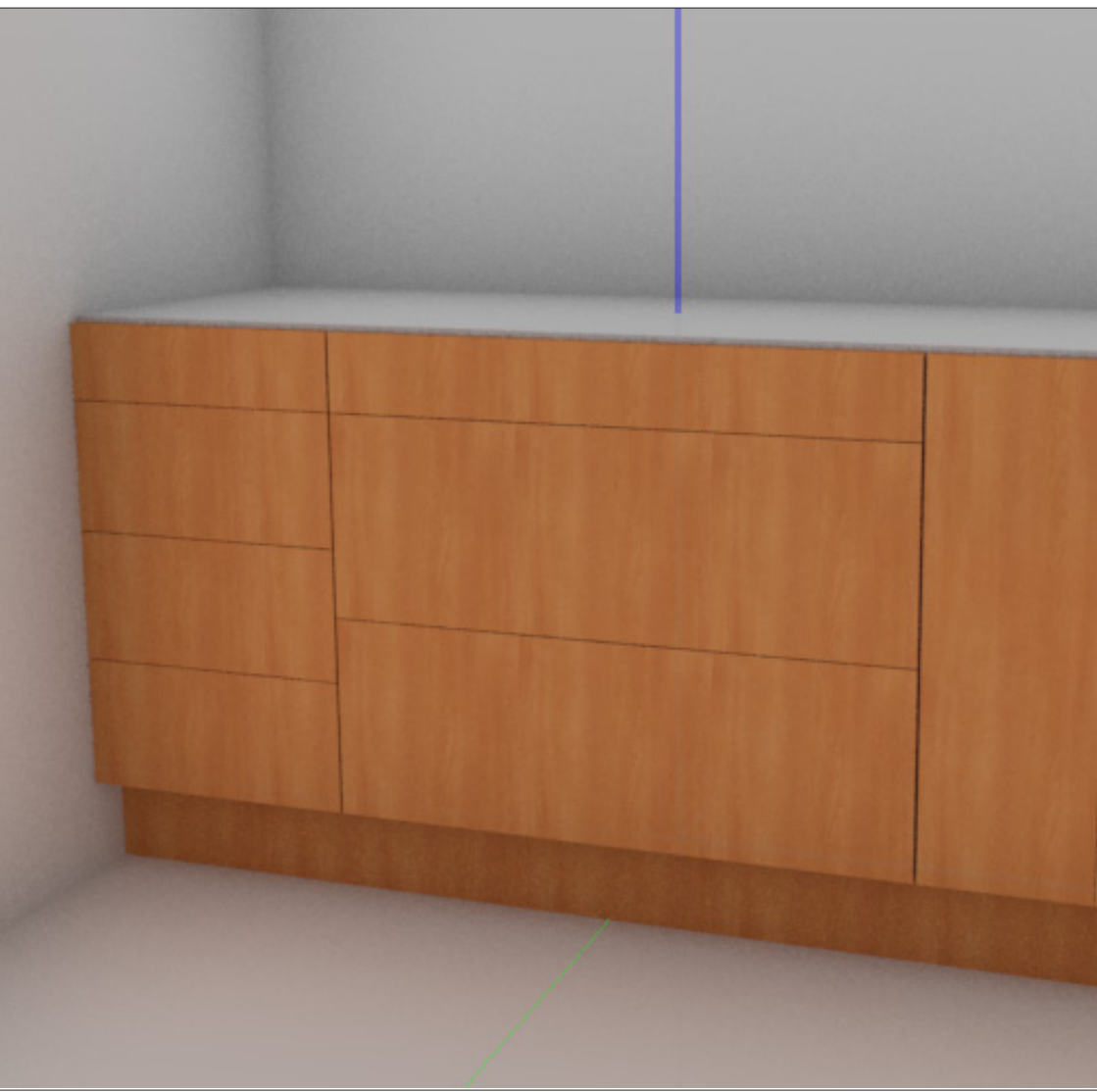
SHORT SHARP MANUALS

1511 Marionette



archoncad.com
Making Vectorworks easy!
http://learn.archoncad.com

For more Vectorworks training information, or to purchase more copies of this book, please email jon@archoncad.co.nz

Contents

Introduction

At the Vectorworks Design Summit (April 2015), they announced that Vectorworks 2016 would include a new visual scripting language. This new scripting language is called Marionette. This is what Wikipedia says about visual scripting languages in general:

In computing, a visual programming language (VPL) is any programming language that lets users create programs by manipulating program elements graphically rather than by specifying them textually. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols, used either as elements of syntax or secondary notation. For example, many VPLs (known as dataflow or diagrammatic programming)[1] are based on the idea of "boxes and arrows", where boxes or other screen objects are treated as entities, connected by arrows, lines or arcs which represent relations.

I have been watching a movie from the Vectorworks design Summit where Dr Biplab Sarkar introduces the audience to Marionette. He goes on to explain that marionette is a new design tool that will allow you to create easy and efficient scripts to develop your designs.

You could think of marionette as being like a recipe to bake a cake. Each step in baking a cake is usually given in the recipe. But unlike baking a cake marionette has to be instructed to do absolutely everything. For example, if you were to tell marionette to add eggs to the cake mix, you would also have to instruct marionette to break the eggs, otherwise you would end up with two eggs and their shells inside the bowl. Usually cake recipes do not tell you to break the eggs because they assume that you know this. Marionette does not know anything and has to be instructed on every little step along the way.

Basic Concepts

Marionette works by connecting nodes together with wires. The data flows from left to right, from one node to the next down the wire. Each node does something simple and then passes its output to the next node. To create a script, you drag and drop the nodes to create a series of steps, at the end of this you get the object that you want. It might look complex, but once it is set up you can look at different variations by editing any of the node inputs. Each node is a Python based script (allowing you to edit the script) and you can turn the marionette object into a plug an object if you want.

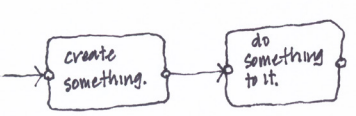
Some other CAD programs have had this kind of visual scripting language available for several years, and the feedback from the users is that this is a very powerful way to create your designs. It might look complicated to use this technology, but it does mean that you don't have to understand the programming language, you just have to understand what simple steps you want your objects to go through. You might say that it will allow complex programming for everybody.

Nodes and Wires

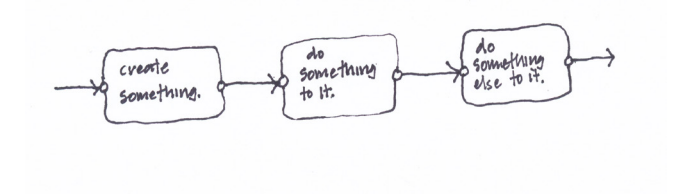
The first concept to understand is that an instruction in marionette is called a node.

- A node tells Marinette to do something

- After telling marionette to do something, you can tell it to do something else.



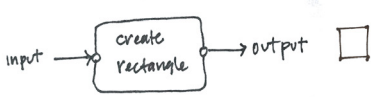
- The strategy is to set up a series of instructions (nodes) that create something, then does something with it, and then does something else to it.



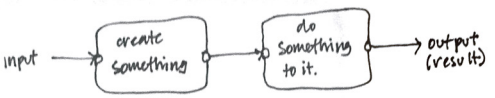
The next concept to understand is that the the series of nodes are connected together with wires. This series of nodes and wires is known as a marionette network. The network always runs from left to right.

- A node usually has an input on the left and an output on the right.

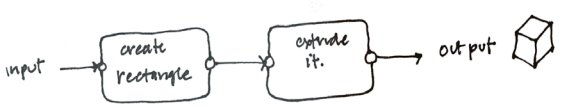
- In this example the instruction is to create a rectangle. We have to start with an input, which could be the size of the rectangle and its location. The the node creates the rectangle, which is the output.



- We could have an input, an instruction to create something followed by an instruction to do something else to it, which should result in an object.

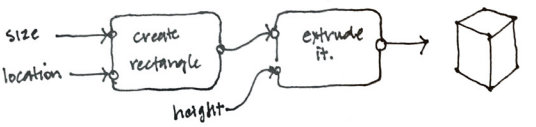


- In this example we have an import, then an instruction to create a rectangle followed by an instruction to extrude, which gives us an output of an extruded rectangle.



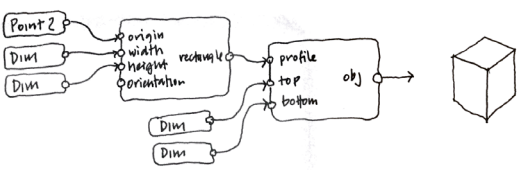
This image shows a simple series of nodes connected together to create an extruded rectangle.

- The node for creating a rectangle requires more than just a simple input. It requires the size and the location of the rectangle.
- The node for extruding the rectangle requires more than just the object, it also requires the height of the extrusion.



And here is the actual marionette network.

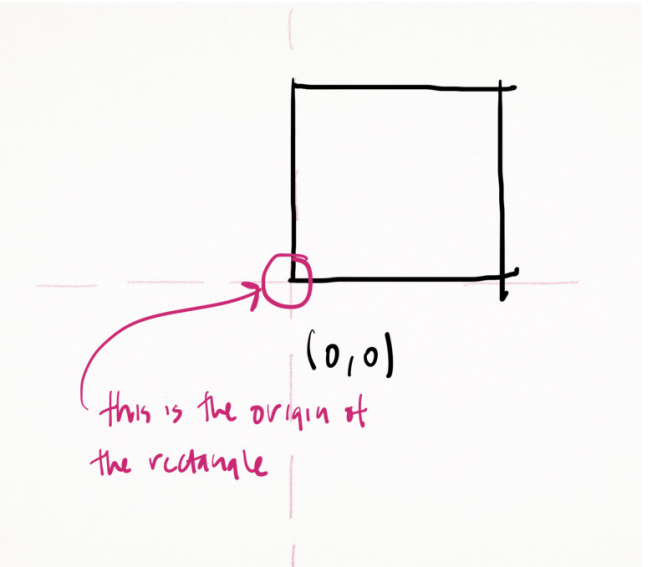
- The node for creating a rectangle requires the origin of the rectangle, which is required to be X and Y coordinates (in other words a two-point input).it also requires a dimension for the work dimension for the height and an angle for the orientation. If you leave the angle for the orientation without an import, it will use zero as the default value.
- The output from the extrusion node is an extruded rectangle.



Origin

The origin of your objects is extremely important, but they are not all the same. For example, the default origin of a rectangle is the bottom left corner, while the default origin of a circle is the centre.

When you are making your objects you have to ensure that you have correctly calculated their setting out points. If you combine different objects (rectangles and circles) you will have to remember that their origins are not both the same, and you may have to do some mathematics to get these two different objects to line up.



Inputs

All marionette notes that create objects require an input. For example if you want to create a rectangle you have to put the height, width, origin, and orientation. if you do not provide an input for these, Vectorworks will use the default values which is usually zero.

The inputs also have to be the correct type. For example on a rectangle, one of the inputs is the origin. The origin import requires an X and Y coordinate, and if you only provide one of these, you are not providing the input in the right format for Vectorworks to use. This will cause an error.

Loops

If you are used to writing computer scripts you will be used to the concept of loops. The current Vectorworks scripting language (vectorscript or Python) has the ability to create loops that have a test at the start and then a series of instructions. At the end of the loop the script will then run the test again to see whether it needs to run through that loop again.

In this example (While – Do) the test is at the start of the loop. If the test condition is true, then Vectorworks will carry out the steps until it reaches the END statement. It then goes back to the start of the loop checks the conditional test and if the test is still true, it will work through the loop again.

Marionette does not include any loops. It uses a different concept. My scripts often include tests and loops, so it can be a challenge to change your thinking away from using these.

Instead of using loops, the strategy is to use a RANGE or a SERIESthese two new concepts will be covered later.

Maths

It really is all about the mathematics. It is easy to create a rectangle, but figuring out how that rectangle relates to other parts of the object requires mathematics. If you want to create a series of objects that follows a sine curve for example, you will have to provide the sine curve formula.

If like me you find maths a challenge, persevere. You will find it gets easier the more you do it and you will find it is a lot of fun to make objects and marionette.

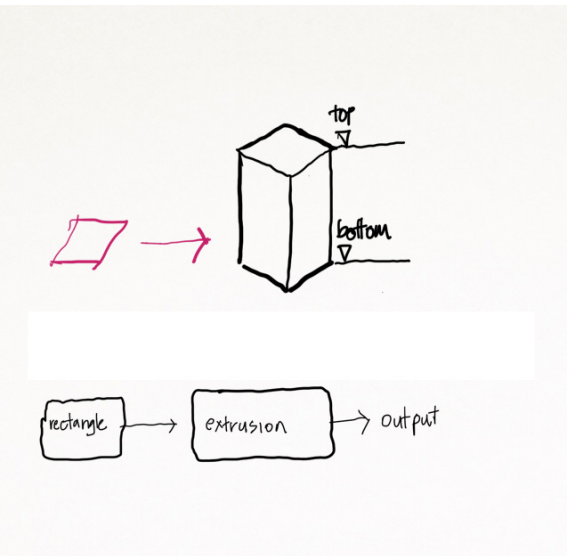
When you look at some of my marionette scripts you will find lots of mathematics. Most of this mathematics is quite simple maths (adding, subtracting, division). The easy way I found to manage this information was to sketch out my object first and use the sketch to help understand the mathematics.

