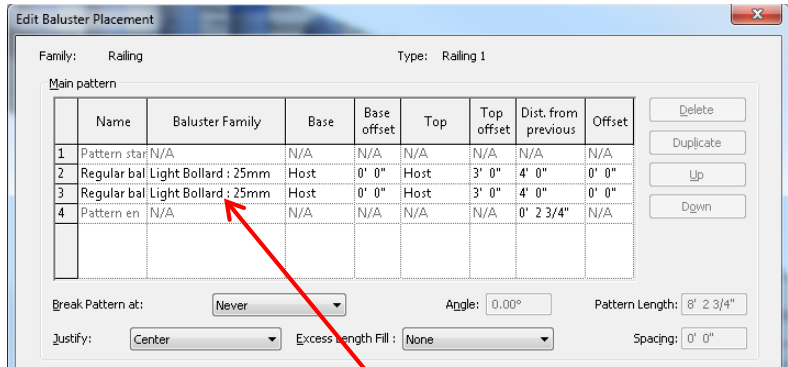
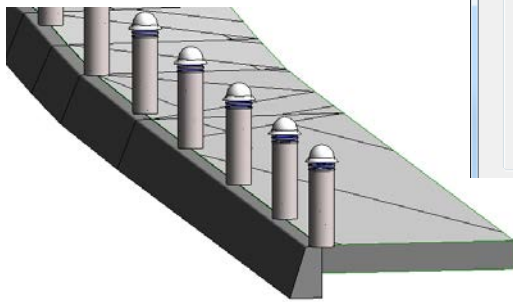
## Curbs:

Make sure your Profile is set to Generic for its type properties. This will allow it to be used as a slab edge or a railing profile. Edit your railing type properties. Within the type editor, edit the Rail Structure and select the profile you wish to use and set the height offset to 0'-0”.



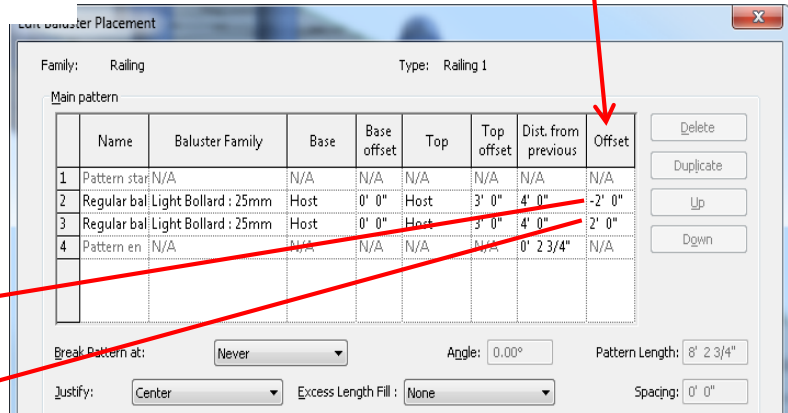
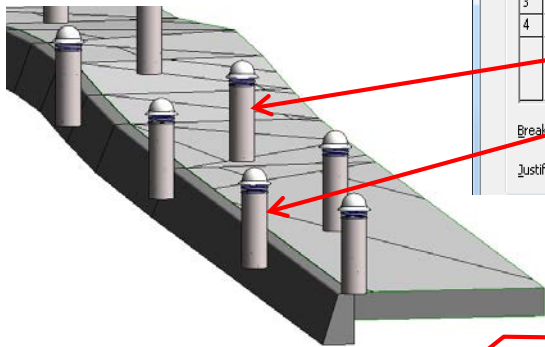
### Balusters:

Create a baluster family and load whatever family you would like to place in an array into the baluster family and load the baluster family into your project. Within the type properties of a railing, edit Baluster Placement and select the new baluster family. For this example I have loaded a light bollard into a baluster family. Use the offset to move the families to the left or right of the rail path.

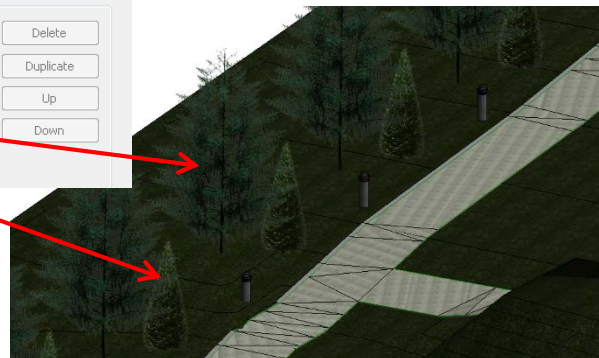
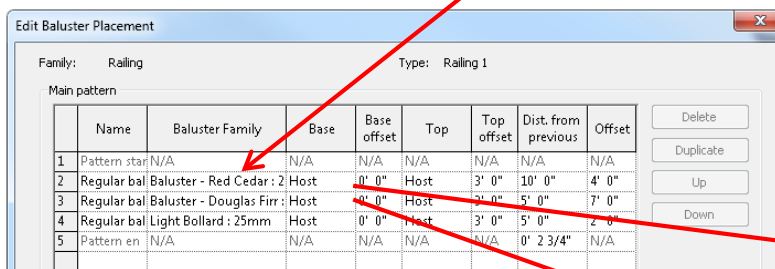


**Nested Family**

**Use Offset to move from**



**Add Trees**

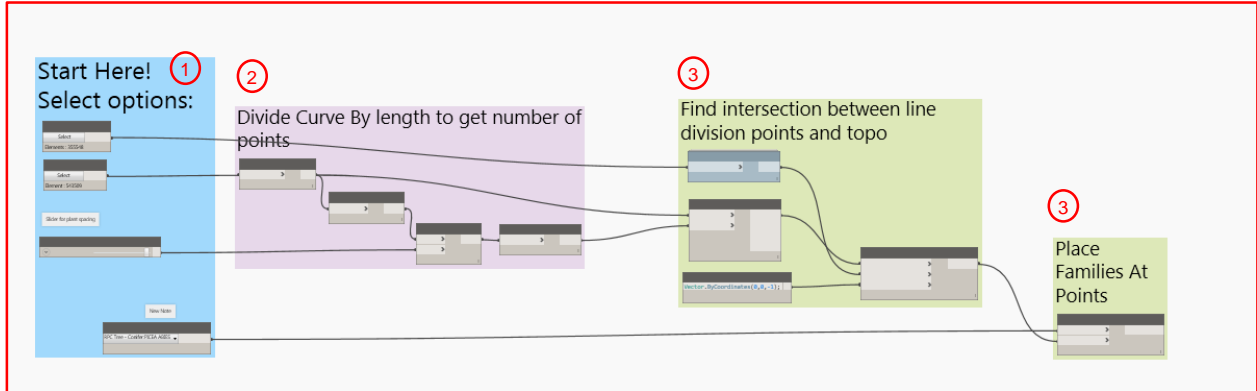


## Designing with Dynamo:

Revit is great at arraying objects and working within dimensional relationships in a linear world. The bad news is that landscape designs tend to be curvy and often not linear. Many of the planting areas have evenly spaced plants in non-evenly spaced or non-rectilinear areas. It's pretty easy to get dynamo to place lines where we have already placed something. It's equally as easy to have Dynamo place families based on other geometry...

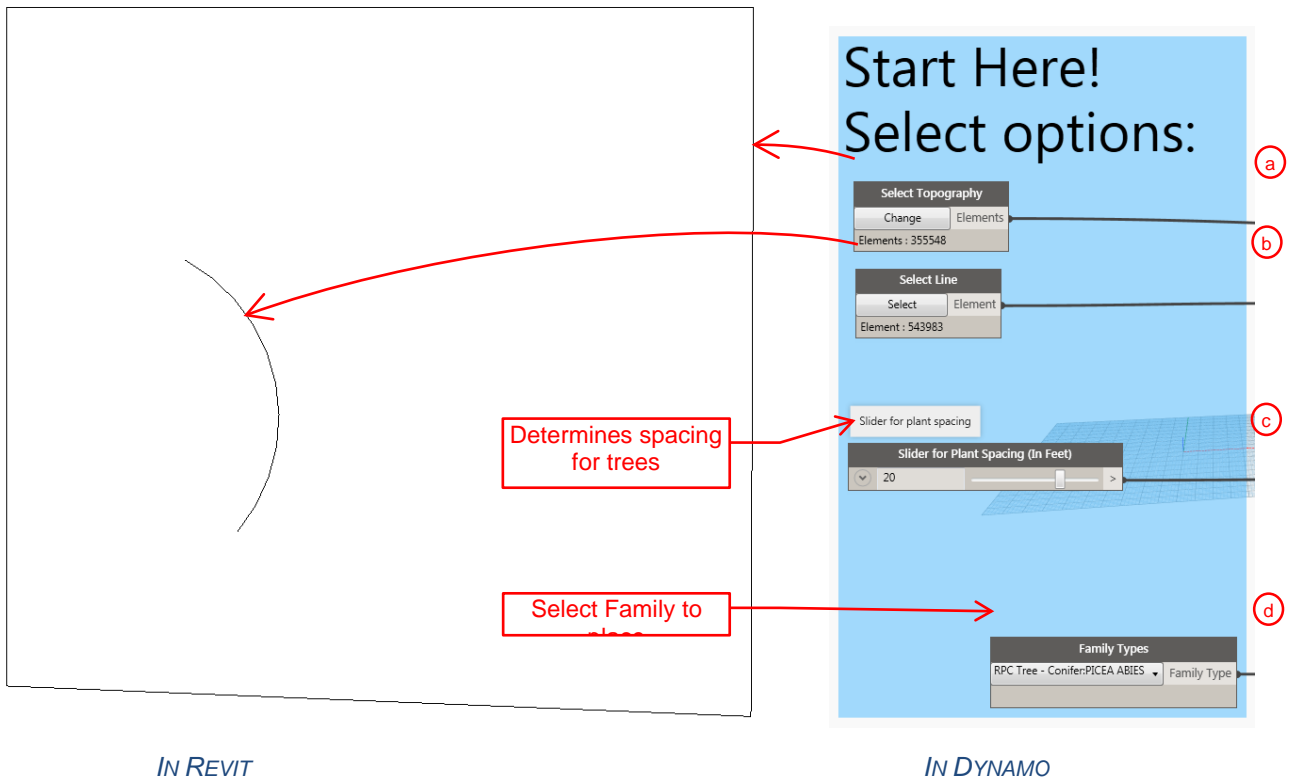
### Select a line and place plants along it:

This workspace does what the title says. The user will select a line, spacing, and a family and Dynamo will place the selected family along the line with the specified spacing.



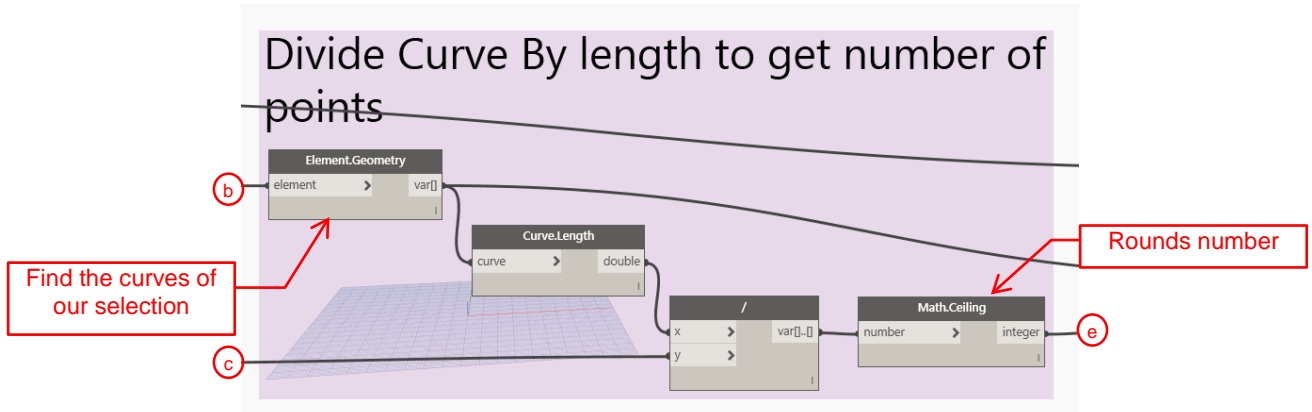
#### 1) Start Here!

In Revit, I have drawn a detail line and topography. The Revit/Dynamo user can select the detail line, topography, input the spacing, and pick the family they would like to place spaced along the line.



