

MFG219625

# Designing Adaptive Inventor Sheet Metal Parts and Assemblies with iLogic

Scott Hallmark  
Physical Security, LLC

## Learning Objectives

- Discover the Skeletal Solid
- Learn how to apply the sheet metal parts to the skeleton using adaptivity
- Learn how to create the iLogic rules and form
- Learn how to drive the assembly

## Description

Come to this class to learn how to create a simple sheet metal hopper using iLogic and adaptivity in Inventor software

## Speaker(s)

Scott Hallmark is a Mechanical Designer for Physical Security, LLC ([www.physec.com](http://www.physec.com)) in Bessemer, AL. He had previously been in the Autodesk Reseller Channel since 2007 specializing in Autodesk Inventor, Factory Design, AutoCAD Plant 3D and P&ID, as well as Autodesk ReCap and Configurator 360. Scott has been selected 7 times for speaking at Autodesk University, 6 of those primarily for AutoCAD Plant 3D. Scott is the developer of Steel Supports Bundle for Plant 3D currently being distributed by Applied Software ([www.asti.com/SteelSupports](http://www.asti.com/SteelSupports)). Scott is a Certified Inventor Professional, AutoCAD Professional, and Certified Autodesk Instructor



@scottscs71



[www.linkedin.com/in/scotthallmark/](http://www.linkedin.com/in/scotthallmark/)



[www.facebook.com/PhySec/](http://www.facebook.com/PhySec/)



PHYSICAL SECURITY, L.L.C.



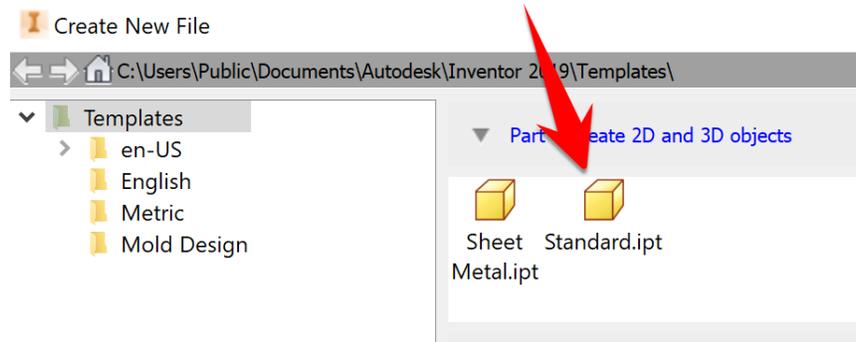
## Discover the Skeletal Solid

Using Adaptivity and iLogic with Sheet Metal is a simple process, probably simpler than you thought it could be. Usually when we talk about skeletal modeling, we tend to think about sketches with lines, arcs and circles, and not solids. However, in this example, it is imperative that we use a skeletal solid to generate the Sheet Metal parts. We will eventually incorporate iLogic into this model but we need to plan that ahead of time as you will see in the following paragraphs.

We will start by creating the volume of the hopper as a solid. Our Sheet Metal parts will be built on the outside of this solid.

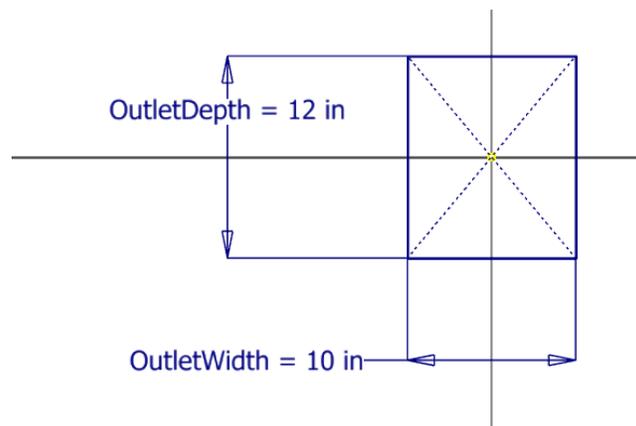
## Use a Standard Part Template

Although we are generating an assembly of sheet metal parts, the main skeletal solid will be created using a standard part template.



## Create the Sketches to Define the Hopper

Define your sketches with iLogic in mind, so in other words, give descriptive names to your parameters. In this example, use *OutletDepth* and *OutletWidth* instead of *d0* and *d1*. Continue this process until you have all of the sketches created to build the solid.