Extrusions

The node for creating an extrusion requires an input for the top and the bottom of the extruded object. unlike your normal extrusion in Vectorworks where you tell Vectorworks the overall length of the extrusion, with marionette you have to tell Vectorworks bottom and the top of your extruded object. The other import is the object that you want to extrude (in this case the output from the create rectangle node).

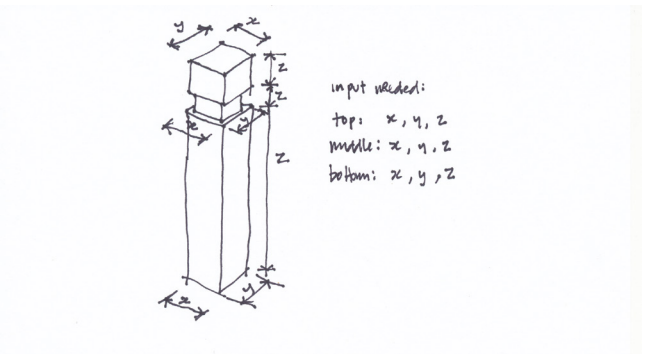
This is different from a normal extrusion which has a bottom Z and then on extrusion height based on that. You will have to adjust your thinking to allow for this change in extrusion heights.

You will be using mathematics to calculate your extrusion heights, but once you understand the difference the should be reasonably straightforward.



Creating a Simple Object

In this project will create a simple extruded object. The purpose of this exercise is to understand how marionette will work on how we can join notes together to create an object.

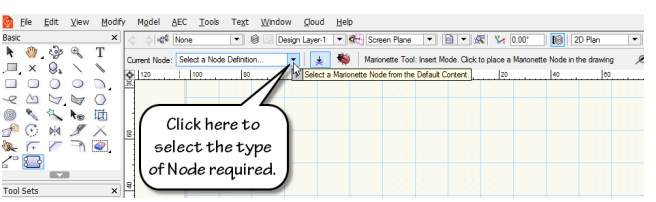


Create The First Part

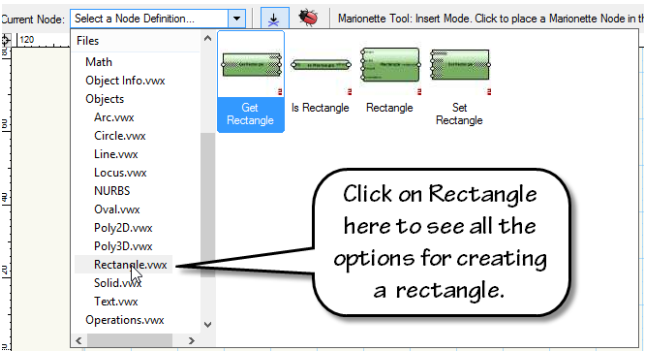
The first part of this object is an extruded rectangle. In order to create this we need to learn to create a rectangle first. If you look at the sketch above your notice that we need an X and Y dimension for the size of a rectangle. This will be our first input.

- Go to the **Basic** toolset.
- Click on the **Marionette** tool

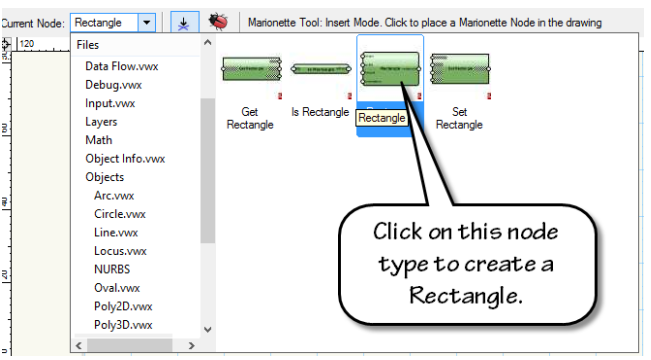
- Go to the Tool bar.
- Click on the pop-up menu to choose the node required.



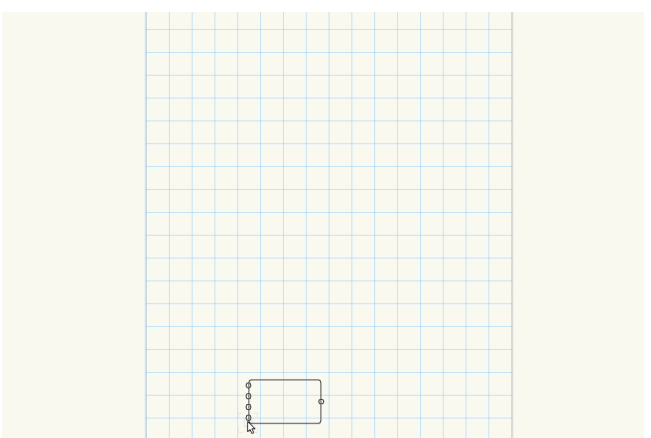
- There are several nodes to choose from and the nodes are grouped together to make them easier to find. Click on the group that deals with rectangles



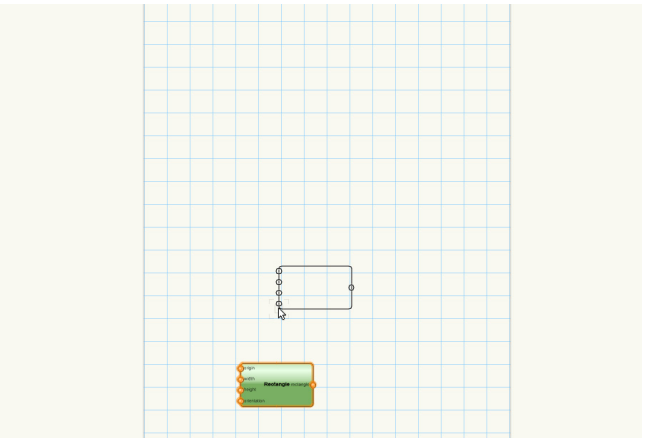
- There are four different notes that deals rectangles, but only one of these deals with creating a rectangle.
- Click on the Rectangle node



- You now have the Rectangle node on the end of the cursor.

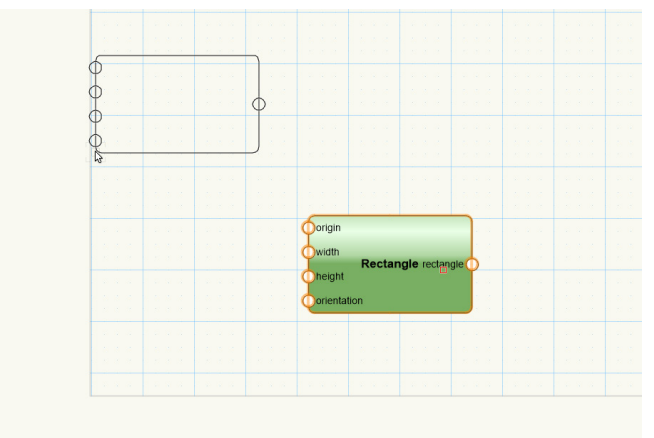


- Click once to place the Rectangle node in the drawing area.
- The rectangle node will still be on the end of your cursor.

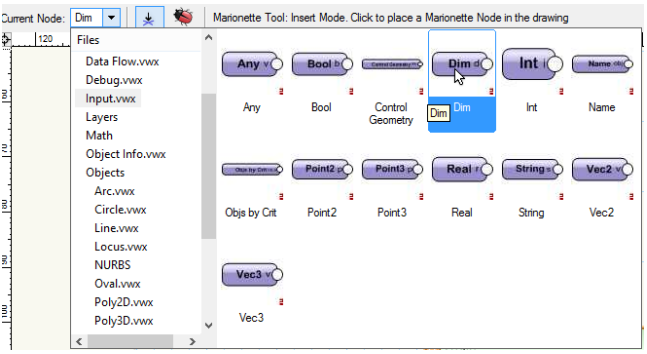


When you zoom in to the rectangle node you will notice that it has four inputs. This means that you will have to go back to the marionette tool and select the appropriate inputs for each the Rectangle node.

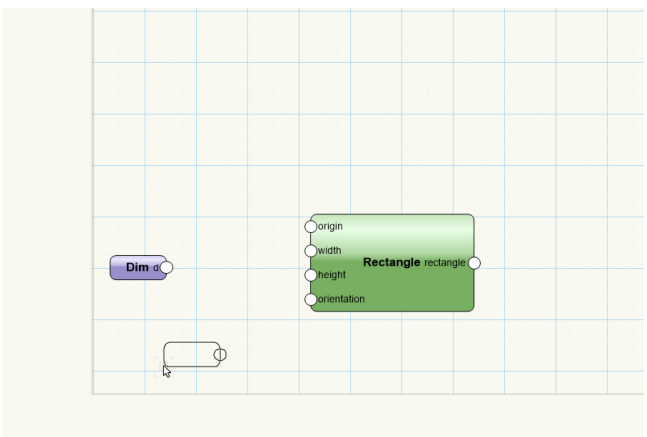
If you do not specify an input value Vectorworks will use the default value zero



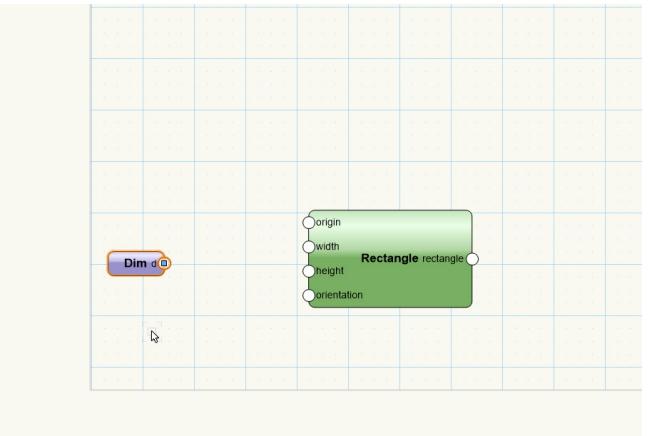
- Go to the Tool bar.
- Click on the node pop-up menu.
- Scroll to find **Input**. There are a range of inputs and you must select the appropriate input for each part of the rectangle node.
- To start with, we want to input the size of our rectangle.
- Choose the **Dim** input.



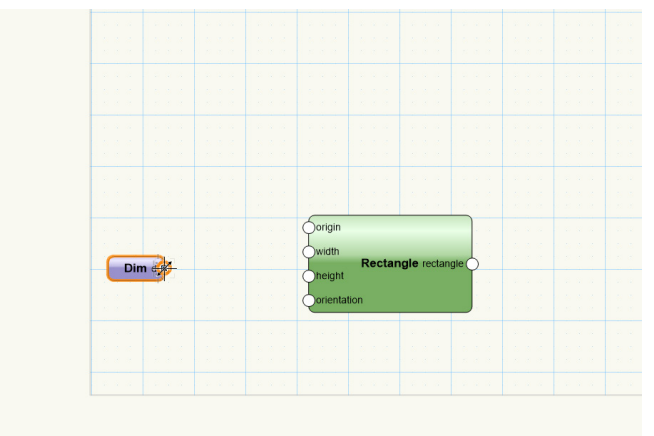
- Move into the drawing area and place the **Dim** input near the Rectangle node.
- Notice that all of the inputs for the rectangle note are on the left-hand side of the node. Marionette always works from left to right and the data always flows from the left-hand side to the right-hand side.
- Place your inputs on the left-hand side of the Rectangle node.



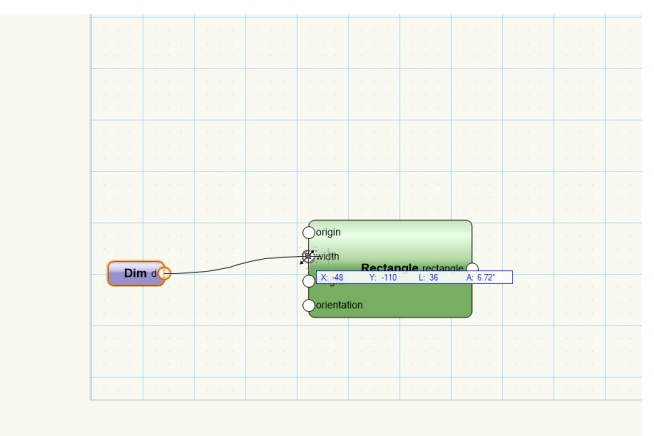
- Go to the **Basic** toolset.
- Choose the **Selection** tool.
- Select the **Dim** input.
- When you select the input you will notice a small blue handle on the right-hand side of this object. This blue handle controls the wire that will join this input to the Rectangle node.