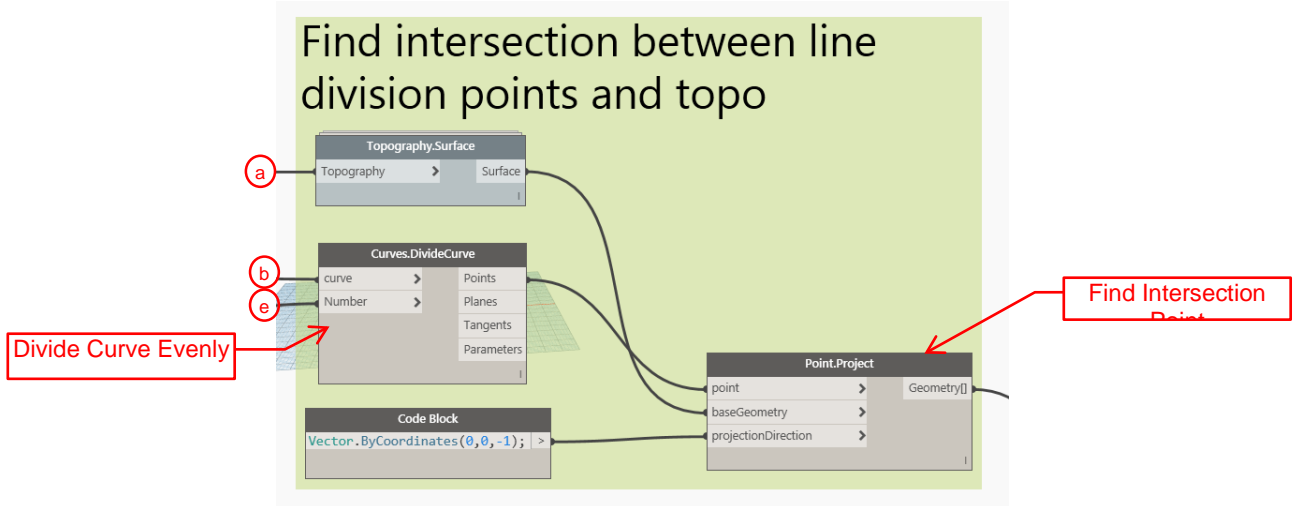
## 2) Divide Curve by distance

The curve divide by length node will divide the selected line based on the specified distance. For this example, I am showing how to equally spread the pattern across the entire length of the line. I will find the length of the curve and divide it by the specified spacing then round it up to an integer resulting in the number of divisions to equally space points not exceeding the specified spacing along the curve.



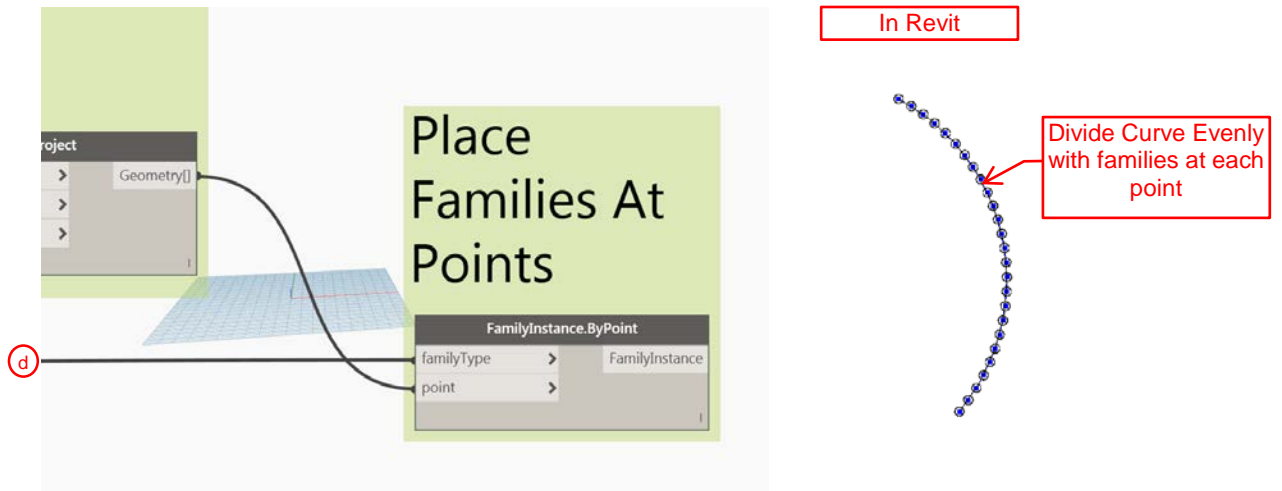
## 3) Find the intersection between the points along the curve and topography

This group will take the points from our divided curve and project them until they intersect with geometry and return a list of the intersection points which is where we want to place trees.

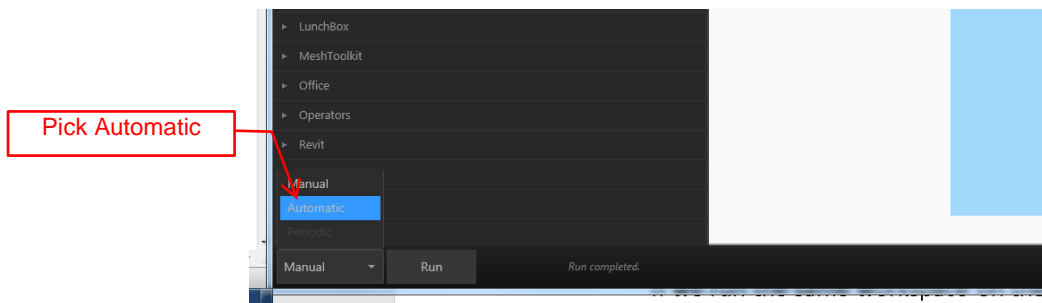


## 4) Place families at Points

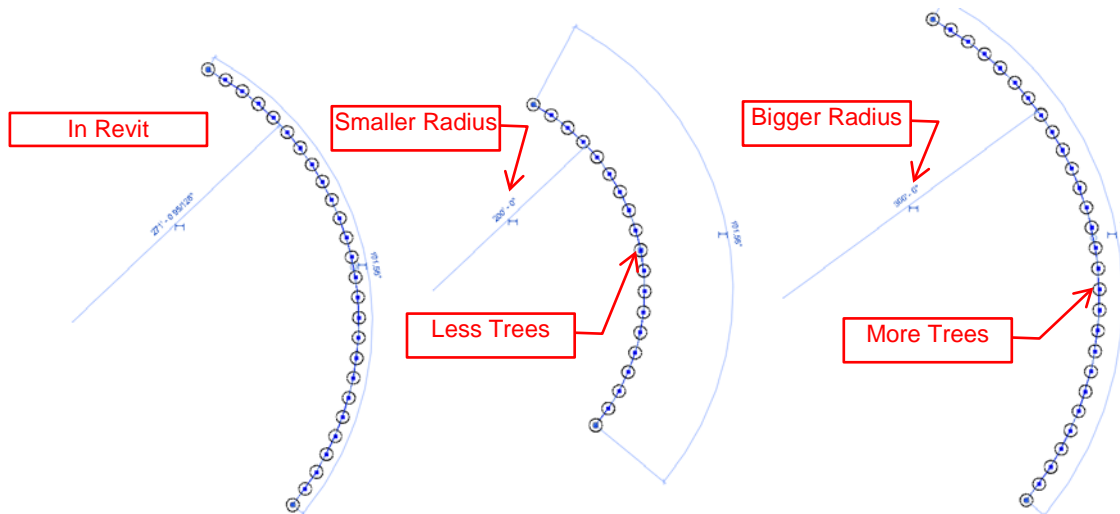
This last group will take list of points from our previous group, a family type, and place the selected family type at each point.



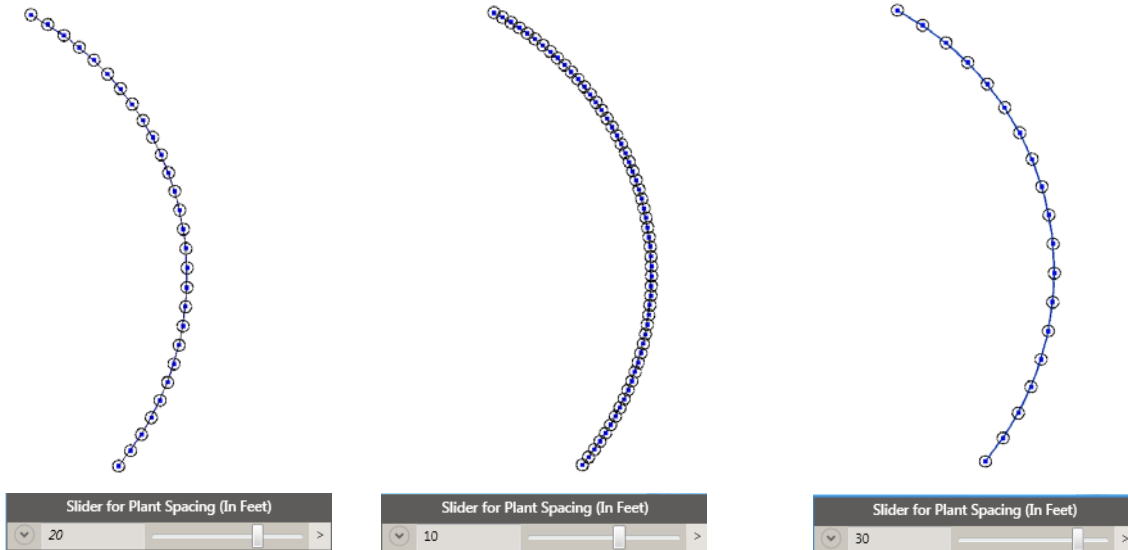
In Dynamo we can change the workspace from Manual to Automatic. This will make dynamo update whenever there is a change to either an input or the geometry it is watching.



With Dynamo in automatic mode, if we change the radius of our curve, the plants will automatically adjust with the same spacing. This can let someone design on the fly.



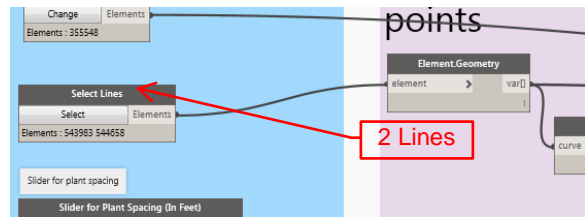
*IN REVIT I CAN CHANGE THE LINE RADIUS AS SHOWN BELOW. DYNAMO UPDATES AS I WORK.*



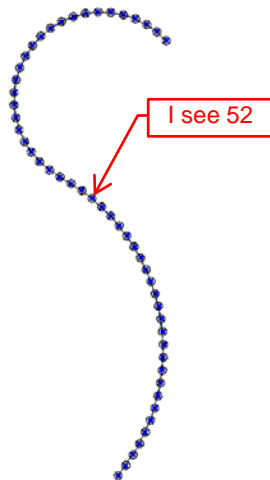
OR CHANGE THE SPACING OF TREES USING THE SLIDER, THE NUMBER OF TREES ON THE LINE UPDATE

### Troubleshooting:

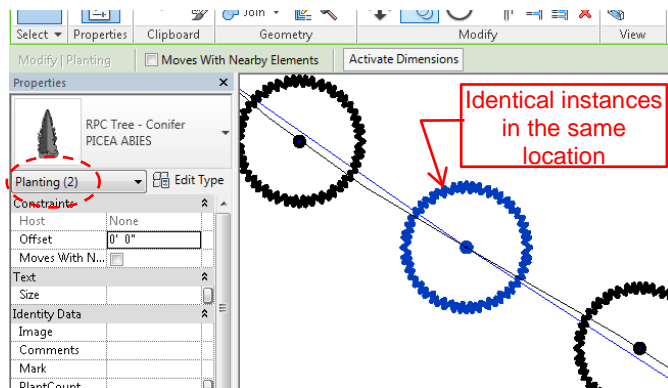
What if we have more than one line? When we select multiple lines and run Dynamo, everything looks evenly spaced. I count 52 trees. However, a schedule to count your plants will show 53 trees. If I select the tree at the intersection point of the two lines, I see there are two trees. The end of the first line and the start of the second line overlap.