### Keeping iLogic in mind...

So as I stated above, develop your sketches with iLogic in mind. Even when creating the extrusions, use named parameters so that we can utilize them in the assembly iLogic code a bit later on. The following image is of the Parameters dialog box after I have created the skeletal solid. There are basically two methods of doing this and both work and it really becomes a matter of preference for you. In this example, I did not create all of the user parameter names ahead of time and then plug those into the dimension values. Instead, I just named them as I created the dimension. Again, either way works so do what feels right to you.

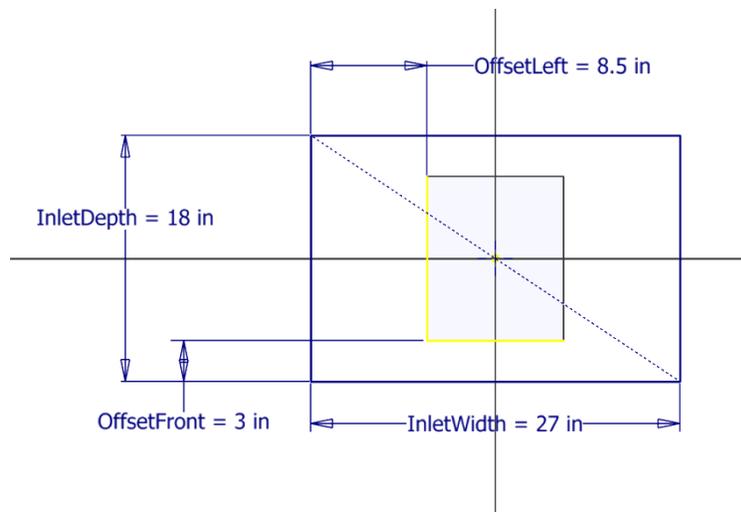
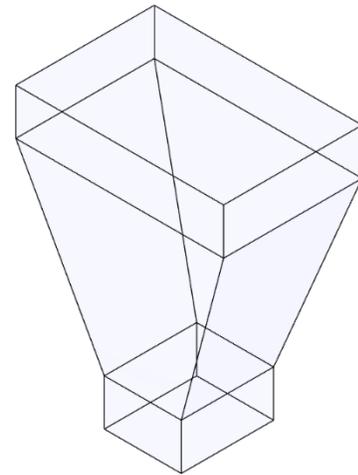
**Shortcut** - you could also create these as user parameters in a template so that they already exist.

Parameters

Parameter Name	Consumed by	Unit/Type	Equation	Nominal Value
<b>Model Parameters</b>				
OutletWidth	Sketch1	in	10 in	10.000000
OutletDepth	Sketch1	in	12 in	12.000000
OutletHeight	Extrusion1	in	6 in	6.000000
d3	Extrusion1	deg	0.0 deg	0.000000
MidsectionHeight	Work Plane1	in	24 in	24.000000
InletWidth	Sketch2	in	27 in	27.000000
InletDepth	Sketch2	in	18 in	18.000000
d7	Loft1	ul	0 ul	0.000000
d8	Loft1	deg	90 deg	90.000000
d9	Loft1	ul	0 ul	0.000000
d10	Loft1	deg	90 deg	90.000000
InletHeight	Extrusion2	in	6 in	6.000000
d12	Extrusion2	deg	0.0 deg	0.000000
OffsetLeft	Sketch2	in	8.5 in	8.500000
OffsetFront	Sketch2	in	3 in	3.000000
<b>User Parameters</b>				
OffCenterHopper		True/False	False	

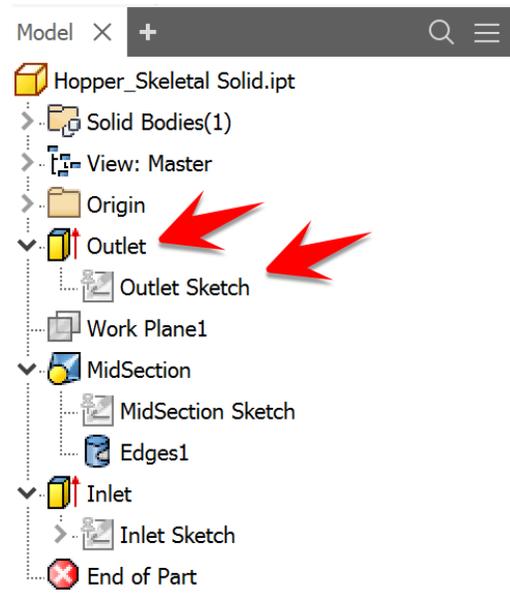
### Completing the Skeletal Solid

Using the Parameters above (Parameter names and solids created) and the picture below, you will see what we plan to end up with for our Skeletal Solid. I have used a “Clear - Blue” Color for this so that I can see thru the part when creating the Sheet Metal later on.