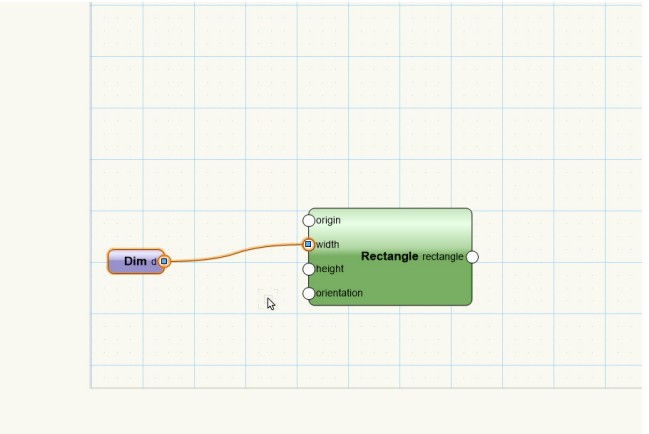
- Move your cursor to the blue handle.
- Click once.



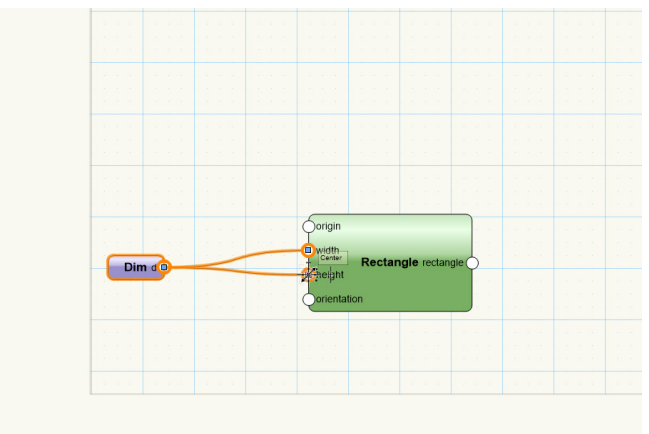
- Move your cursor to the required input on the Rectangle node.



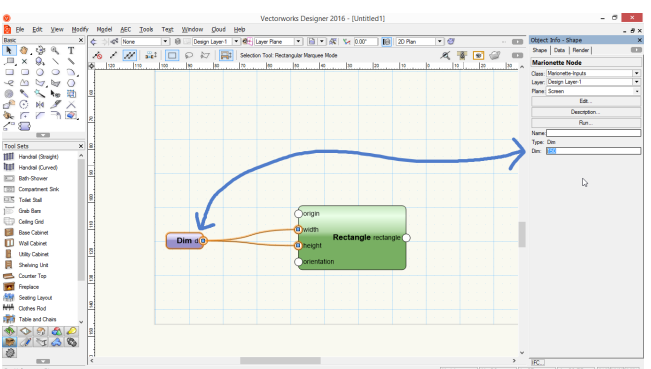
- Click once.
- This is now connected the Dim output to the Rectangle input.
- Any data that you type into the dim input will now flow along the wire into the Rectangle node.



- If you wanted the same dimension for the width and for the height of the rectangle, you can use the same output node and connect it to more than one input on the rectangle node.
- In this image you can see that the dimension output has been connected to both the width and the height input for the rectangle. This means that the rectangle object will now be square.

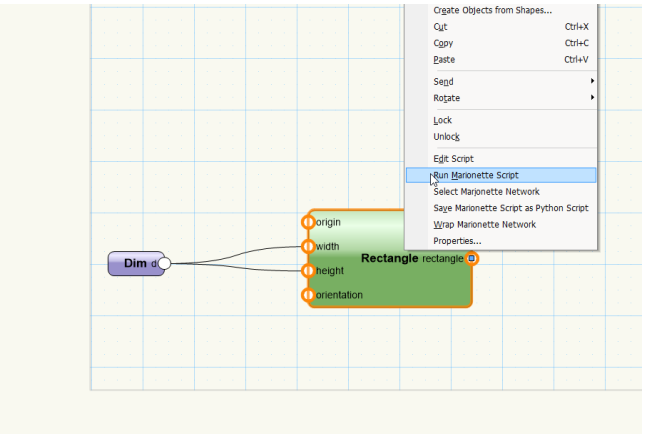


- To enter information into the Dim node, select the Dim node, then use the Object Info palette.

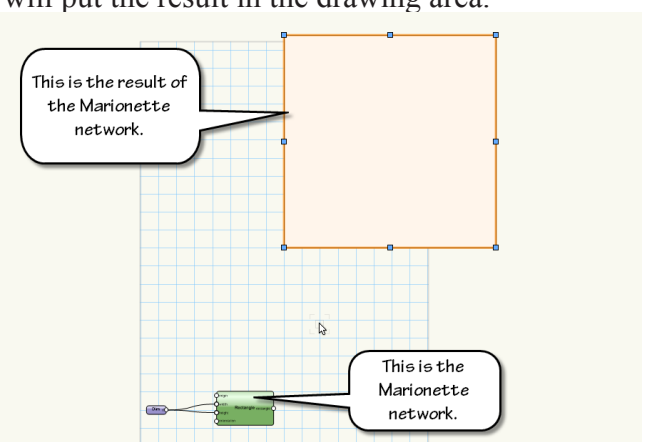


- when you have entered the required information, right click on any part

of the Marionette network and choose **Run Marionette Script**.

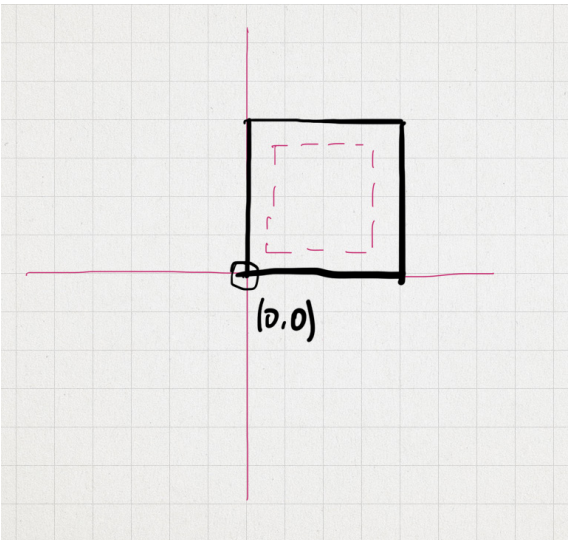


Vectorworks will then carry out all the steps in this Marionette network and it will put the result in the drawing area.



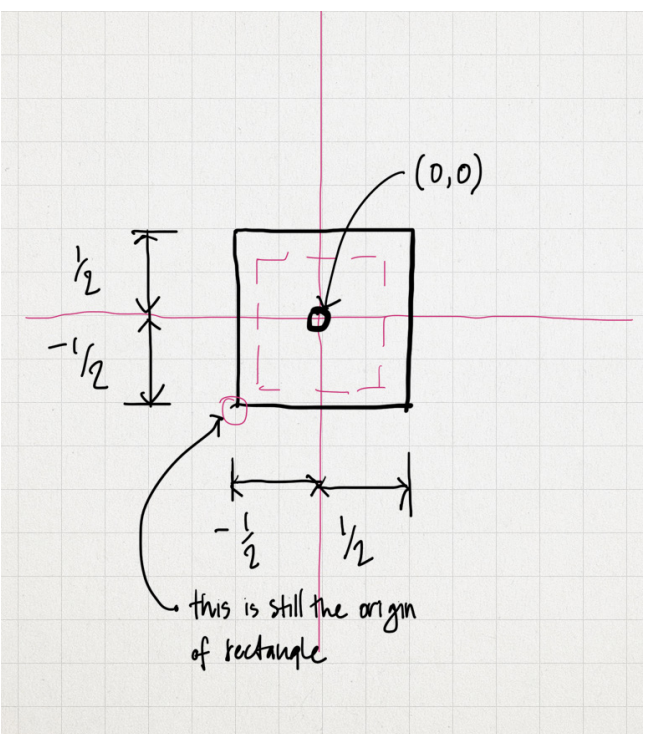
Origins

It is important to understand the origins of the objects in marionette. A rectangle has its origin at the bottom left while a circle has its origin at the centre. If you try to combine a rectangle with a circle, the origins are not consistent and you might find your objects don't line up properly.



Part of the design of your marionette network is to ensure that your objects will line up when they are created and this might require some mathematics. For example,if you want the centre of your rectangle to be placed at (0,0) then you would have to start the rectangle at the right location.

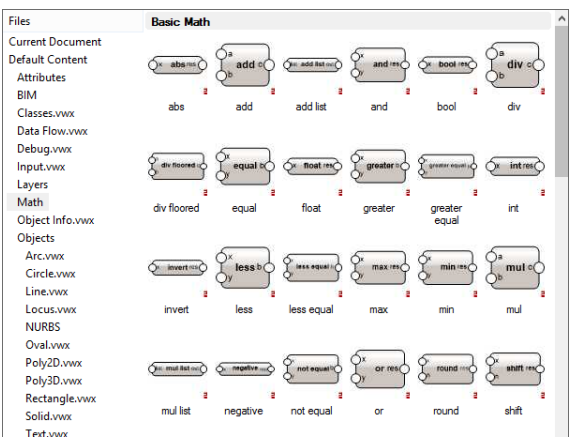
The actual location you should use would be -1/2 in the X direction and -1/2 in the Y direction for the rectangle. This will have to be built into our marionette network for the origin of the rectangle.



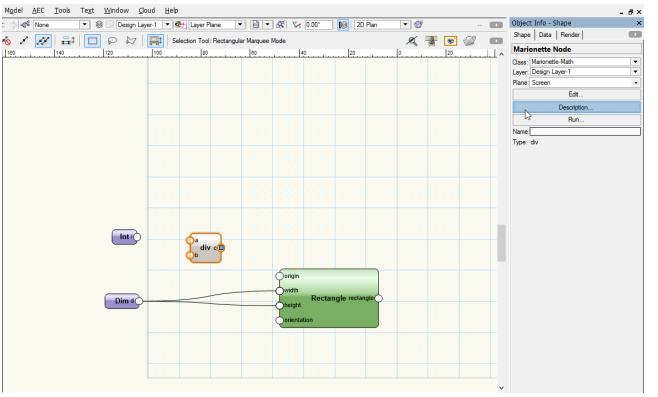
- Go to the **Basic** toolset.

- Click on the **Marionette** tool

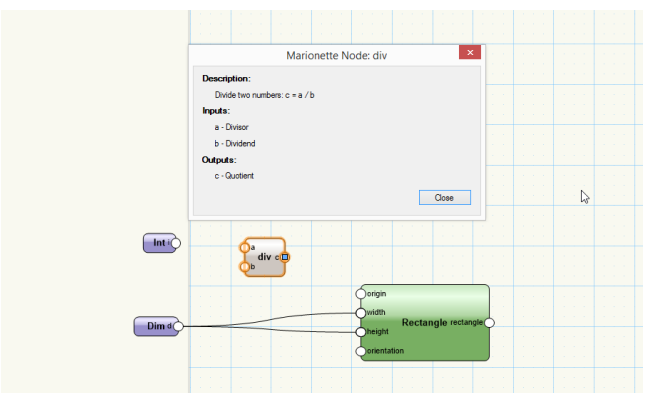
- Go to the **Tool** bar.
- Click on the pop-up menu and choose **Maths**.
- There are several nodes for mathematical operations.
- Click on the **Div** node.



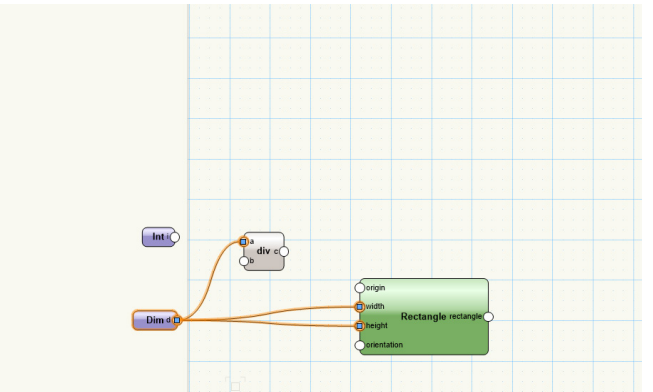
- You now have the **Div** node on the end of the cursor.
- Click once to place the **Div** node in the drawing area.
- it isn't always obvious what the inputs and what the outputs from each node are. One way to find out is to look at the description.
- When you have a node selected, go to the Object Info palette.
- Click on the Description… button.



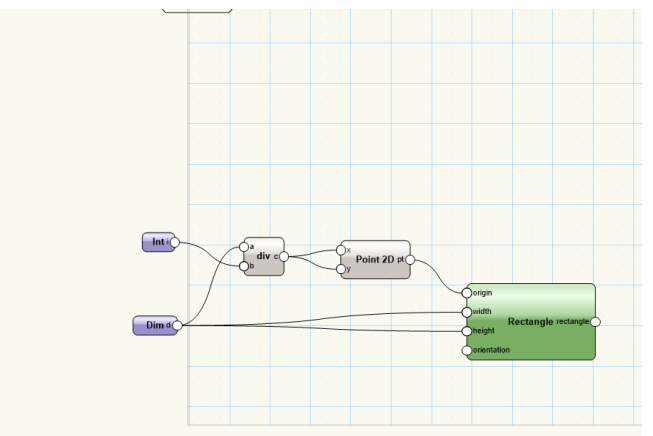
- This will open a dialog box explaining what the inputs and the outputs relate to.
- In this example it is showing you that the output (C) from the **Div** node is A divided by B. for our example we have to put the size of the rectangle into the A input, and 2 into the B input so that we can find the size of 1/2 the rectangle.