A	B	C
COUNT	MARK	BOTANICAL NAME
EVERGREEN TREES		
53	PIC-A	PICEA ABIES
Grand Total: 53		



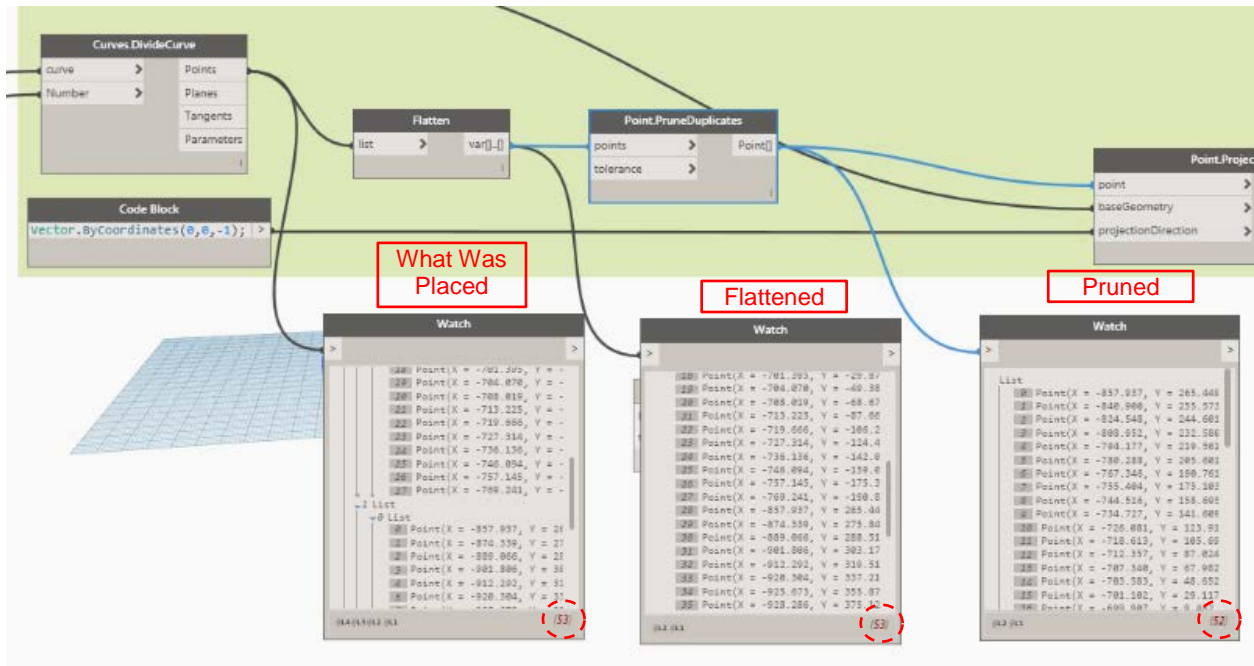
53????



This is easy enough to fix using the Prune Duplicates node. The watch nodes in the image below show the information coming from our nodes. You can see the points coming from divide curve are two separate lists. This is one list for each line that we selected. I'm using the flatten node to convert this to a single list. Notice the bottom right corner of the nodes both say {53} meaning there are 53 items contained in the list. When we pass the flattened list through the

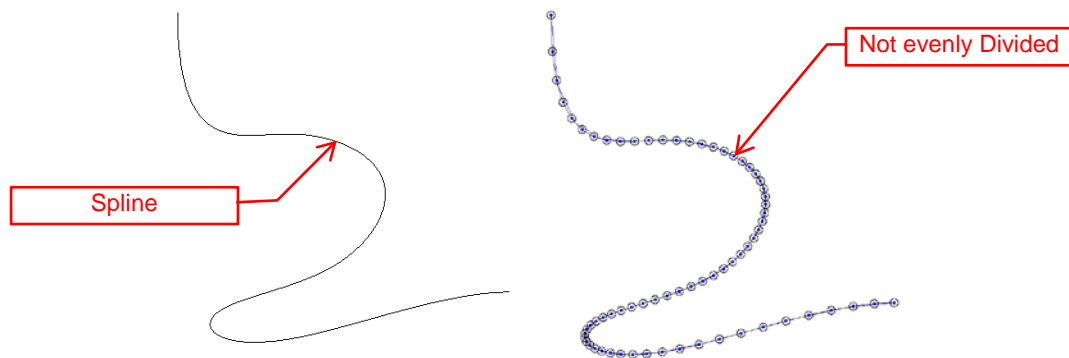


Point.PruneDuplicates node, the watch node now says {52}. Run the pruned list through the place families at a point node and you should have more accurate results.



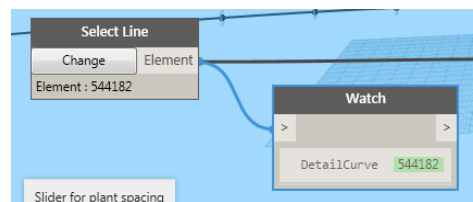
### Splines:

If we run the same workspace on the spline shown below, we do not get the desired spacing results.



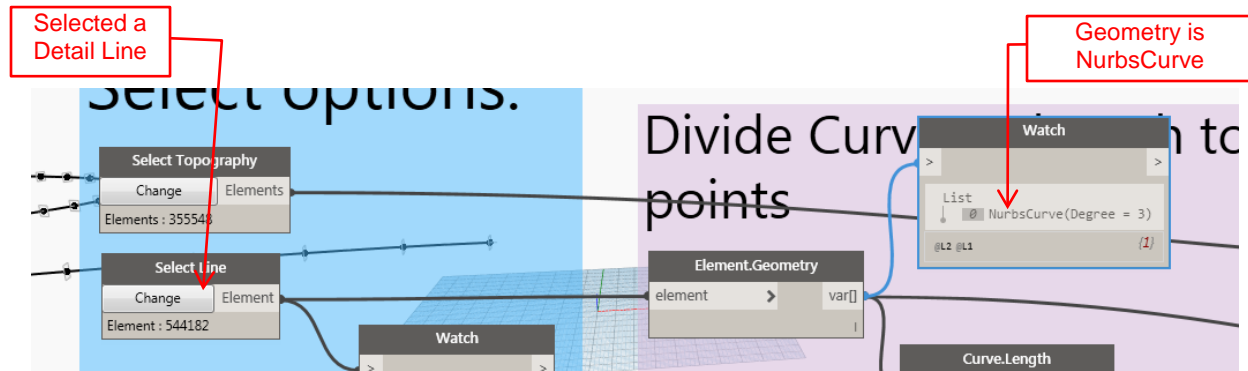
Whenever I try something new with Dynamo, I use a few nodes to see what I am working with. The Watch node, Parameters, and Geometry give me a lot of good information for what I need to do (and usually inspires a few ideas).

The Watch Node will show whatever is coming out of a node. You can select the node and usually see this information but when I'm trouble shooting, the watch node lets me quickly see what information is coming out of a node.



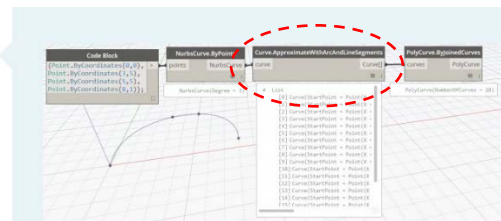
The Geometry node will show the dynamo geometry version of whatever you have selected. This is important for finding if you need to convert your geometry to be compatible with the input

of a node you are trying to use. When I do this to the selected spline, the watch node coming off or element geometry tells me the spline is viewed as a NurbsCurve.



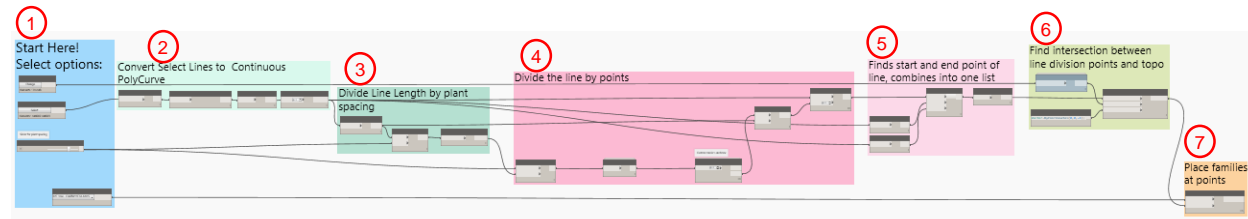
If you do a web search for Dynamo BIM Convert Nurbs Curve to Polycurve the Dynamo BIM Forum has an answer.

<http://dynamobim.org/forums/topic/convert-nurbscurve-to-polycurve/>



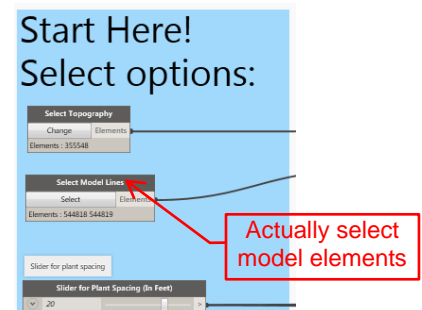
SCREENSHOT FROM DYNAMO BIM

We've already used the divide curve node, so here's a different method of splitting this line that will cover a single line, spline, or multiple lines.



### 1) Start here

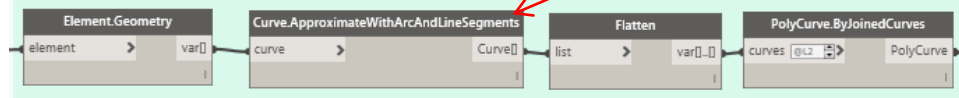
I've covered this one in the previous sections but I did want to point out something on this. You may have noticed that one of the selections says Select Model Lines. You can double click on a node and rename it within the current workspace. This adds some notes and helps your users understand what you want them to do. This is actually the select model elements node but I renamed it to tell someone using this on a project to select a line.



### 2) Convert lines to continuous polycurve

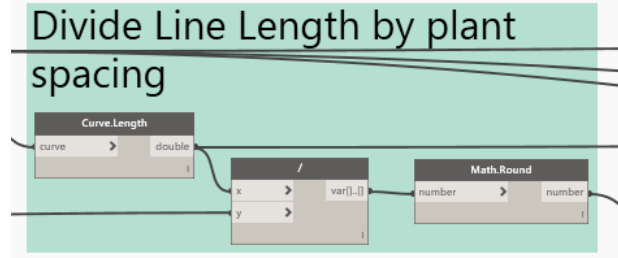
This group, converts the element geometry to a format that I can use by converting the spline to line segments and converting those segments to a continuous Dynamo line. The reason for the continuous line is to avoid needing to prune duplicates like we did when we selected the two lines.

## Convert Select Lines to Continuous PolyCurve



### 3) Divide line length by plant spacing

This group will take the length of the curve and divide that by the plant spacing number. This will tell me how many division points are along the curve.



### 4) Divide the line by points

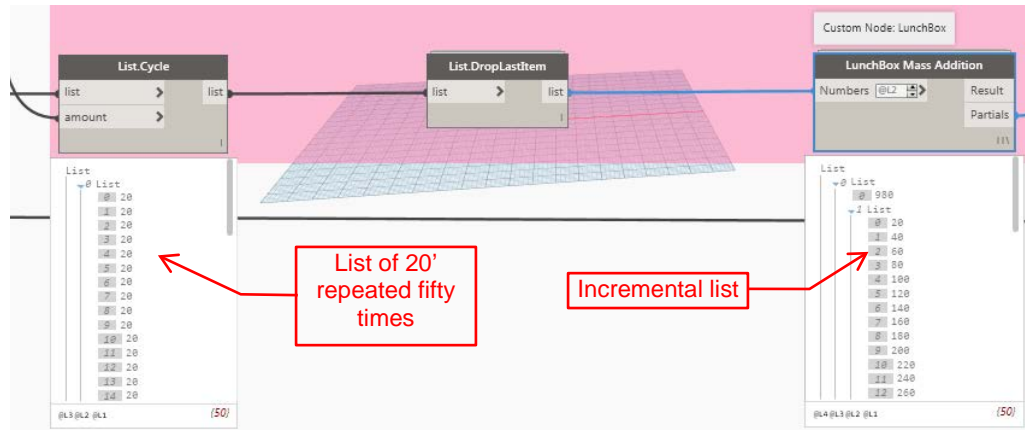
The Curve.PointAtParameter node looks at the entire curve as a whole so a value of one is the end and 0 is the start of the curve. If you enter .5, that will place a point at the mid-point of the line.

As an example, let's say a line is 100' long for easy math and we want to divide it at intervals of 10'. The first segment point at 10' is 10% of the overall line length so using the Curve.PointAtParameter node, if we enter value of .1, Dynamo will place a point 10' from the start of the curve.