I am placing this in here for now, so you can place dimensions, but we will circle back to this, so you can see how we are going to be able to create and offset in the Inlet and Outlet centers of the Hopper.

You may want to consider naming the Features also. This is helpful not only in iLogic but for anyone that has to use your models later on. The more descriptive you can be in naming, the better your model will be overall.

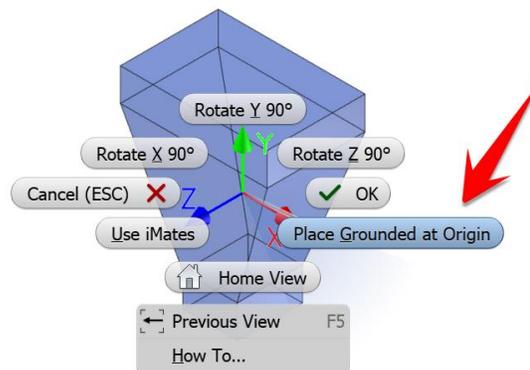


## Learn how to apply the sheet metal parts to the skeleton using adaptivity

This is where the fun begins! This process is so simple that if you have not thought about doing this before, you will wonder “why I have I not thought of doing this before?” We will begin using the outside of the Skeletal Model to create our Sheet Metal parts and make them Adaptive. When design intent requires a change in the skeletal model, it will adaptively update the sheet metal. Be careful to follow the instructions because if you leave out this one important step, it will not work. That step will be obvious just below in the instructions.

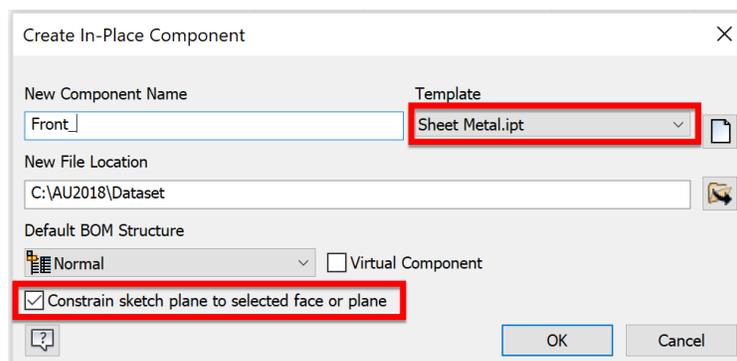
### Place the Skeletal Solid in the Assembly

In order to create the sheet metal parts, we have to place the skeletal solid in the assembly. It is best practice to insert the skeletal solid grounded at the origin.

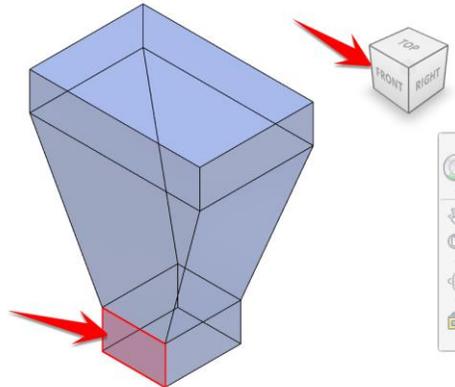


### Create the Sheet Metal Part while in the Assembly

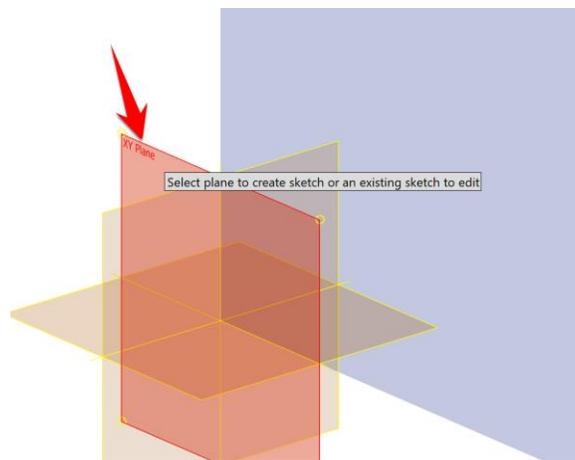
By creating the sheet metal parts in the assembly using the Skeletal solid, this will enable adaptivity automatically. You will see the hurricane symbol (as I like to call it) next to the sketch, features and part in the assembly. Be sure to select the correct template and the constrain option should be selected.