- We can use the size of the rectangle, divided by half, make this into a negative and use this as the origin of the rectangle. Just to refresh your memory, if you divide the size of the rectangle by -2, you will get the negative half of the rectangle that we require.
- Place an integer (Int) input and set its value to -2.



- Use the Marionette tool to place a Point2D node (which you will find in the Points area). the origin of the rectangle requires a coordinate input (x,y), which is what the Point2D node creates.
- If you refer to one of the earlier sketches you may have noticed that we need to have the origin of the rectangle with a minus X and a minus Y location.
- You can use the output from the division for both the X and the Y on the Point2.
- Connect the output from the Point2D the origin of the rectangle.



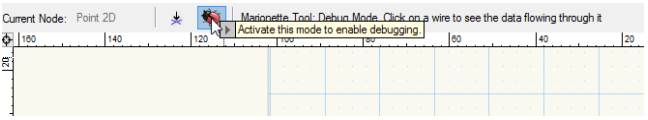
Debug

It is extremely useful to be able to check to see what the data flow is down each wire. That's what the debug mode is for.

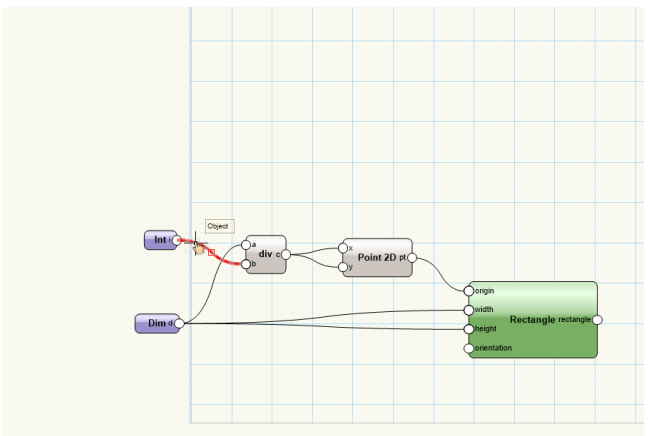
Debugging is what you call it when you look for errors in your script, and the debug tool allows you to look for areas in your script by checking the data that is flowing down each wire.

- Go to the **Basic** toolset.
- Click on the **Marionette** tool

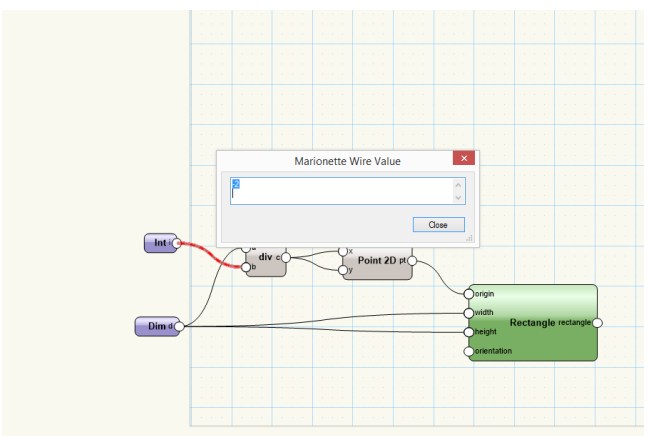
- Go to the **Tool** bar.
- Click on the Debug mode.



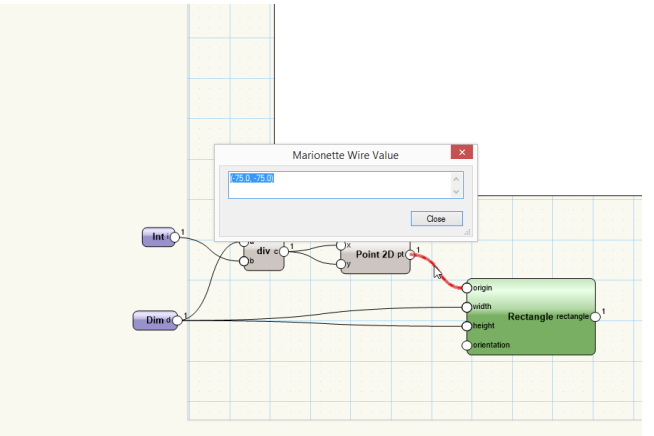
- Move to one of the wires that connects the nodes. The wire should highlight in red.



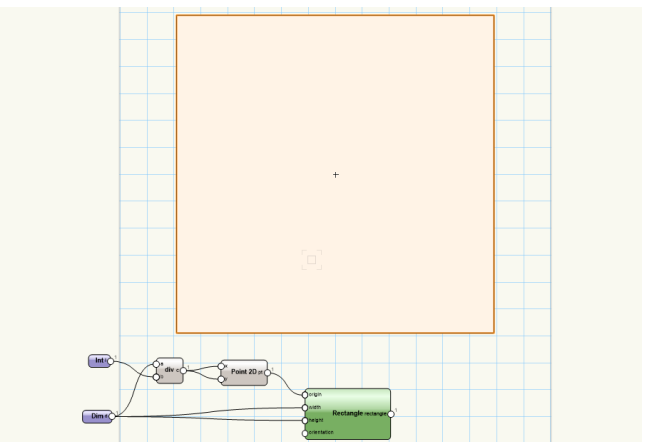
- Click once on the wire.
- A dialog box will open showing you the data that flowing down the wire. The data might be a number, as shown in this image.



- Here you can see the calculation of our origin, which should be half of the size of our rectangle, but with negative numbers.



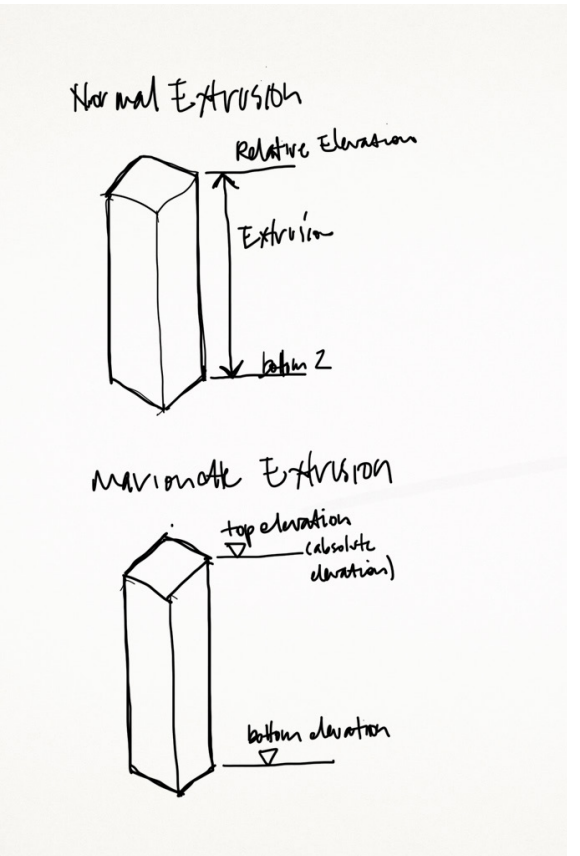
In this image you can see the result of our script on our Marionette object. You can see that the origin of our rectangle has been moved and our rectangle is now centred at (0,0).



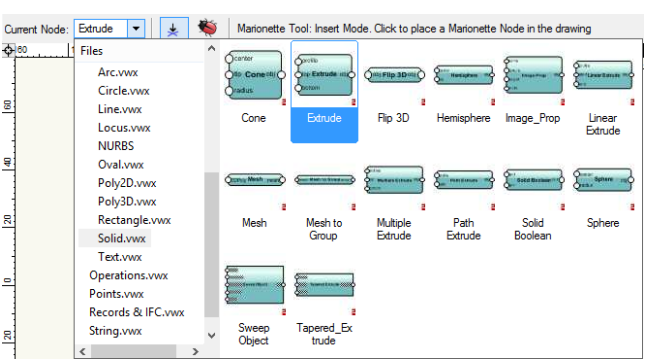
Extrusion

Now that we have our rectangle in the right location and we have the origin starting correctly, we can now start to look at how to extrude this object. when you create an extrusion in Vectorworks using the extruded command, you only have to give it the length of the extrusion. You could call the elevation to the top of this extrusion a relative elevation.

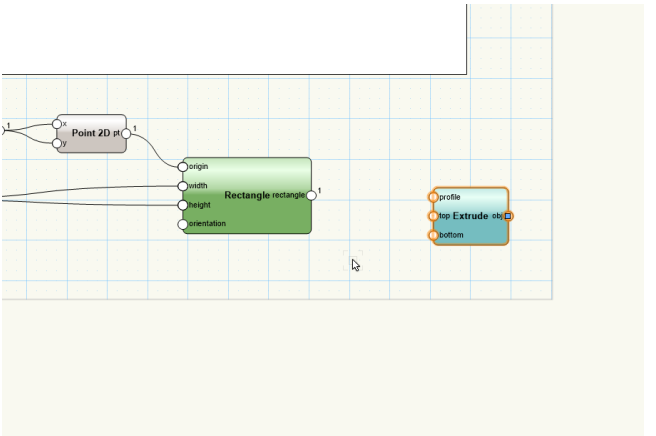
In marionette, you have to give the start and the end of the extrusion, which are measured in absolute terms from zero on the design layer.this can be very strange until you get used to it.



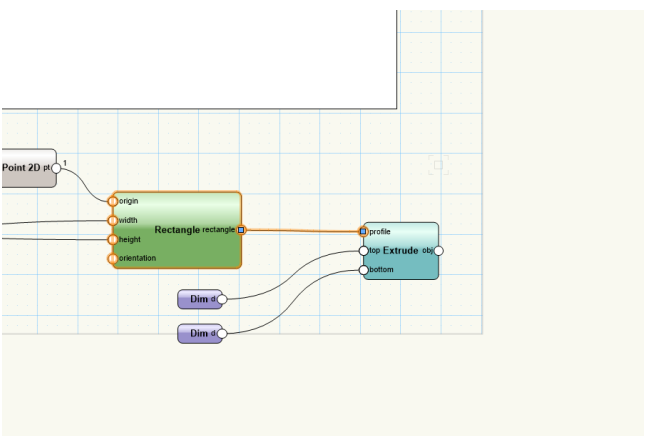
- Go to the **Basic** toolset.
- Click on the **Marionette** tool
- Go to the **Tool** bar.
- Click on the **Current Node List** pop-up menu.
- Scroll down the list of node types until you get to **Solid**.
- from the Solid node types choose **Extrude**.



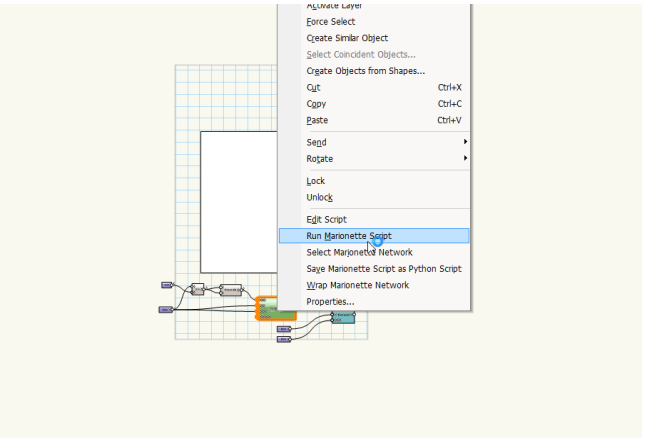
- Move your cursor into the drawing area and place your extrude node to the right of the rectangle node.



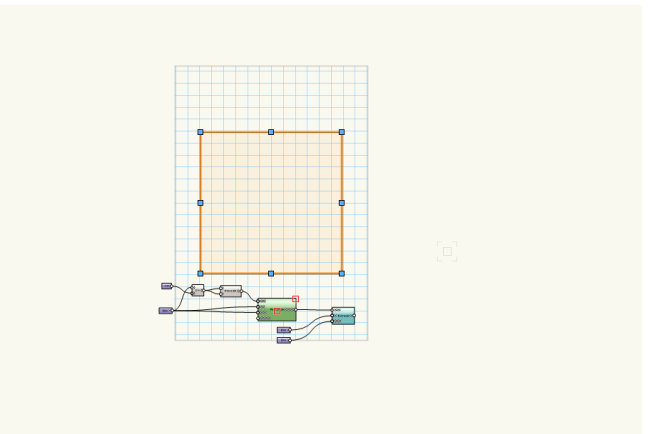
- The extrude node requires two inputs, one for the bottom elevation and one for the top elevation.
- You can copy the **Dim** nodes that you've already used and drag them to the left of the **Extrude** node. A technique for dragging a copy is to drag the object and hold down the Option key on a Macintosh or the Control key on a Windows machine.
- Connect the output of the **Rectangle** node to the **Profile** input on the **Extrude** node.