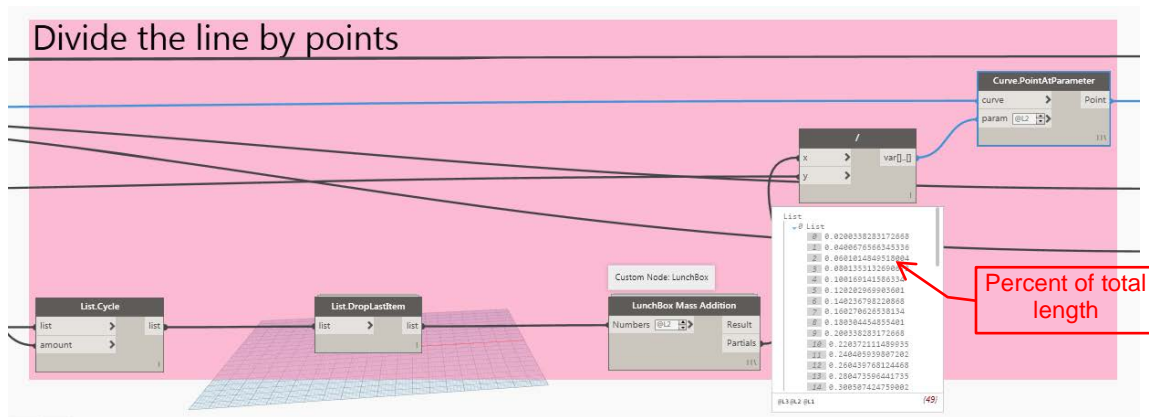


I take the number of points needed to equally divide the line based on our plant spacing and determining what percent of the overall line length each point is at. To do this I need the spacing, and number of points. The list cycle node will make a list of the spacing distance repeated by the number of division points needed.

Going back to the example of the 100' line split at 10' increments, if I divided the overall length by the plant spacing ( $100'/10'=10$  **points**). Using the list cycle node, I would make a list of 10 items each item being 10'. The lunchbox Mass addition node will make a list adding each segment to the previous one. Below, I have expanded the results we have a 50 item list of 20' the LunchBox Mass Addition node shows a list of 20, 20+20=40, 40+20=60...



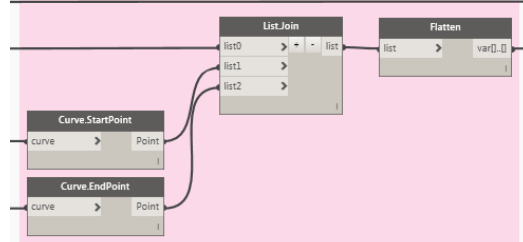
The mass addition results represent the distance along the curve that each point would sit. If we divide the overall length of the curve by each point in our list, we get the percent of the overall curve to input in the Curve.PointAtParameter Node.



### 5) Find start and end point combine to make overall list

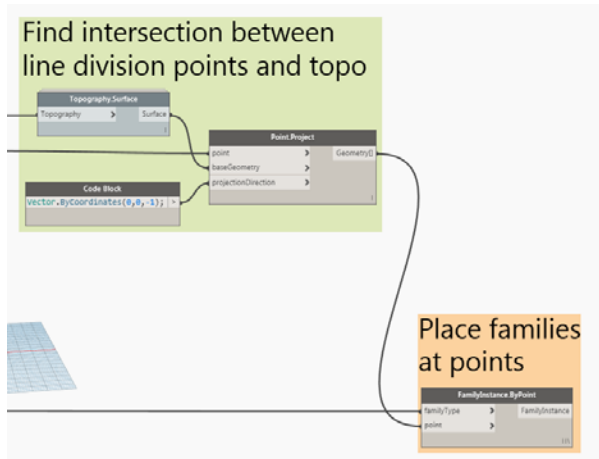
This will take the start and end point of the curve, combine them into a list using the list join tool. I used the flatten node to make a full list of each point that we want to place a family.

### Finds start and end point of line, combines into one list

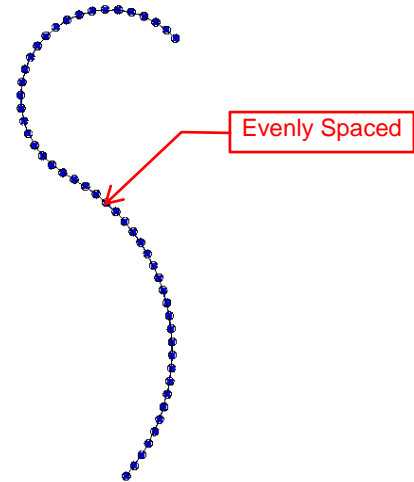


### 6&7) Find Intersection of points with Topography Then place families

Now we have evenly spaced lines along the spline.



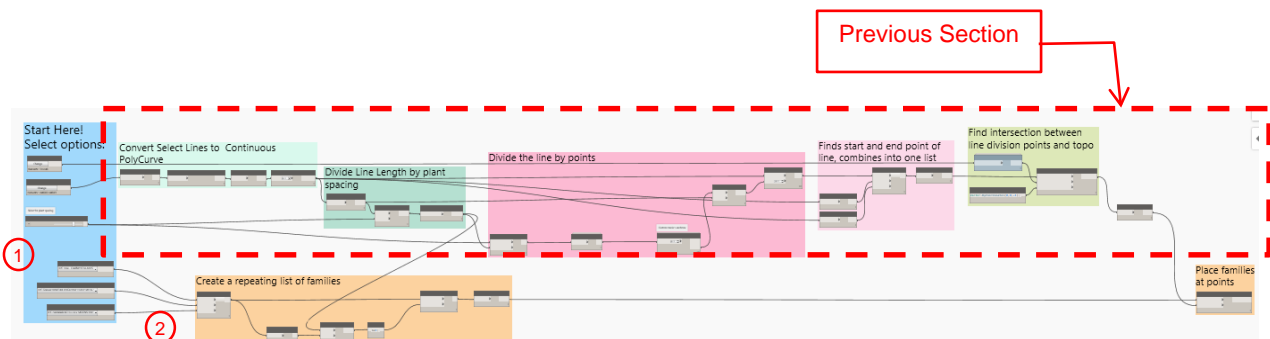
IN DYNAMO



IN REVIT

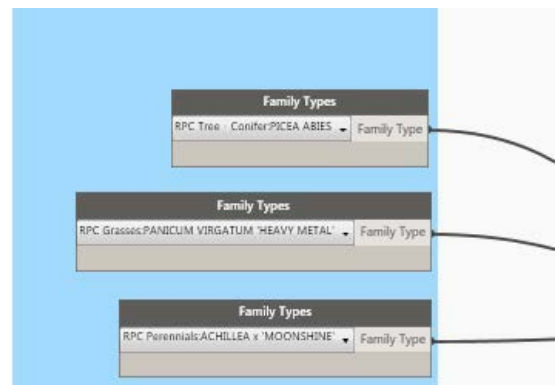
## Planting Patterns:

A common design for landscape architects is to place patterns of plants. Here is a way we can place three different family types along our path.



### 1) Start Here!

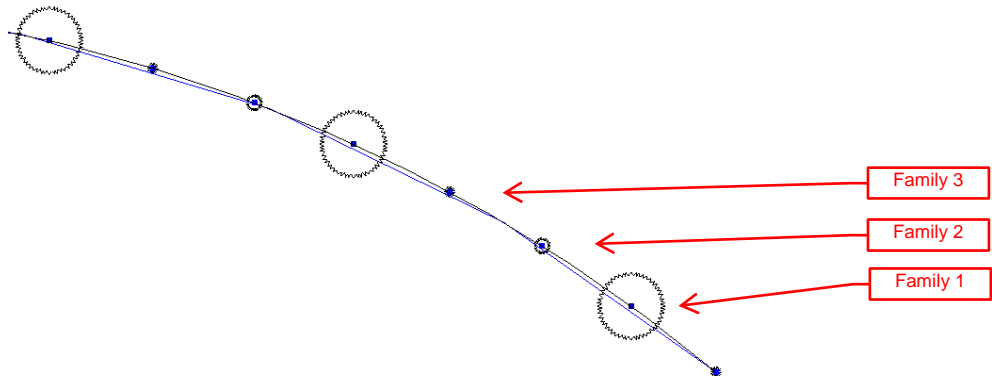
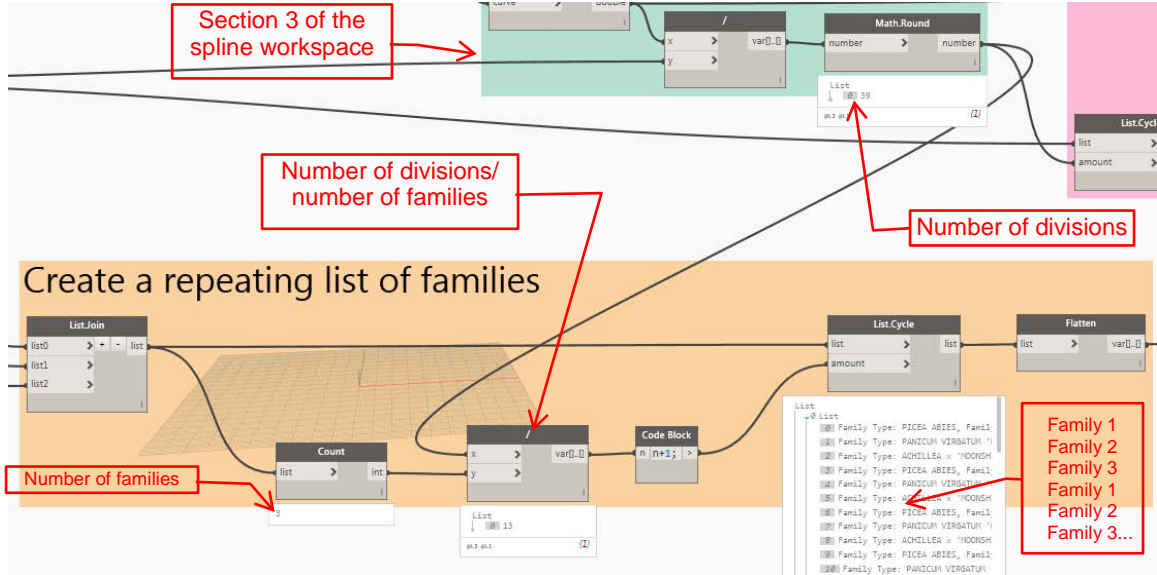
Instead of selecting one family, use three family type nodes to pick three different families.



### 2) Create a repeating list of families

What this group does is combine the selected families into a repeating list of families to be placed. Section 3 of the Spline workspace found the number of divisions required to place families evenly across the length of a selected line. This group will take the number of divisions and divide it by the number of selected families to find the number of times the list of families should be repeated. As an example I am repeating 3 families. If the line requires 15 divisions to evenly place a family at 5' spacing, I would need a list of 15 families to place. To keep the same pattern, I would repeat my list of 3 families 5 times.

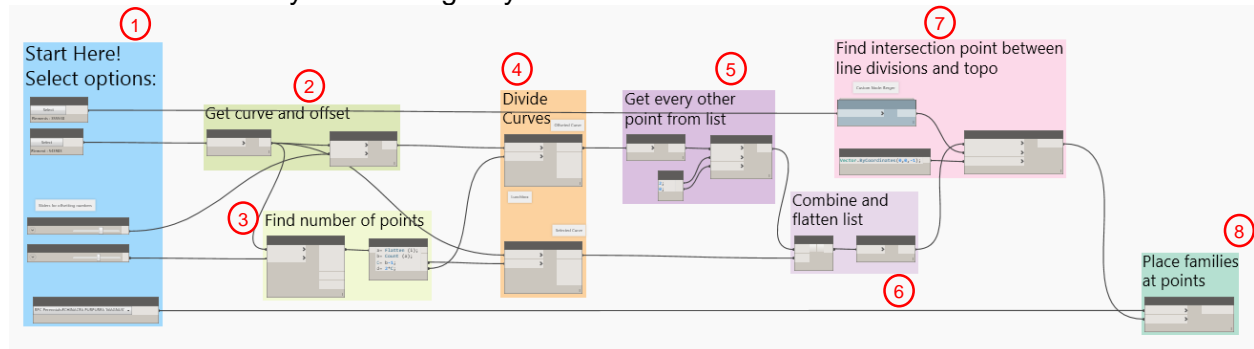
The list combine node will combine the selected families into a single list. The list count node will count how many items are in that list. The divide node is dividing the number of divisions of the overall line by the number of selected families. This provides the number of times the list of families will be repeated to create the list of families to be placed. The List.Cycle node will repeat the list of families by the number that we produced from the divide node.