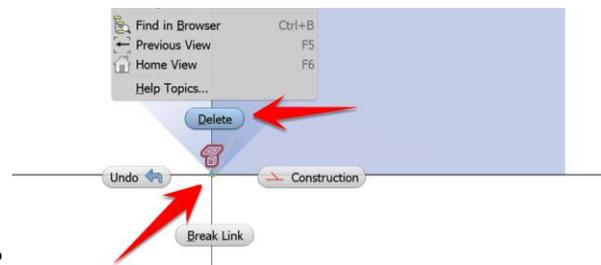
Then select the Front Outlet Face of the Hopper.



Create a Sketch and select the default work plane in the same plane as the Front Outlet face.

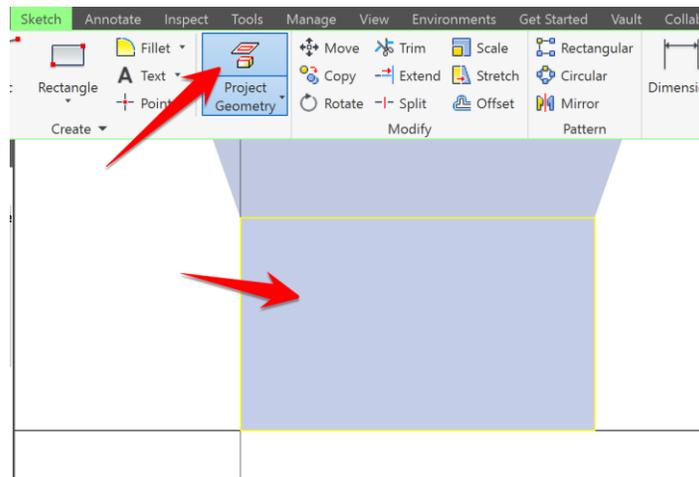


**!!! Here is the key item to remember to do !!!** If you have this option turned on in Applications Options > Sketch, then you either need to turn it OFF before creating these parts, OR you can just delete the projected origin point before you go to the next step. The reason for this is that the projected origin will never change and if your projected outlet geometry is constrained to that point, it will skew your sheet metal part and not update correctly when you modify the skeletal solid.