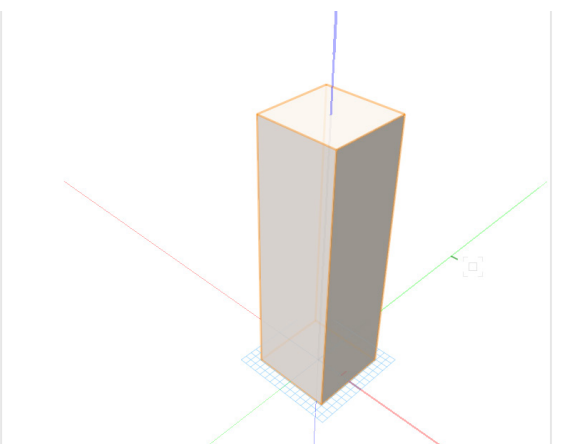
- If you want to test the script so far, right click on any of the nodes and choose **Run Marionette Script…** from the contextual menu.



Vectorworks will work through the marionette steps and create the resulting marionette object, which will appear on the screen as a group.



If you check your work in 3-D you should see an extruded object.

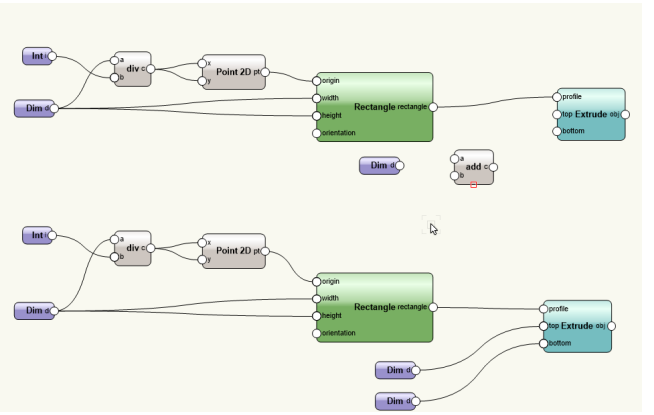


Second Part

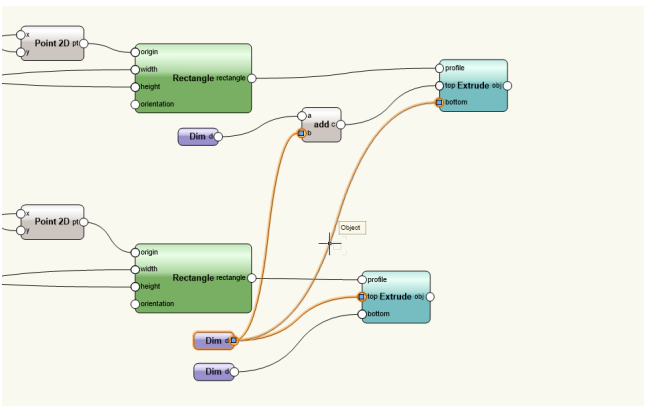
The next part is to build a small extruded rectangle that will sit on top of the first part. For this object we would like to have individual control over

its extrusion height, which means we need a new input for the extrusionand we also need to control the size of the rectangle, which means to new inputs for the width and height.

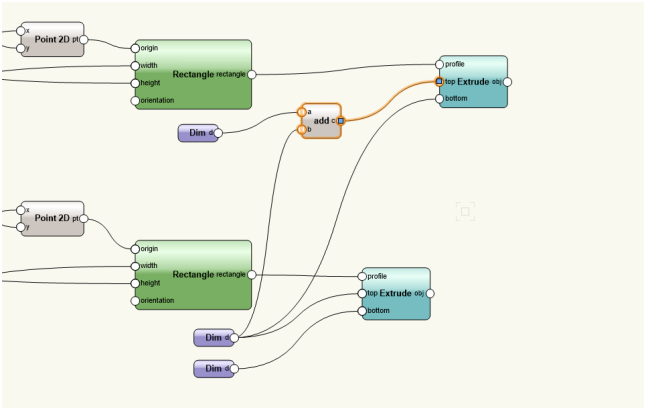
- Remember that you can copy and paste any of the existing inputs or nodes. Also remember that we have solved the problem of the origin so reuse those nodes as well.



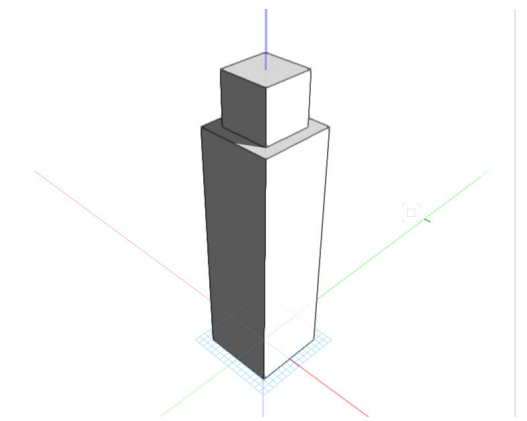
- If you refer back to the earlier part of the manual we talked about extrusions being absolute heights, this becomes crucial at this point because we already have an extruded object with the top and bottom and we need to use the top of the previous object as the bottom of the next object.



- Then we can take the information from the top of the previous object add it to our extrusion height that we require and then use those to set the top of a new extrusion.



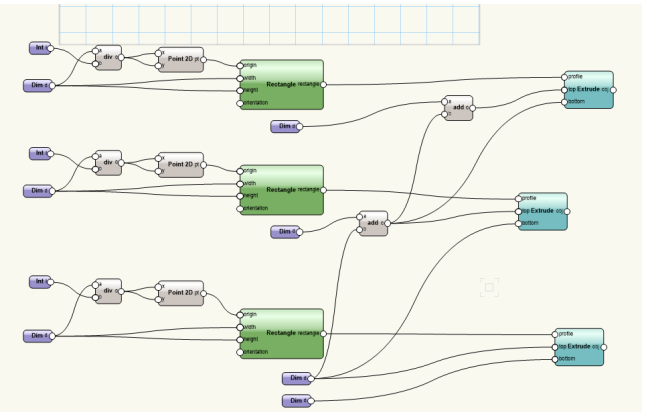
When you run the marionette scriptand you look at the object in 3-D you should get one object sitting on top of the other. If you do not then there is an error in your script.



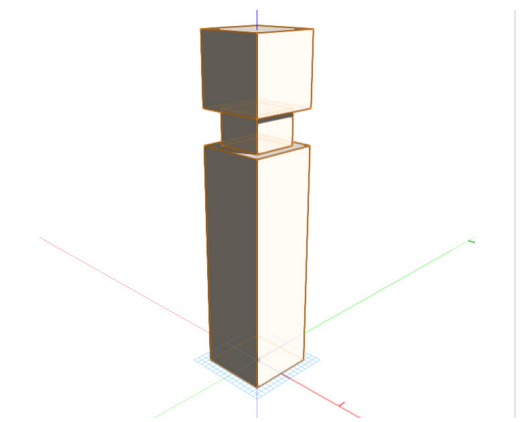
The Top Part

The top part is another extruded rectangle that will sit on top of the previous part. For this object we would like to have individual control over its extrusion height, which means we need a new input for the extrusion and we also need to control the size of the rectangle, which means to new inputs for the width and height.

- Remember that you can copy and paste any of the existing inputs or nodes. Also remember that we have solved the problem of the origin so reuse those nodes as well.
- Use the top of the previous object as the bottom of the next object.



- When you run the marionette script and you look at the object in 3-D you should get one object sitting on top of the other. If you do not then there is an error in your script.



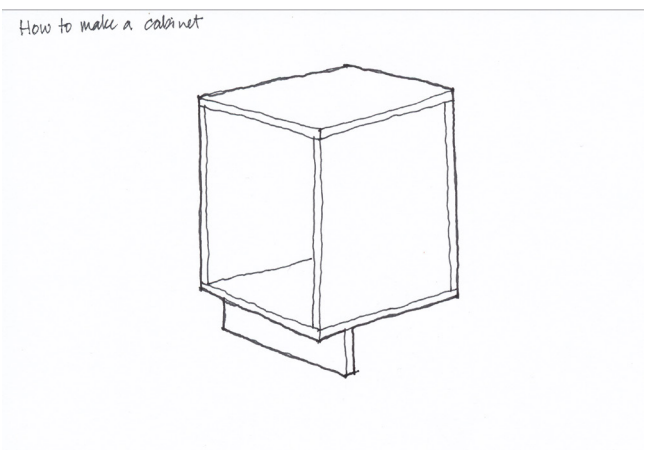
Creating a Cabinet

In this exercise we will look at designing a cabinet. to keep this cabinet simple it will have no doors and no drawers, it is just a simple carcass.

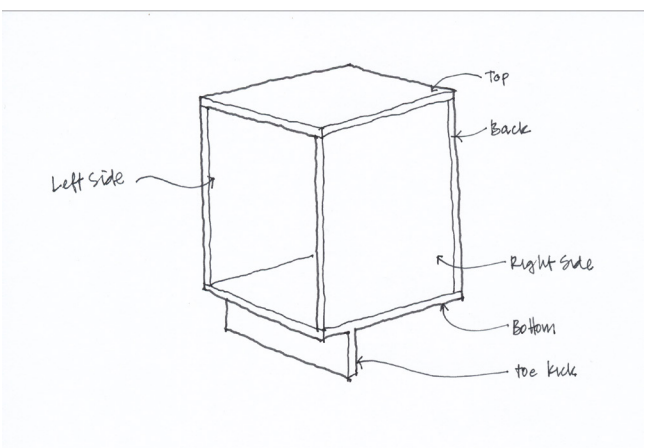
Start with the Design

I find it extremely useful to start by sketching out the design of the object that I want to create.

- If you look carefully at the sketch you might notice that I have sketched not just the outline of the cabinet, I have also sketched the individual components. This allows me to think about where the components stop and start and where they connect together.



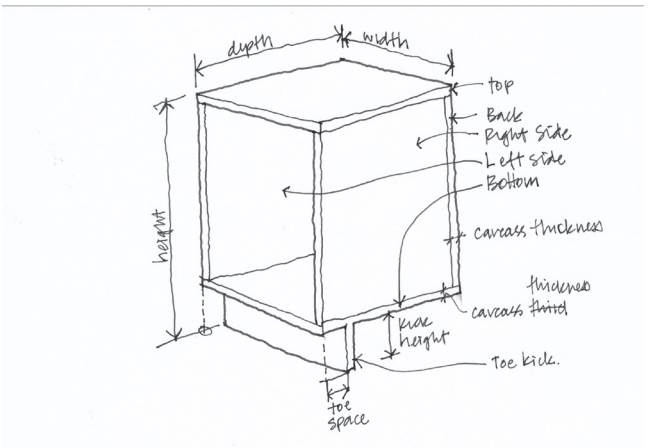
- The next step is to name the components. This cabinet is can be built out of several small parts, so it's useful to name each one with the unique name.



Now that all the parts have been named you can start to calculate where you need dimensions. Just because I have chosen where I want to have my

dimensions, and what I believe are the important dimensions, it doesn't make it right. There are several ways that you could dimension the subject and your choice will come down to a philosophy about the dimensions that you think are important.

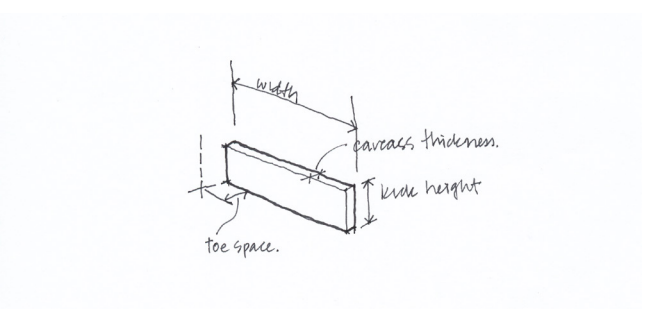
- All these dimensions will be the required input values to build the cabinet. For each dimension shown on the sketch I will have to create a marionette input.



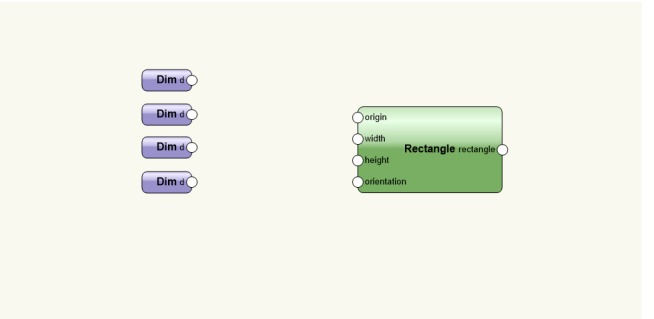
Build The Kick

I found it was easy to create my marionette network if I broke my object down into small simple chunks and then looked at each chunk as an individual.

In this image you can see the kick object. I have also worked out which dimensions I will need for this part. I will need an input for the cabinet width, the kick height, the toe space (this is the setback of the kick from the front of the carcass), and the carcass thickness.