*IN REVIT*

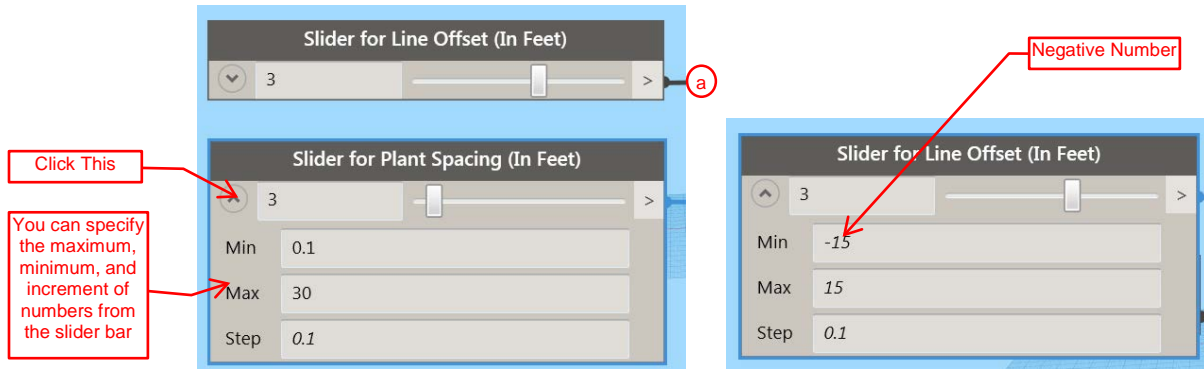
### Creating rows:

Landscape designers frequently place plants equally spaced along a path in rows. You could draw two lines but why not leverage Dynamo for this.



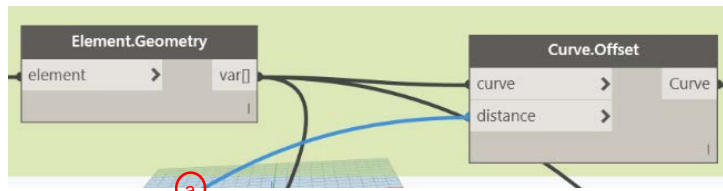
### 1) Start Here!

I have added another slider to the selection group for the offset distance of the line. As a bit of extra information, you can adjust the numbers the slider can output. For the offset, I allowed the slider to have a positive or negative number so the end user can offset either direction from the selected line.



### 2) Get curve and offset

This group uses the offset node to offset the selected line.



### 3) Find number of points

In the previous examples, I used the curve length node and the divide node to find the number of points for placing plants. There are a lot of different ways to skin a cat. Rather than divide the curve length, I am using the curve divide by distance node to find the number of points in the list telling me how many points to divide the line by.

I am also using a code block to reduce the number of needed nodes. Code blocks can save space and time while making your workspaces. The code block shown below is flattening the list of points then counting how many items are in that list. All of these start with a definition of a letter so that I can use that value later in my code block. Breaking it down, (i) is the input value and we are saying a equals the flattened list of (i) the input. Then b equals the count of (a) our flattened list. Then c equals the count of the flattened list minus 1 and finally, d equals the count of the flattened list minus 1 times 2.

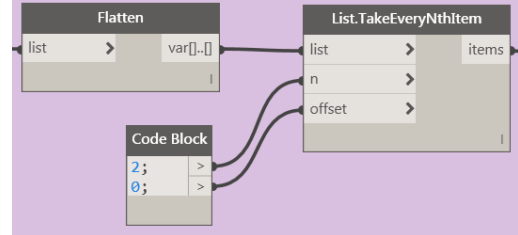
## Find number of points

PLANT SPACING IMAGE

#### 4) Divide Curves

In the image above the last formula is  $d=2*c$ .  $C$  is the number of points I am dividing the selected line by and  $d$  is the number of points I am dividing the offset line by. I am using this to have an alternating pattern similar to the plant spacing image.

### Get every other point from list



#### 5) Get every other point from list

As mentioned above, the offset line is divided twice as much as the selected line. Using the list.TakeEveryNthItem node, we can take every other point from the list. This will find the point between the points of the selected line giving the desired planting pattern. The red dots shown in the diagram below indicate the locations we are collecting for where to place plant families.

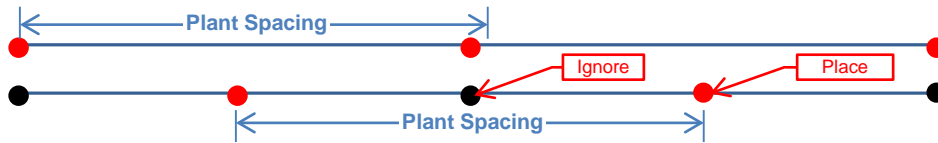
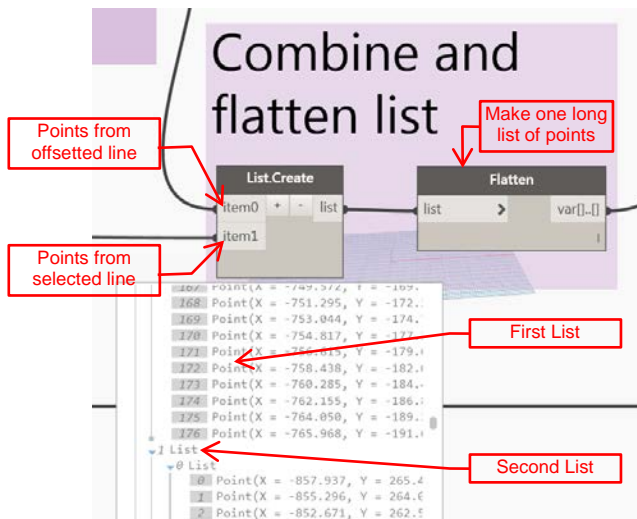


DIAGRAM FOR POINTS TO PLACE PLANTS

#### 6) Combine and flatten list

The list join node concatenates the incoming lists into a combined list. This will make a list of lists. To keep this information easier to work with. I'm using the flatten node to make it one long list.



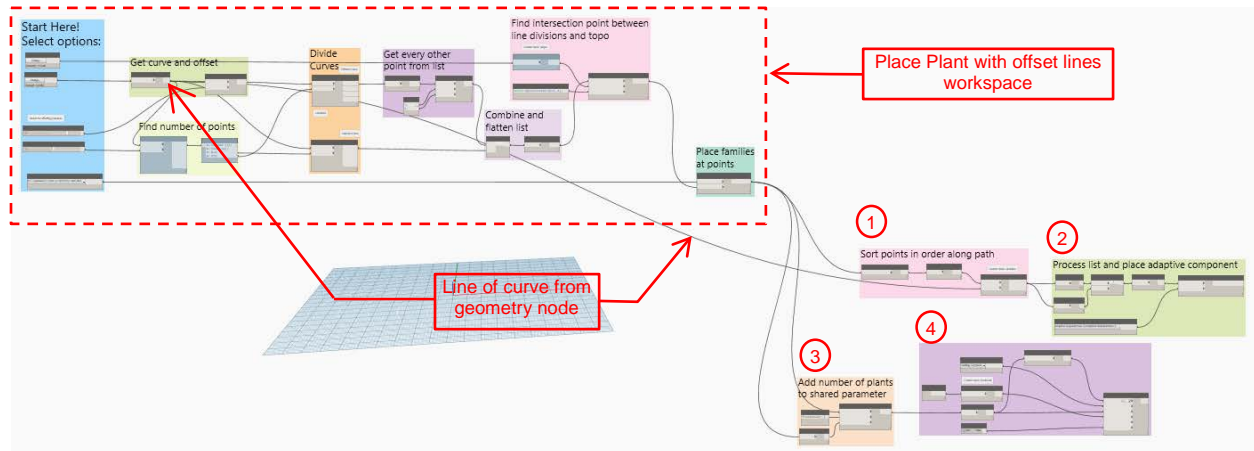
### 7&8) Find intersection of Points and place the family type at points

We've seen this one a few times... This workspace is an example of where switching Dynamo to being Automatic mode is really handy. Using the two sliders for plant spacing and offset, a designer can fine tune where the plants get placed quickly and easily by adjusting how far apart the rows are and plants paced along those rows.



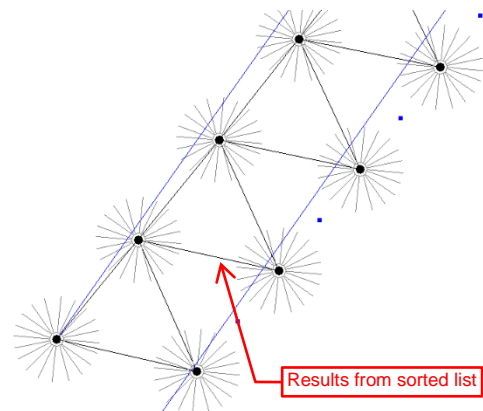
## Annotating:

Earlier in this document I showed a method of selecting the elements in a view and annotating the plants. A lot of the work done in that workspace was to sort by family type and group runs of plants together. In these last few workspaces, we are creating the runs of plants so there is a lot less work required to annotate the drawings. The plants are placed in the order I would like to annotate them. Expanding on the offset workspace, the image below shows the additional nodes needed for placing a line based adaptive component and tagging a plant.

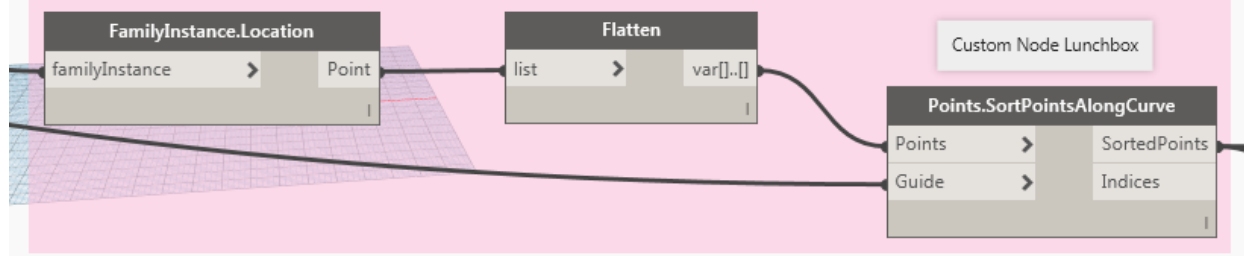


### 1) Sort Points in order along path

Dynamo processes information it is given in the order it is received. The previous workspace combined a list of points from the selected offset lines that we used to place families along. This information is not in the order we would like. To sort a list of points to place the adaptive component along use the `Point.SortPointsAlongCurve` node from the Clockwork package. This node will sort points along a curve. Using the selected curve, this node will sort the points in an order that we would use to document our drawings.

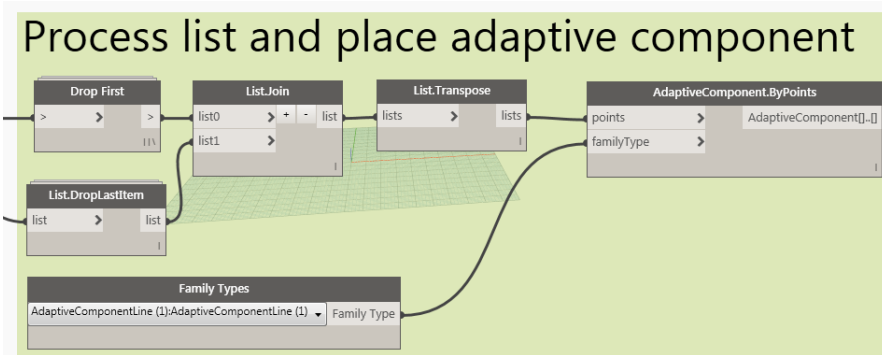


## Sort points in order along path



## 2) Process list and place adaptive component

This should look familiar



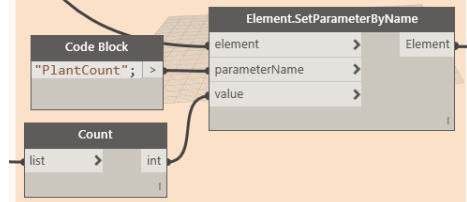
## 3) Add plant count

This should look familiar as well. This group will find the count from our list of plants and adding this to a shared parameter.

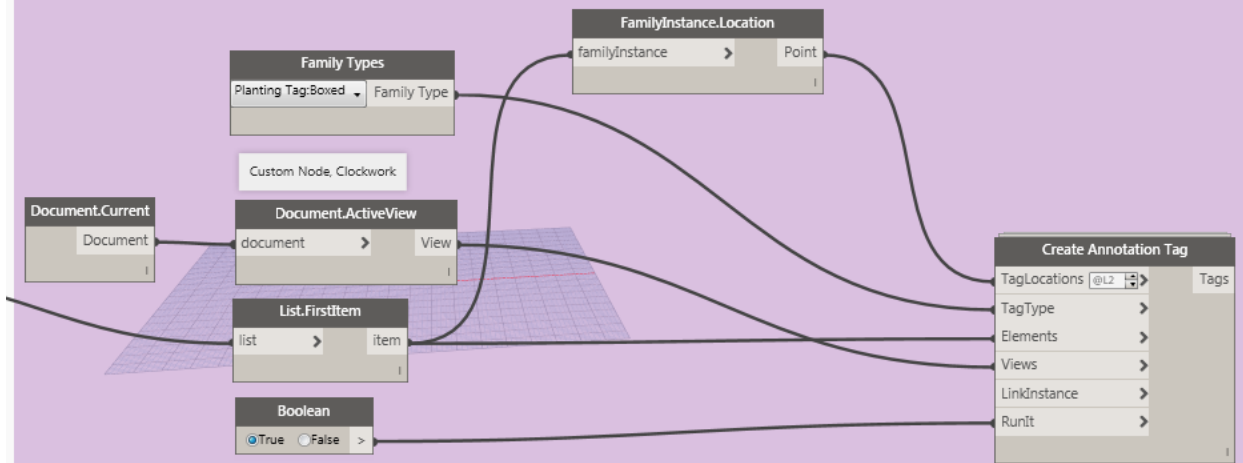
## 4) Tag Family

In this group, I'm taking the first item from the list of plants and tagging it.

## Add number of plants to shared parameter



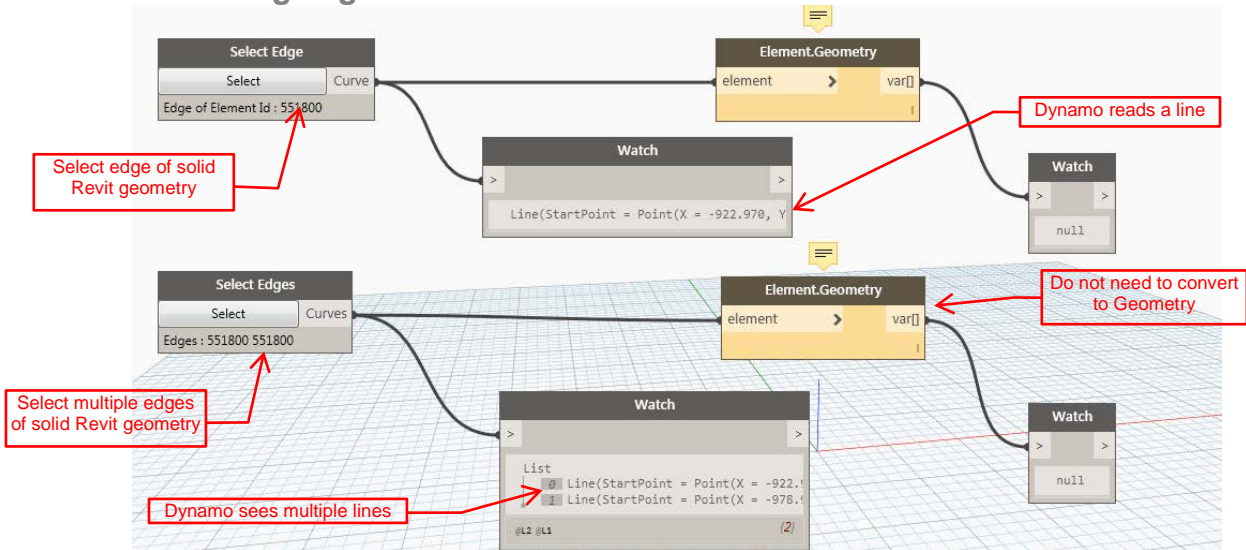
## Tag one item in list



## Working with Revit geometry:

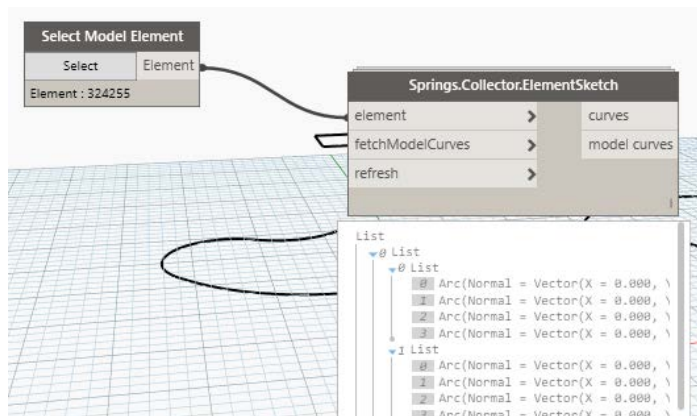
The previous workspaces utilize dynamo curves to place families. Revit Geometry can convert to curves in a lot of ways. This allows us to create some great workflows for placing families. Using floors and filled regions to model and annotate our drawings, we can the select edges (lines) we want to work with rather than draw a line. In Revit we can select the edge of a floor slab using the select edge node. Dynamo sees the edge as a line.

### Selecting edges:



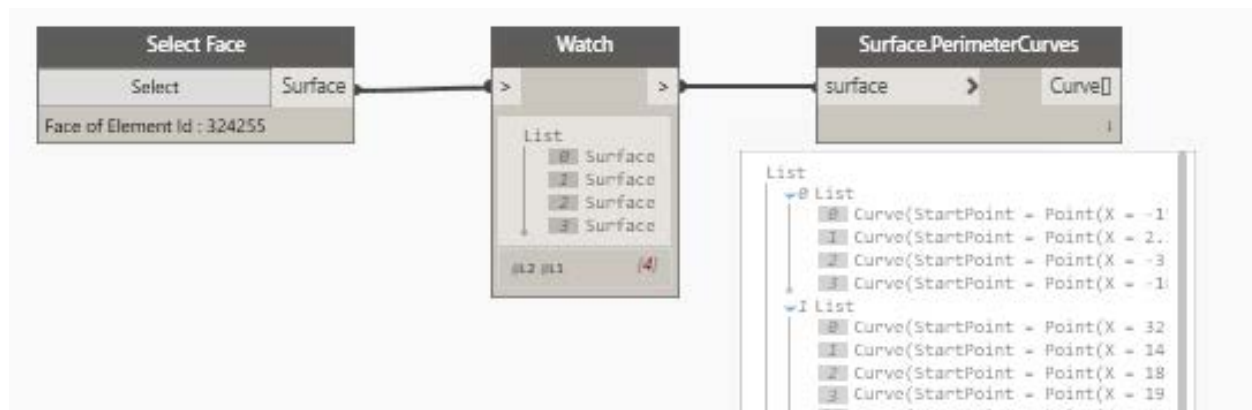
### Collect Sketches:

There is a custom node in the springs package called Springs.Collector.Element.Sketch. What this does is finds the Revit sketch line work of Revit Elements. This will work for Floors, Ceilings, Filled Regions, and even site plan sub-regions. This powerful little node will let you quickly find the lines of just about anything that is created in Revit that enters sketch mode.



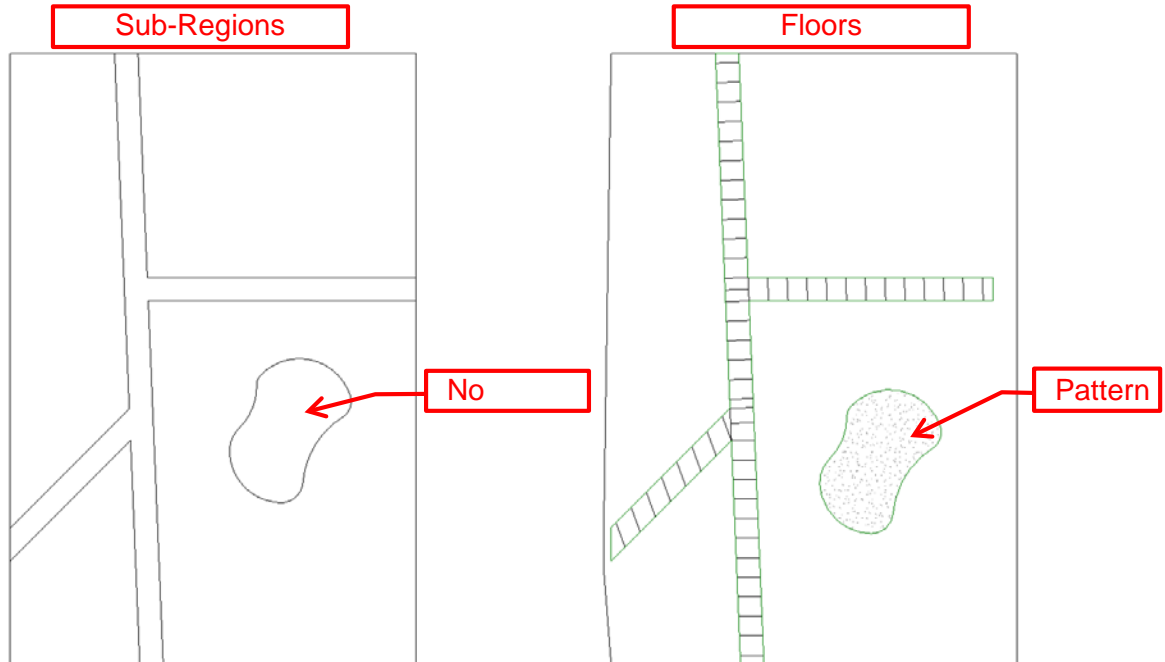
### Select Face

The select face node allows a Dynamo user to select a surface area of a Revit element like a floor or a masking region. The Surface.PerimeterCurves node extracts the curves of the perimeter of the surface. There are many other methods of getting curves. The point is that you can use Dynamo to select a type of geometry and convert that to curves and you can use any of the previously shown methods of placing families along those paths to start leveraging the Revit model to assist with your landscape design.



## Planter Beds (Combining it all together):

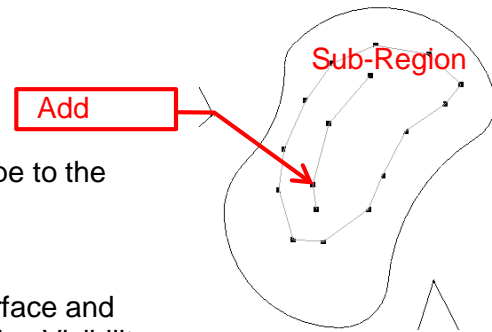
In Revit, planter beds can be modeled similarly to hardscape using topography sub-regions or floors. The benefit of using a floor is that the floor will show layers in section. This is helpful with planter beds if you need a certain depth of dirt and is why this is my preferred. Topography sub-regions are quick and easy and will simplify the process of following the site grade. If you do not intend to use Dynamo in your workflow sub-regions may be the better method for you to use.



### Adding slope:

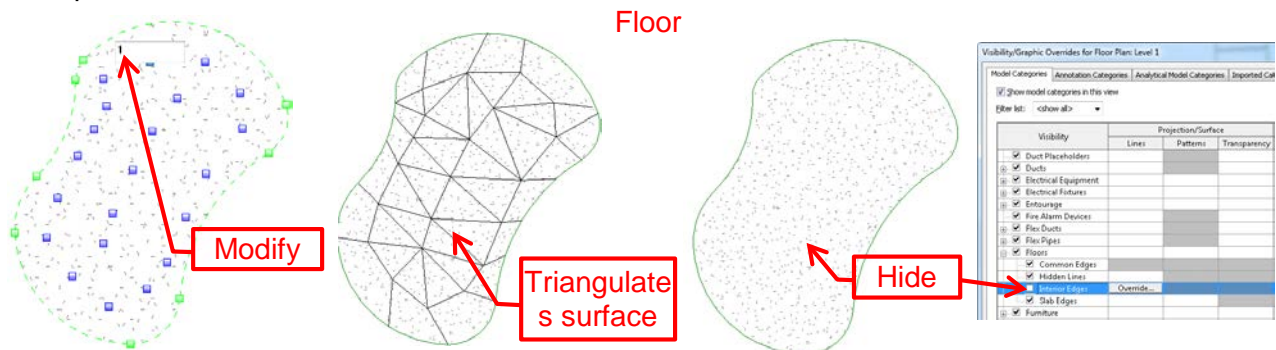
#### Sub Regions

Edit the topography and add elevation points to add slope to the planter bed if you are using Sub-Regions.