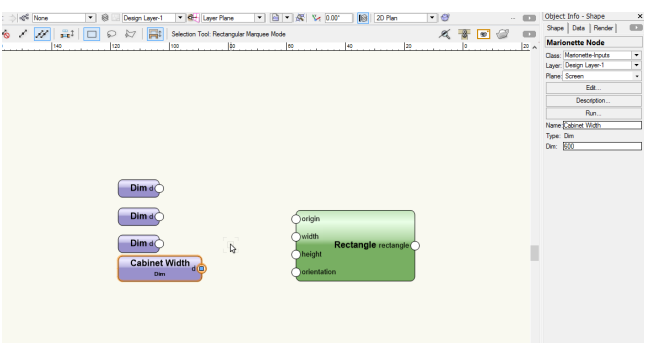


- Use the marionette tool to place a rectangle node and four inputs.

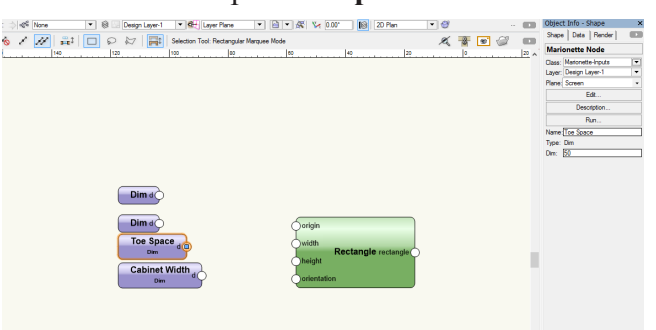


As we start to create more complex objects, it could get confusing as to which input is which.

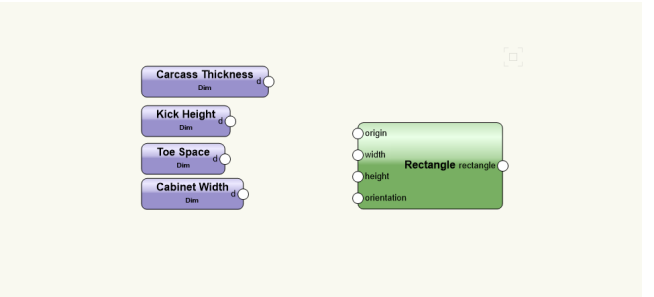
- Select the first input node.
- Go to the **Object Info** palette.
- There is a field on the Object Info palette for naming the input. Change the name to **Cabinet Width**. Any inputs that you specifically name this way will appear later on the Object Info palette when you convert this marionette script into an object



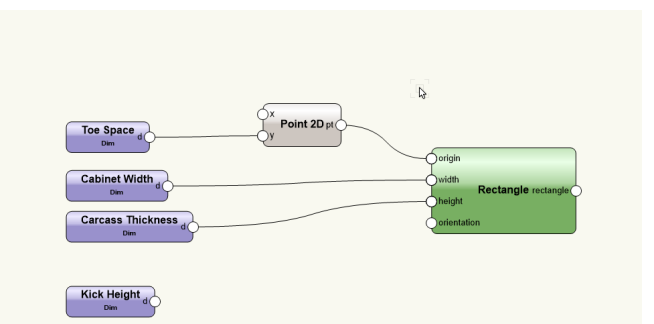
- Name the next input **Toe Space**.



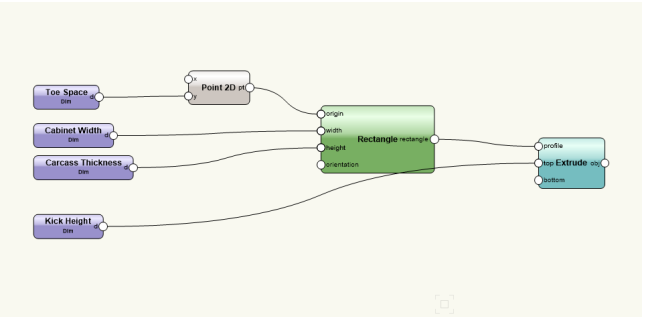
- Name all the inputs.



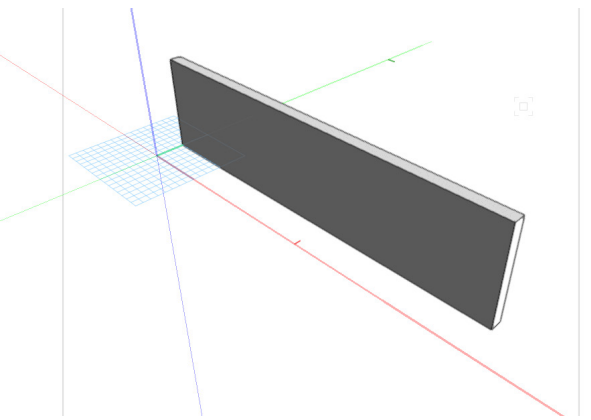
- The Toe Space input is used to create a setback for the kick from face of the carcass, but it cannot be used directly for the origin input of the rectangle node, because this input requires coordinates (X, Y).
- Use the **Marionette** tool to find a Point node.
- Place the **Point2D** node between the toe space in the rectangle nodes.
- Connect the Toe Space output to the Y input on the **Point2D** node. We do not need to use the X input, if it is left empty, Vectorworks will use a default value of 0.
- Connect the output from the **Point2D** node to the **origin** input on the Rectangle node.
- Connect the **Cabinet Width** input to the **width** input on the Rectangle node.
- Connect the **Carcass Thickness** input to the **height** input on the Rectangle node.



- Use the **Marionette** tool to find the **Extrude** node.
- Place the **Extrude** node after the Rectangle node.
- Connect the output from the rectangle node to the profile input on the extrude node.
- Connect the **Kick Height** input to the **top** input on the Extrude node. We donot need to connect anything to the bottom input, Vectorworks will use the default value 0.

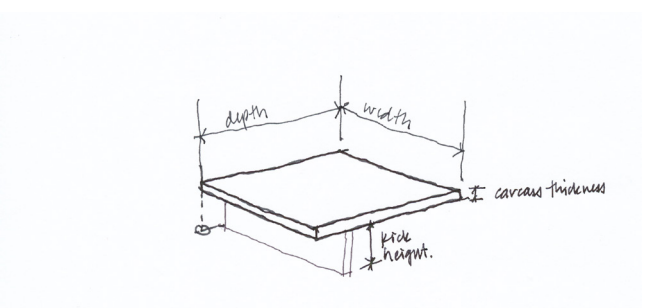


- Run the marionette script and you look at the object in 3D to check hte object. Check for any errors in your script.



Creating the Carcass Bottom

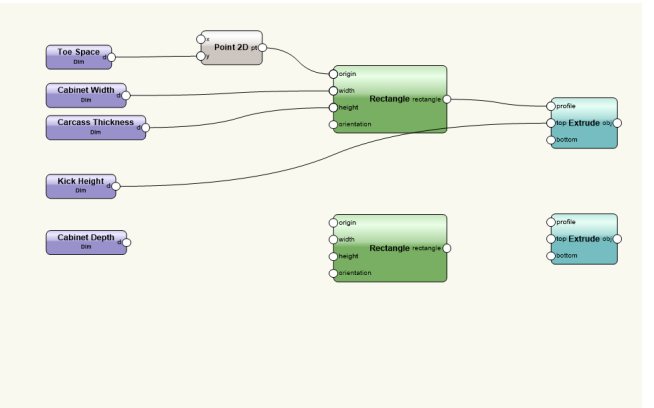
In this image you can see the carcass bottom object. I have also worked out which dimensions I will need for this part. I will need an input for the cabinet width, the cabinet depth, the kick height, and the carcass thickness. some of these imports I have already used for the kick object, and these inputs and values can be reused on this object. The only new input I need is the cabinet depth.



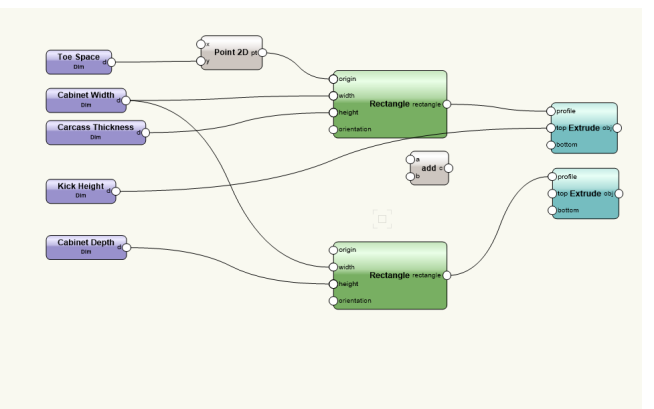
- Create a new Dim input. You can either use the marionette tool to place a new input or you can drag a copy of one of the existing inputs and

change its name on the Object Info palette. I find reading a copy the quickest way to duplicate nodes.

- Drag a copy of the rectangle and extrude nodes.
- I find it easier to place all the nodes I need and then connect them to the information that they require.

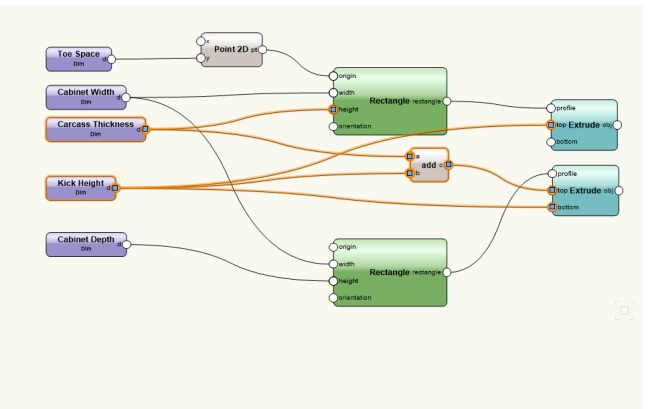


- Connect the **Cabinet Width** input to the **width** input on the new rectangle node.
- Connect the **Cabinet Depth** to the **height** input on the new rectangle node.
- Connect the output from the new rectangle mode to the profile of the new extrude node.
- use the marionette tool to place a node to add two numbers together. You will find this node in the Maths area

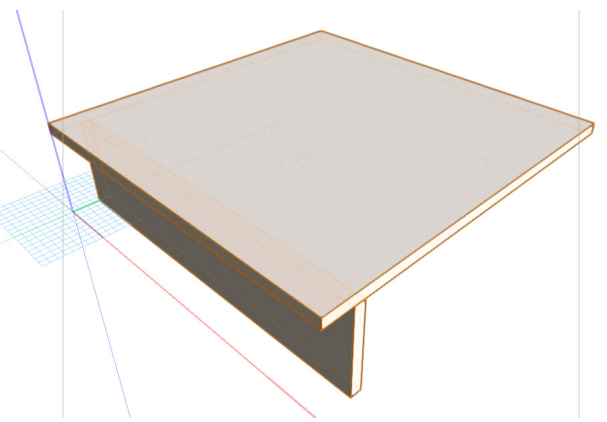


- Remember that the way extrusions work is that you are going to define the start and the end elevation of your extruded object. If you look back at the earlier sketch you'll see that this object (the bottom of our carcass) elevation at the start of this object is the top of the kick and the elevation at the top of this object is the kick plus the thickness of the carcass.
- Connect the kick height to the bottom of the new extrude node.

- Connect the carcass thickness and the kick height to the **add** node.
- Connect the output from the add note to the top input for the new extrude node.



- Run the marionette script and check your results in 3D. If it does not look correct there is an error. Go back and fix any errors.



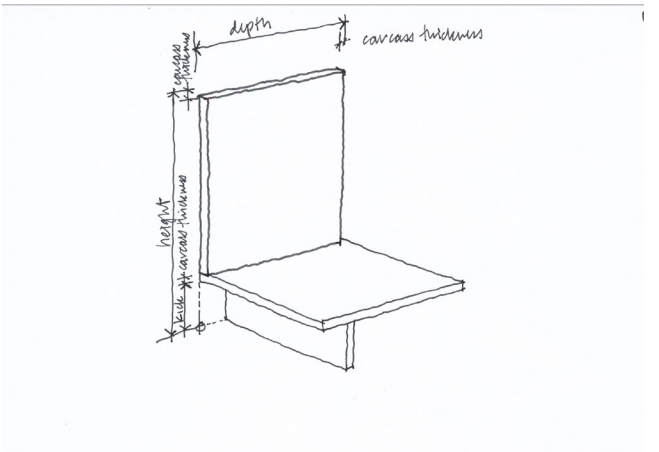
Left Side

sketching out just one part of the time allows you to see what information you require for this one part. The way that I have placed the dimensions gives a clue as to the way we will calculate the sizes of the subject.

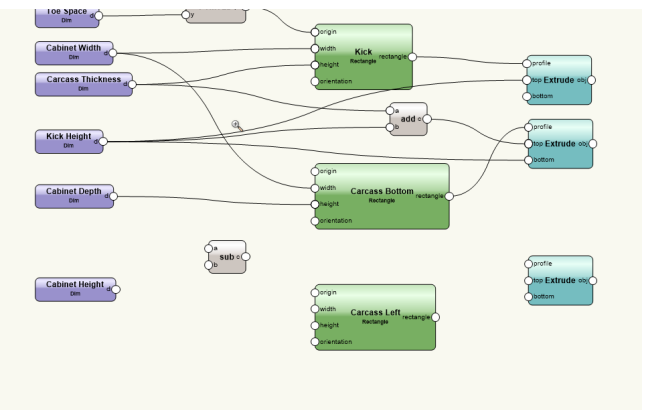
The top height of this object is equal to the overall height of the carcass less the carcass thickness. This is a new input, so when you input will have to be created for our marionette script. we will also require a new node to subtract the carcass thickness from the overall height of the object. this will be a subtract node from the maths area.