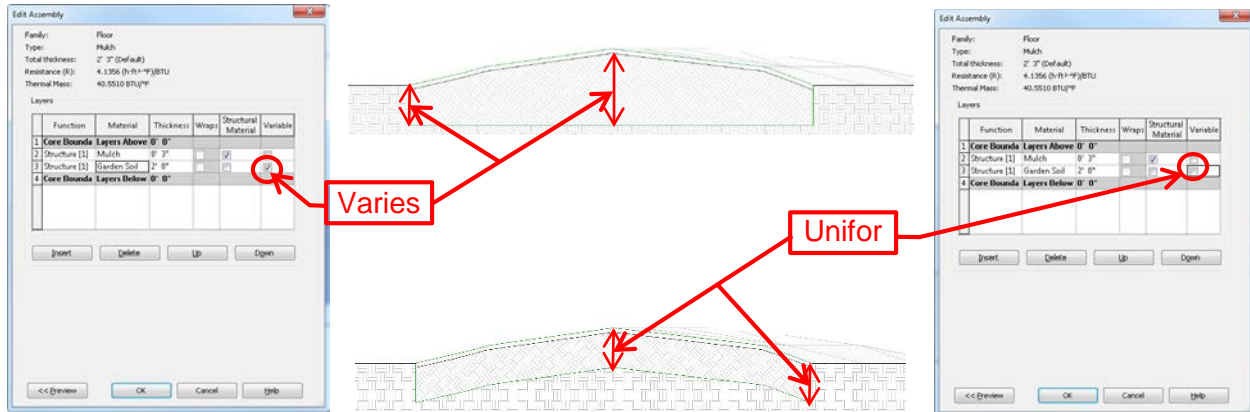


#### Floors

Use the shape editing tools to add points to the floor surface and Modify Sub Elements to change the points elevations. Use Visibility Graphics to hide surface lines

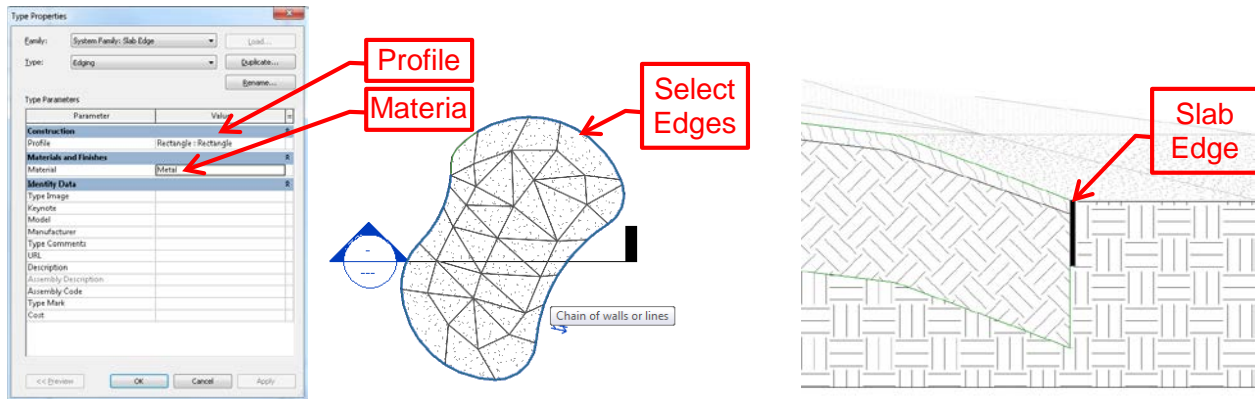
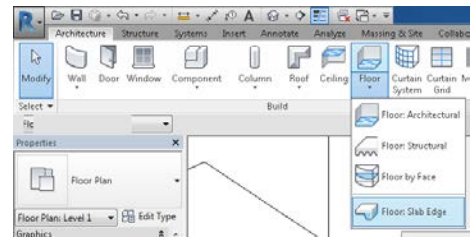


**In section:**



**Edging:**

When using floors to make planter beds, edging is easy to model using a slab edge. Make a profile that matches the cross section of your edging. Make a new slab edge type, select the profile and assign the desired material then pick the edges of the planter bed.

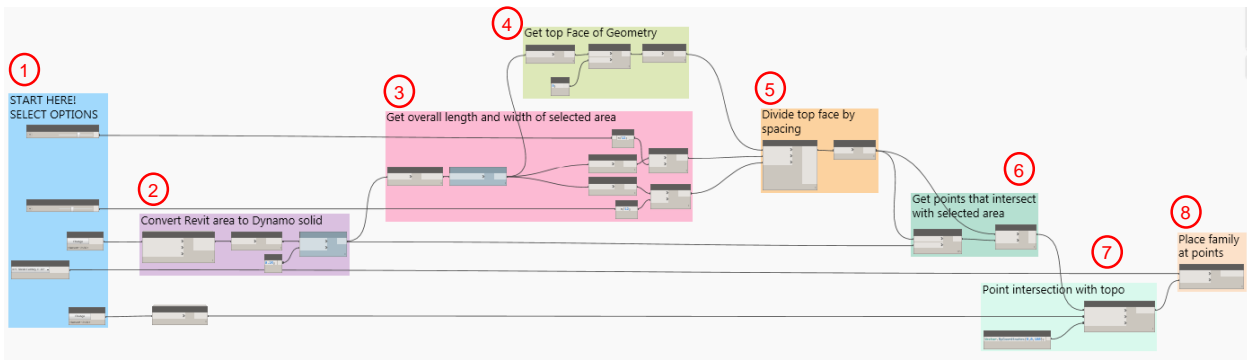


**Placing plants:**

This section is going to cover a few different methods of placing plants using dynamo. This is going to build on some of what has already been covered in the segment on placing trees. If information is covered in another section, I will note that and save a few trees but not re-writing the same information.

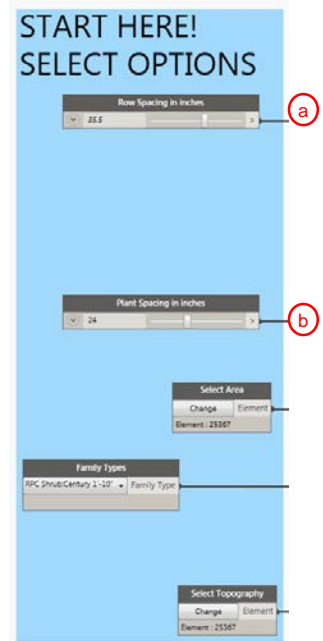
**Evenly Spaced Plants in a region:**

This workspace will take a selected region, evenly place points across the face of the region, and place families at those points.



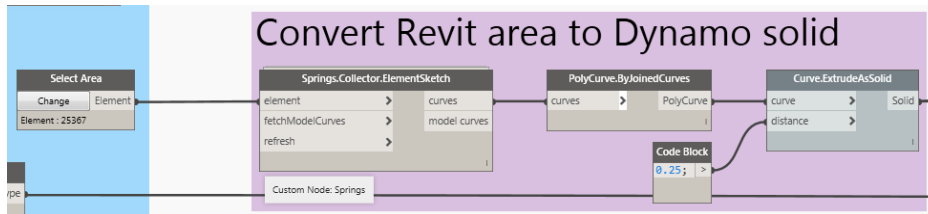
### 1) Start Here!

This group is not much different than the previously shown starting points. There are two sliders renamed Row Spacing in inches and Plant Spacing in Inches. I'll show an alternate to this in the next workspace.



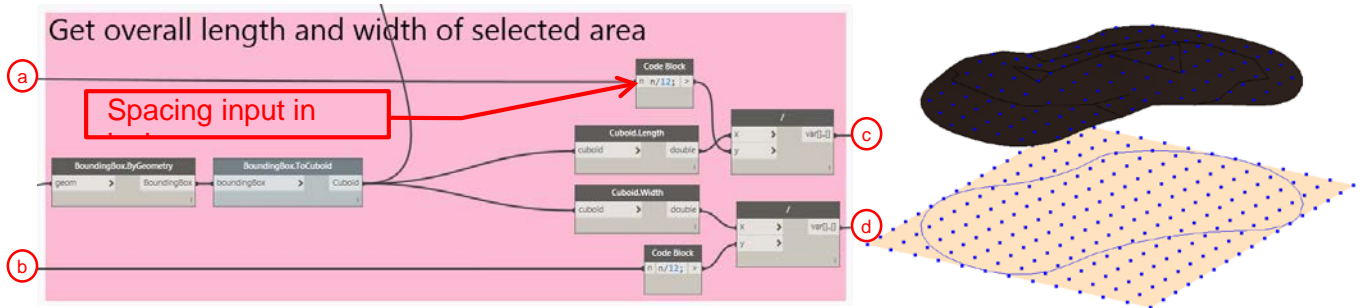
### 2) Convert Revit area to Dynamo Solid

This group is setup to work if you select a filled region, floor, or sub-region. The springs.Collector.ElementSketch will get the curves of the selected region. The polyCurve.ByJoinedCurves node combines all the curves of the sketch into a continuous loop. The Curve.ExtrudeAsSolid node will extrude the loop into a Dynamo Solid that we can use later.



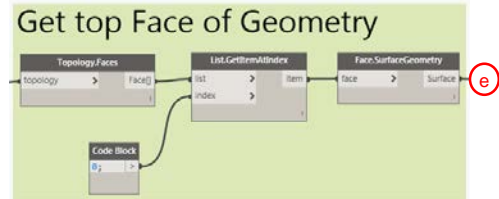
### 3) Get overall length and width of selected area

This uses the bounding box node which will create a cube around the selected geometry. This cube will be used to tell us the overall length and width of the region to determine how many divisions are needed to evenly space the specified family.



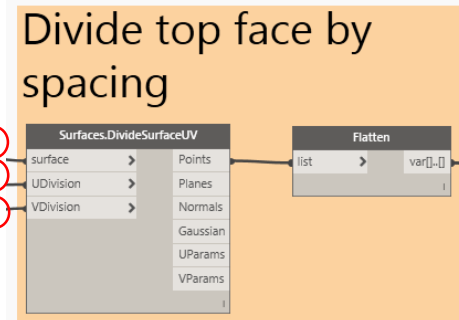
**4) Get top face of geometry**

This group converts the bounding box cube into its underlying surfaces. Because a cube has 6 surfaces and we only want to work with the top, the List.GetItemAtIndex node was used to get the top face.



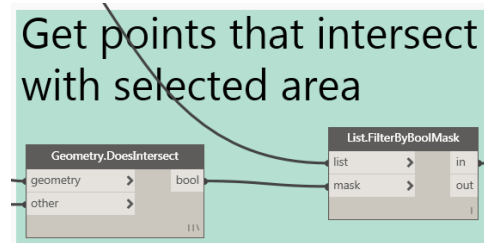
**5) Divide top face by spacing**

This group uses the Surfaces.Divide.Surface node to split the top surface of the bounding box from the previous group.



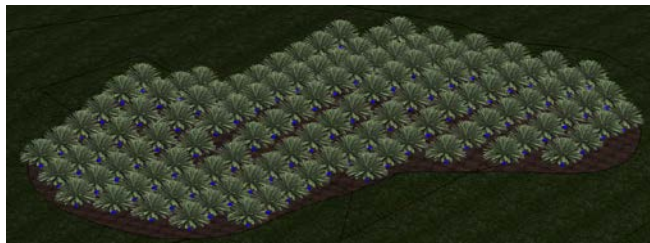
**6) Get Points inside of selected area**

The previous groups were used to create a uniform grid of points based on our spacing inputs. This group will take the extruded sketch lines of our area from group 2 and find which points from group 5 intersect with the solid. This gives us a list of the points where we want to place a family



**7&8) Project to topography and place families**

The last two nodes we've looked at a few times. If you need to revisit this go back to the previous sections.