The length of the side is equal to the overall depth of the carcass less the carcass thickness. We already have this input for the previous part.

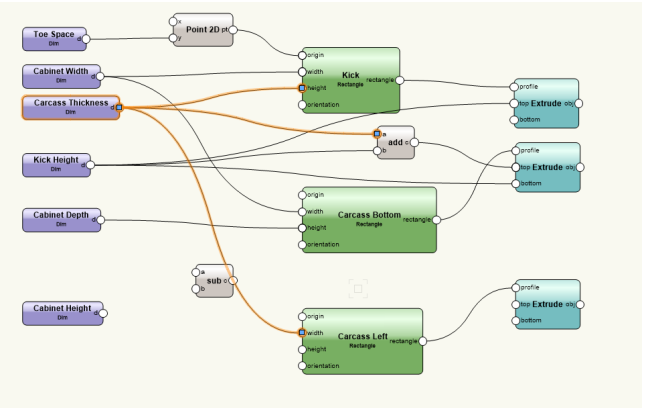
The bottom of this object is equal to the kick height plus the carcass thickness. We are ready have these inputs as well.



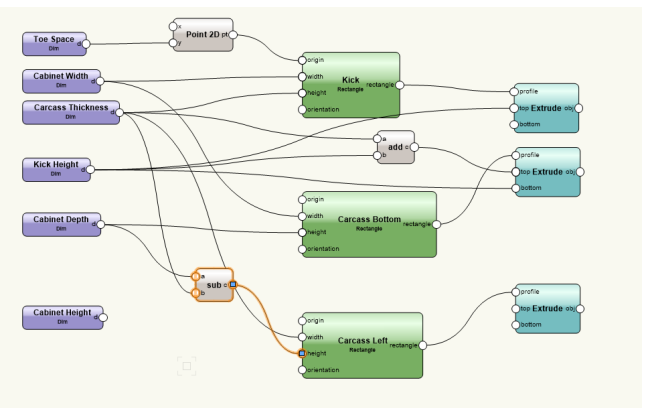
- Place the new nodes. Remember that it's often easier to drag a copy of existing notes and rename them.
- You will find it a lot easier to deal with the more complex marionette scripts if you start to name all of your nodes. Notice in this image that even the rectangle notes have been named to make it easier to understand which part is which.



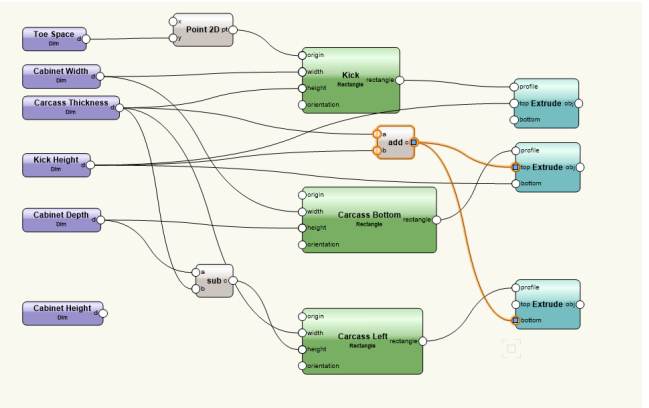
- Connect the carcass thickness to the width input for the new rectangle node.
- Connect the output from the add note to the top input for the new extrude node.



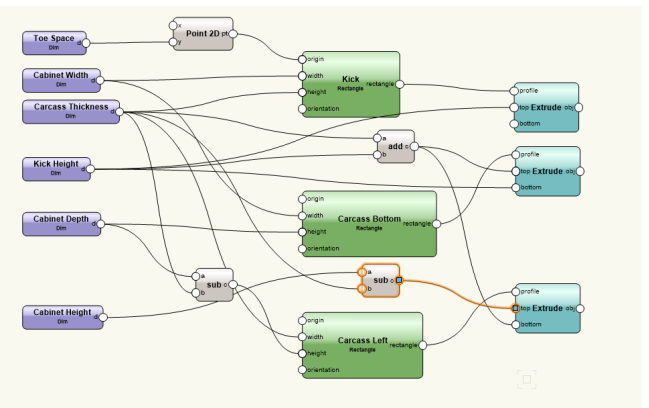
- Connect the **Cabinet Depth** to the **A** input for the **sub** (subtract) node and connect the **Carcass Thickness** to the **B** input. this will give us the depth of this part that we need (overall depth minus the carcass thickness).
- Connect the output from this note to the height input of the rectangle node.
- Connect the output of the rectangle node to the profile of the new extrude note



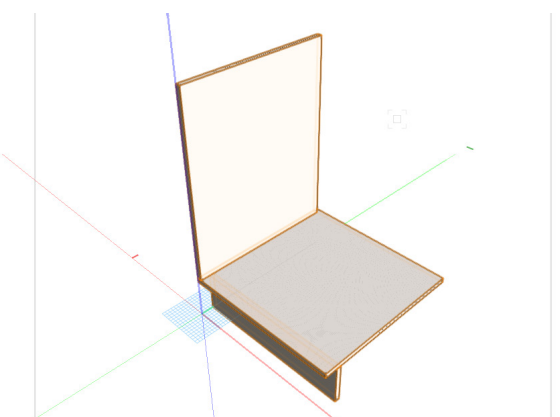
- We already have the calculation for the start of this extrusion, it is the same elevation as the top of our carcass bottom object.
- You can use the same output from one node to go to several places, so connect the output of your addition node from the last part and connect it to the bottom of the extrude node.



- To calculate the top of our left side we will need to take the overall height of the cabinet and subtract the carcass thickness.
- Place a new subtract to node.
- Connect the overall cabinet height to the A input.
- Connect the carcass thickness to the B input.
- Connect the output to the top of the extrude node.



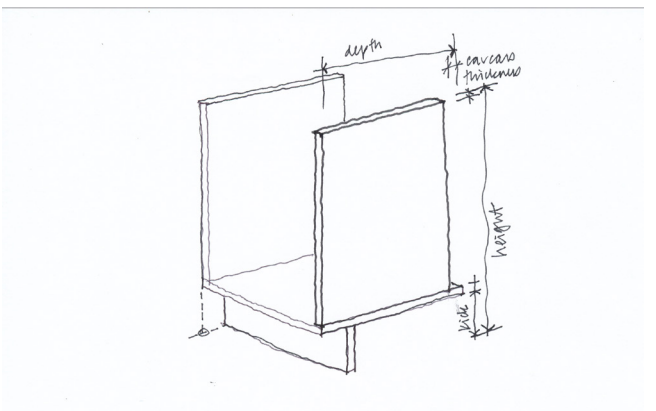
- Run the marionette script and check your results in 3D. If it does not look correct there is an error. Go back and fix any errors.



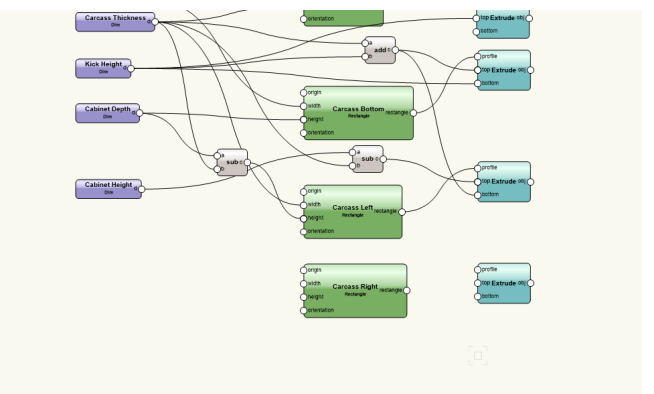
Right Side

The right side is almost a perfect copy of the left side. I say almost because its origin is different, which means that we have to calculate the overall width of the unit and subtract the carcass thickness.

We do not require any new inputs for this part, but we do require an additional subtract node.

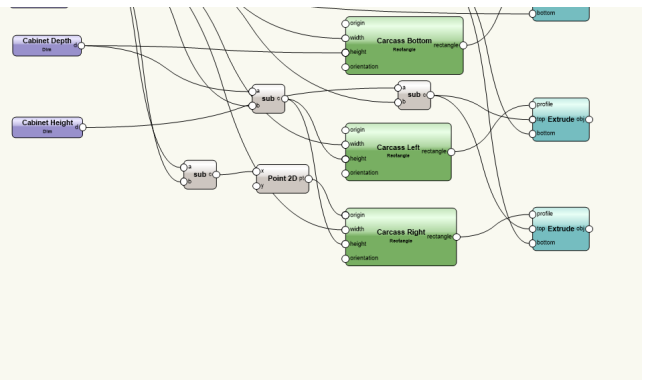


- Drag a copy of the rectangle and extrude nodes.

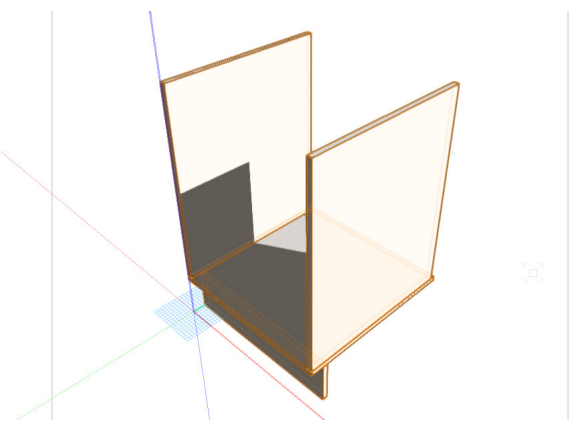


- because we are moving the origin of this part, we will have to repeat the technique we used for the kick part, we will have to calculate a position and use the Point2D node to convert this to coordinates.
- Place a subtract node and place a copy of the Point2D node to the right of it.
- Connect the **Cabinet Width** to the **A** input of the subtract node and connect the **Carcass Thickness** to the **B** input of the subtract node.
- Connect the output of the subtract node to the X input on the Point2D node.
- Connect the output of the Point2D node to the origin input on the new rectangle.

- Use the same inputs for the width and the height of this rectangle as you used for the left side of the carcass.
- Connect the output of the rectangle node to the profile of the new extrude node.
- Use the same inputs for the top and bottom of your extrude as you used for the left side of the carcass

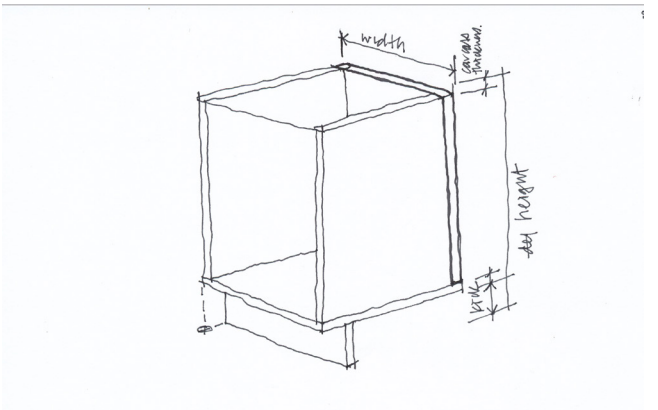


- Run the marionette script and check your results in 3D. If it does not look correct there is an error. Go back and fix any errors.

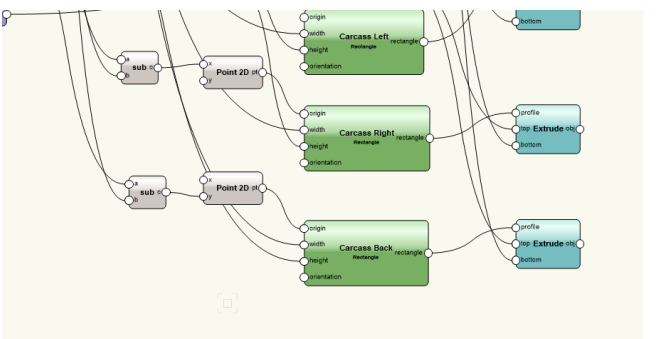


Back

By now we have all the inputs we need, but they will have to be rearranged slightly to make this object. We already have the cabinet width input, the carcass thickness, the cabinet depth, the elevation for the start, and the elevation to the end of our extrusion. The only thing we need to calculate for this component is the origin of the the rectangle.

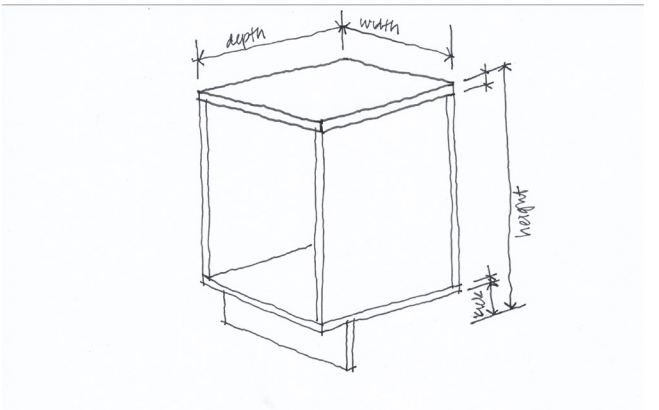


- The origin for this object is the cabinet depth minus the carcass thickness. We already have these inputs so all we need is a mathematical function to subtract the carcass thickness from the cabinet depth.
- Connect the inputs from the Cabinet Width in the Cabinet Depth to the new rectangle node.
- Connect the elevation inputs that you used for the left and right side extrusions.

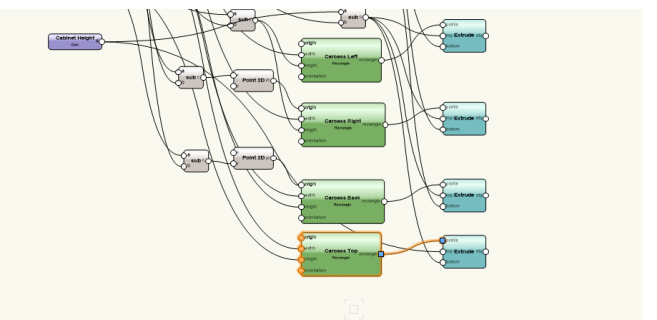


Top

After all the work you've done so far the top is relatively straightforward. It's the overall cabinet width and the overall cabinet depth. The only challenge is the extrusion elevations, but this information is also easily available.

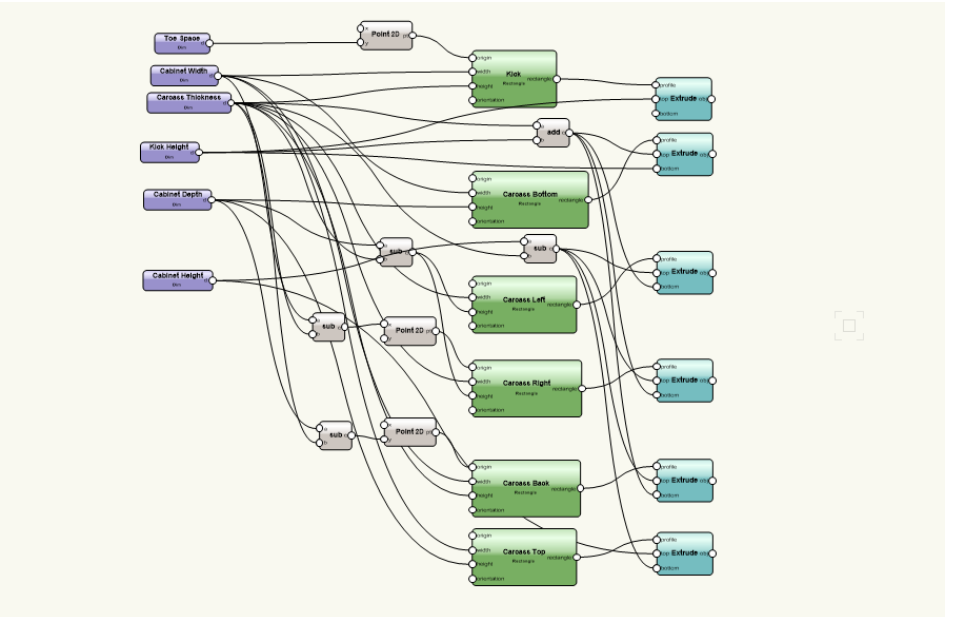


- Connect the Cabinet Width to the width input for the new rectangle node.
- Connect the Cabinet Depth to the height input for the new rectangle node.
- Connect the output from this rectangle to the new extrude node.
- Connect the same output you used for the top of the right side to the bottom of the new extrude node.
- Connect the Cabinet Height input to the top of the extruded node.

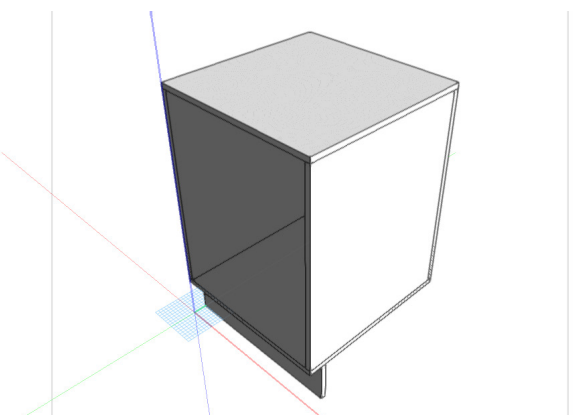


This is what the final network looks like for my example. Because there is no rule about the positioning of your nodes in the network, your network might look different to mine.

It might look like I am trying to be overly neat and tidy, but you will find that it pays to leave a lot of spaces between your nodes so that you can easily follow the network.



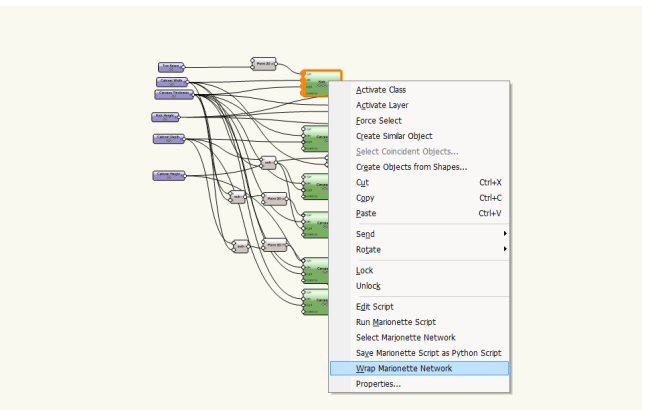
- Run the marionette script and check your results in 3D. If it does not look correct there is an error. Go back and fix any errors.



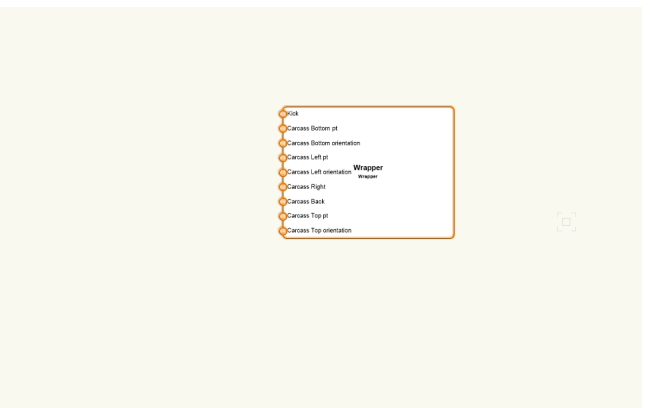
Wrapping a Marionette Network

There are times where you create a network that you want to reuse and other locations. An example of this would be where you want to repeat a series of steps. Creating a wrapper around your network will temporarily hide all the steps, leaving a single marionette node.

- To wrap a marionette network, right click on it.
- Choose **Wrap Marionette Network** from the contextual menu.



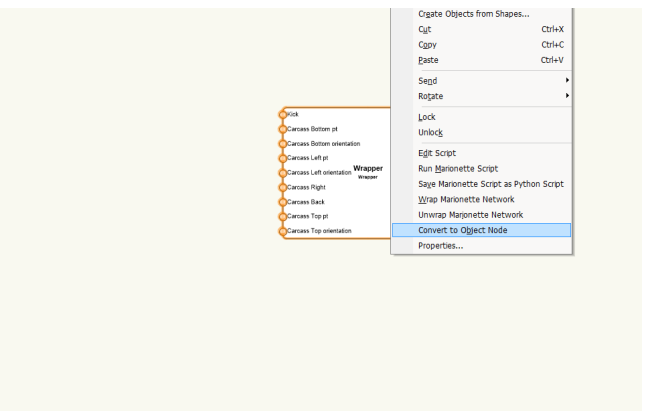
- A wrapped marionette networkwill appear with a different colour and it's name will be presented as Wrapper.



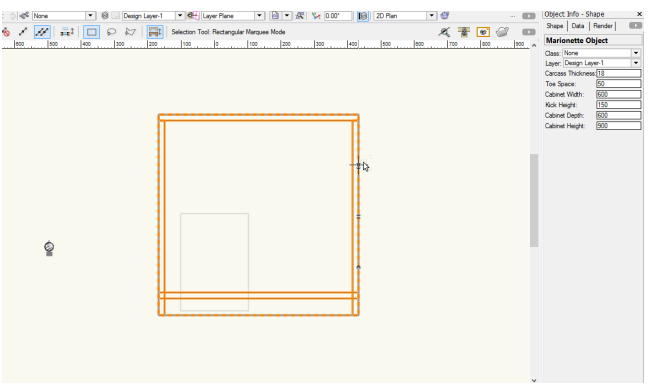
Converting a Wrapper to an Object

Once you have created a marionette wrapper you can convert it to a Marionette Object. The great thing about a marionette object is that it appears on the Object Info palette like a plug and object.

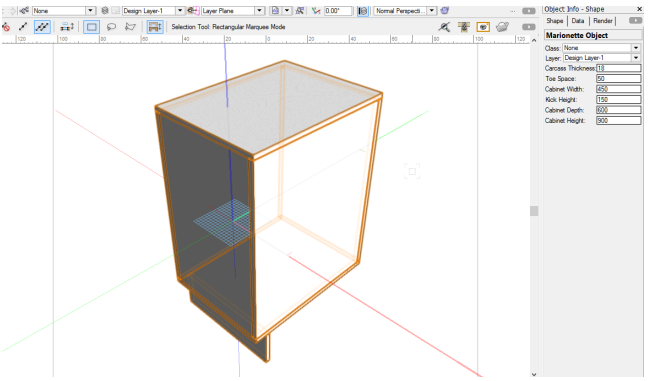
- Right click on the marionette wrapper.
- Choose Convert To Object Node from the contextual menu.



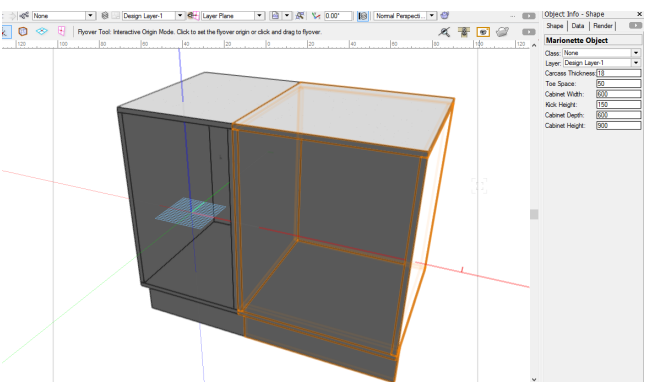
- This is now a marionette object. All of the named inputs will appear on the Object Info palette. Any inputs that haven't been named specifically will not appear.



- This is a fantastic use for marionette, the ability to create your own objects that are still parametric (can be changed on the Object Info palette.



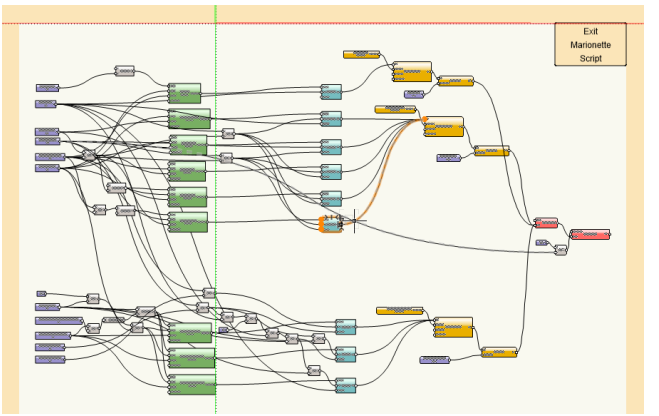
- When you duplicate one of your marionette objects, you can adjust its settings separate from any of the other objects in the drawing. This gives you the ability to use the same marionette object to create different sized cabinets.



- In this image you can see that I've taken the cabinet I have added doors and drawers to them. You can also add marionettes nodes that will control the textures on the carcass and drawers/doors separately.

- When you double-click on a marionette object, you will see the marionette network that was used to create it. You can edit this network to make changes to your object.

- You can use this technique to duplicate marionette objects then edit the network to change it. This is how I created a three drawer, for drawer, two-door, and single door cabinet.



What's Next?

This manual has only been an introduction to marionette. I wanted to make sure that I could take away the mystery of how to create a marionette network and how to make a simple object.

There are other resources that you can use to build on your marionette skill. Vectorworks have created a special area in the Vectorworks community board for you to ask marionette questions and where you can find shared marionette objects.

I believe the ability to take a shared marionette object and then edited to suit your requirements is a huge benefit in marionette.

You can find the Vectorworks community board at this link:

https://techboard.vectorworks.net/ubbthreads.php?ubb=cfrm&c=10

Thank you

We trust that you have enjoyed working through this manual and that it has been informative and constructive.

For more information, please visit: http://www.archoncad.co.nz/. If you just want someone to help you learn Vectorworks, to carry out some Vectorworks contract work, or you want someone to make Vectorworks easier, contact us, as this is a service that we also offer:

jon@archoncad.com

Thank you again,
Jonathan Pickup
November 2015