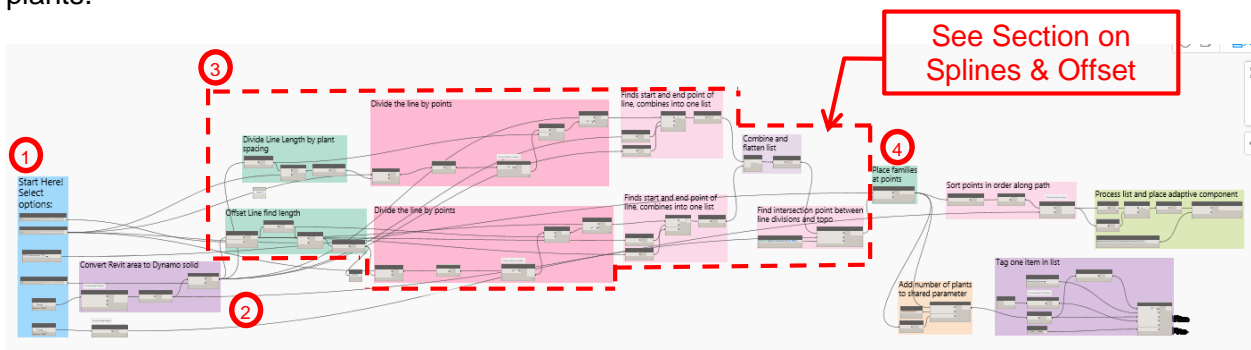
30" SPACING



40" X 60" SPACING

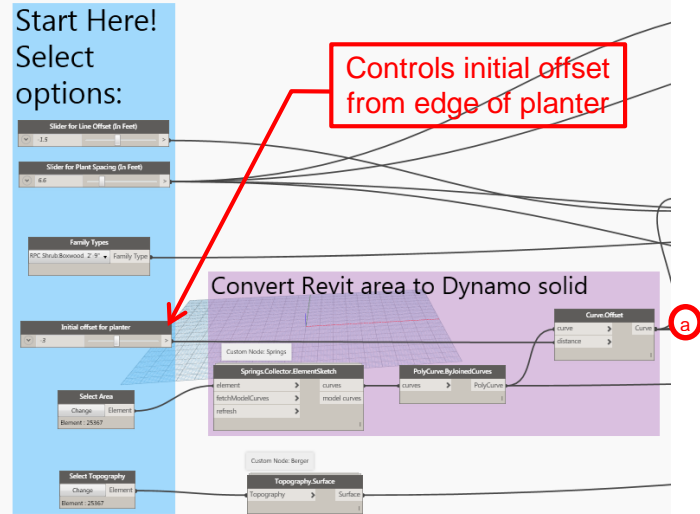
**Combining lines and regions:**

A common planter design is a row or two of evenly spaced plants around the edge of the planter with the remaining planter filled with evenly spaced plants. This next workspace combines the spline technique and working with offsets. This workspace will have the user select a region in Revit, get the sketch lines of that region and use those to create the offset lines for placing plants.



**1) Start Here**

I've added an additional slider to control the initial offset of the line from the outside edge of the region we select.

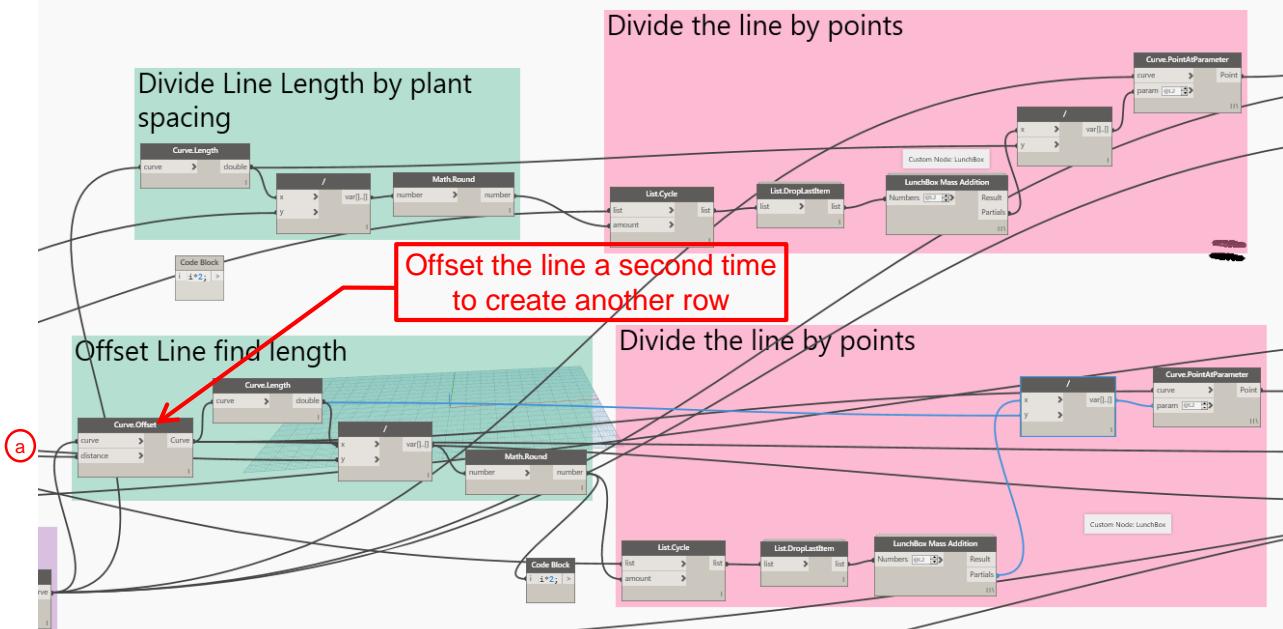


**2) Convert Revit area to Dynamo Solid**

This group is similar to placing plants by filled region. Springs.CollectorElementSketch collects the sketch line of the selected area. The Curve, offset node offsets the line away from the planter bed perimeter. A negative number is offsetting inward.

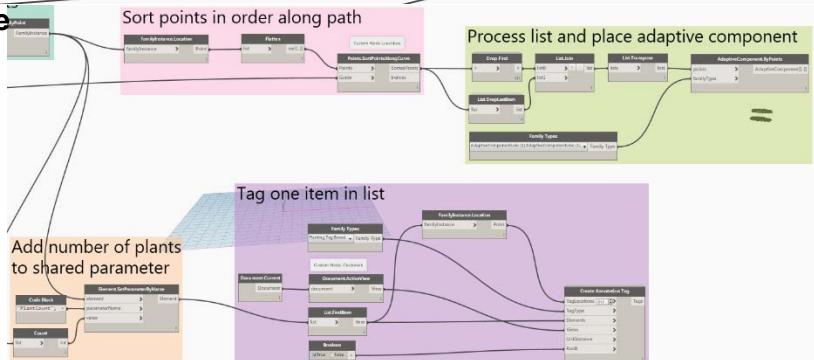
**3) Splines and Offset**

I am showing this section again only to show the flow of information. (a) in the image above is being used like the selected line in the offset workspace.



**4) Place Families and annotate**

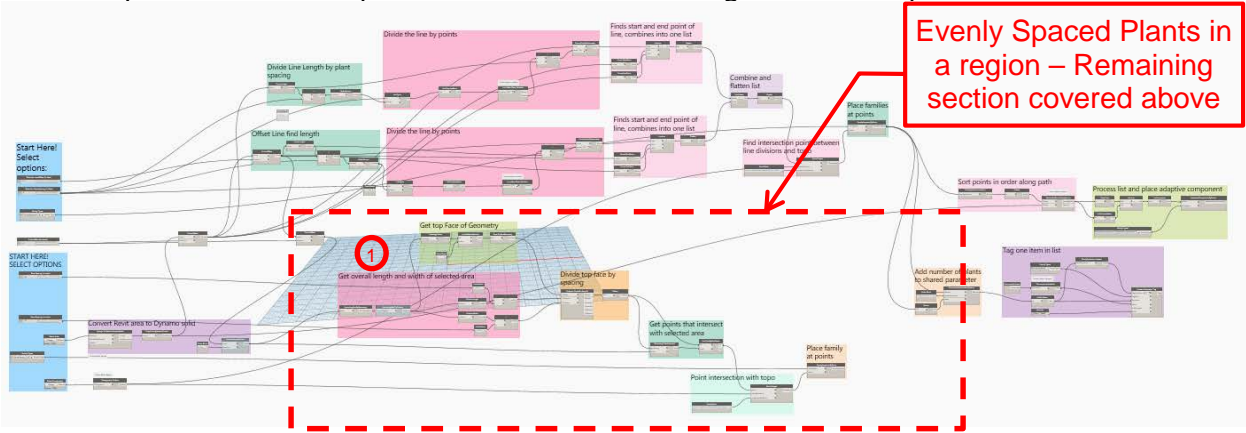
Nothing new here:





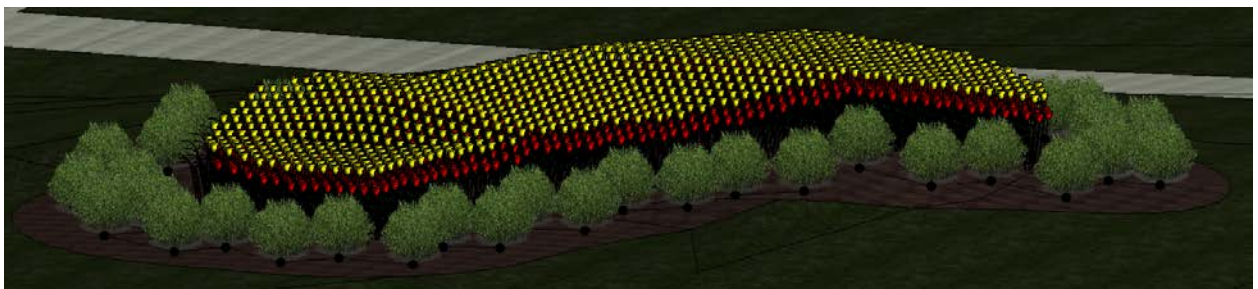
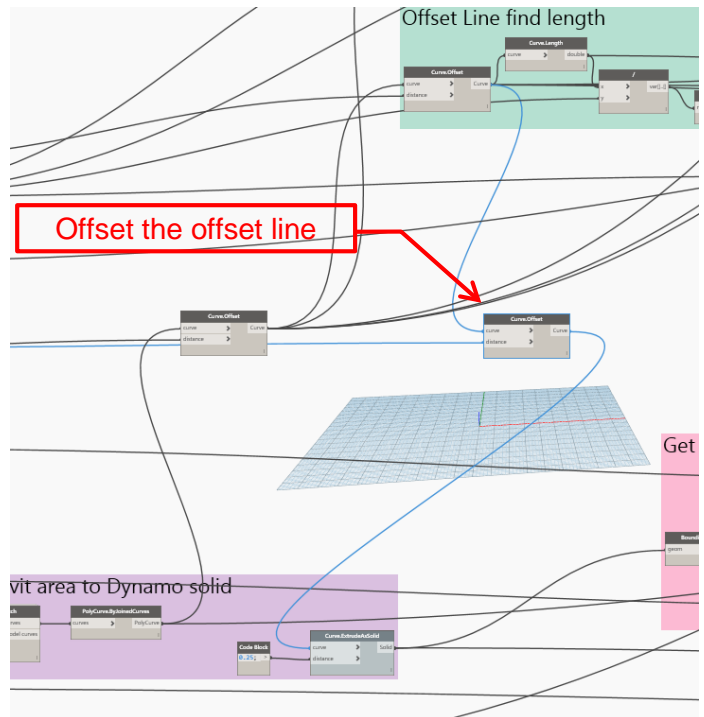
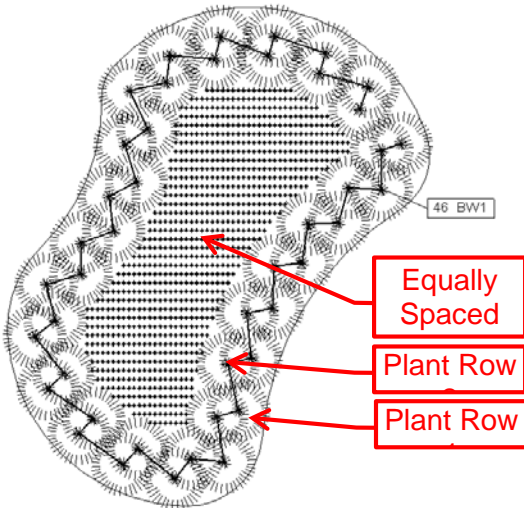
**Taking it further:**

If we combine this workspace with the plants by region and use the line of the second row of plants to create our region to fill with plants, we can have a workspace that if we select an area, Revit will place two rows of plants and infill the remaining section with plants.



**1) Offsetting the offset lines**

Feed the curve of the second plant line into the curve offset node and use this to create the solid geometry to intersect the list of plant points with and you'll be able to find the interior section of the planter.





## **Conclusion:**

Revit's database and combination of 3D and documentation ability make it a good software platform for landscape architecture. The addition of Dynamo into your workflow can make Revit a great software platform for landscape architecture. We've looked at a lot of different methods of using Revit for landscape design and documentation.

I do not want to discourage anyone from jumping in and using Revit for their landscape designs. Here is my advice. Start small. Placing trees and scheduling the counts alone will add value to your project. Don't shoot for the purist point of view in using Revit. Get there gradually. Revit has drafting and modeling tools. This is the business of landscape architecture and we need to make a profit on the jobs we produce so take each step incrementally. Start by linking your CAD details to Revit, get used to the program interface and build from there.

I have uploaded all of the workspaces I will be showing in this presentation and the custom nodes. They work with Revit 2017 and Dynamo version 1.2 at this point in time. Thank you for attending the class or reading this document. I hope that you learned something from it.

**Thank you for attending! And please fill out the speaker evaluation.**