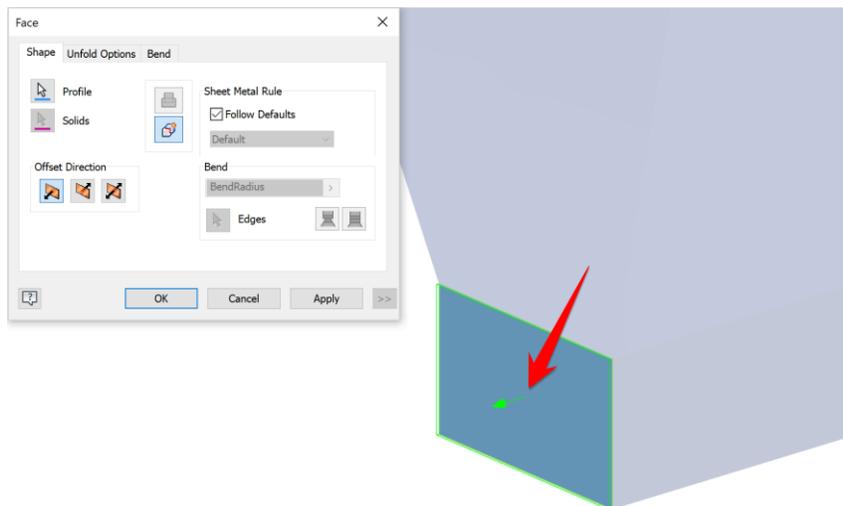


OR

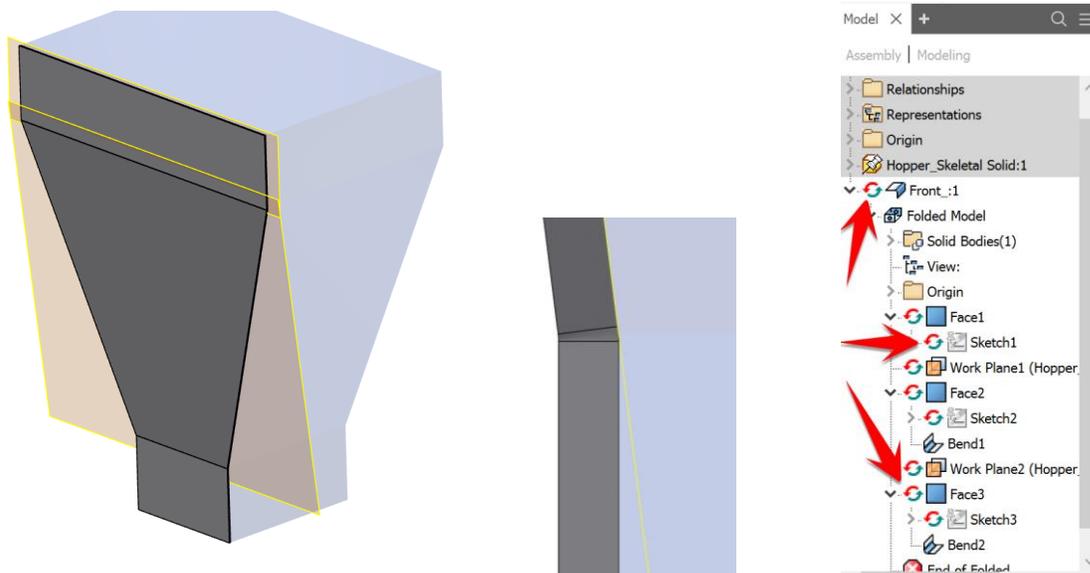
While in Sketch Mode, select Project Geometry and select the face of the Front Outlet of the Skeletal Solid.



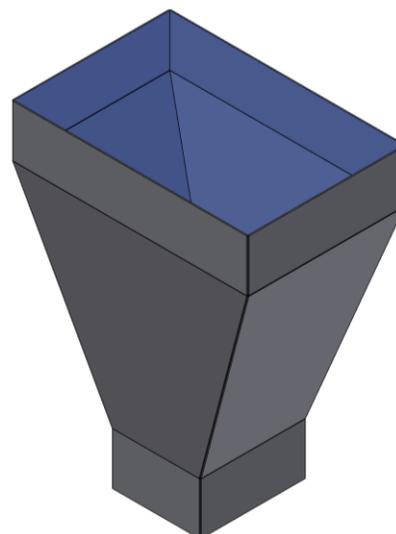
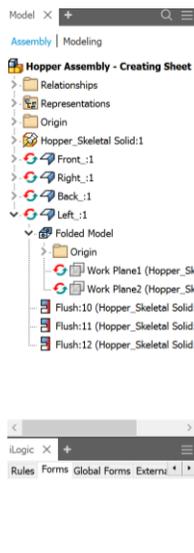
Finish the Sketch and create the Face from the projected geometry. Be sure the face is to the outside of the skeletal solid because the solid represents the volume of the hopper. Also, the four sheet metal sides of the hopper will clash if they are generated to the inside.



Repeat this process until all of the Front Sides are created in the same FRONT.IPT Sheet Metal Part. Bends are automatically created and notice the browser for the Adaptivity Symbols. Work planes are automatically created and left visible. If you turn off visibility while in “edit” mode in the parts, they are not permanently turned off, as you may have already figured out. You need to “open” the parts in their own window to turn off visibility permanently.



Now, repeat the process for the other 3 sides of the hopper. You should end up with 5 files in the assembly: 1 Skeletal Solid and 4 Sheet Metal Parts. Be sure to set the Skeletal Solid to Reference so it does not show up in the BOM. Feel free to add other features to these parts like flanges or cuts, all of which could end up being driven from and iLogic Form.



## Learn how to create the iLogic rules and form

We will start off simple with this example and just map the required parameters from the Assembly to the Skeletal Solid. This will allow us to drive the Skeletal parameters with Assembly parameters so that we do not have to open the Skeletal model to drive the assembly. Below is an example of the code needed to map the parameters. In order for this to work, the parameters have to exist in both the Skeletal model and the assembly.

```
'Mapping Assembly Parameters to Skeletal Model Parameters.
Parameter("Hopper_Skeletal Solid:1", "OutletDepth") = OutletDepth
Parameter("Hopper_Skeletal Solid:1", "OutletWidth") = OutletWidth
Parameter("Hopper_Skeletal Solid:1", "OutletHeight") = OutletHeight
Parameter("Hopper_Skeletal Solid:1", "MidsectionHeight") = MidsectionHeight
Parameter("Hopper_Skeletal Solid:1", "InletDepth") = InletDepth
Parameter("Hopper_Skeletal Solid:1", "InletWidth") = InletWidth
Parameter("Hopper_Skeletal Solid:1", "InletHeight") = InletHeight
Parameter("Hopper_Skeletal Solid:1", "OffsetLeft") = OffsetLeft
Parameter("Hopper_Skeletal Solid:1", "OffsetFront") = OffsetFront
Parameter("Hopper_Skeletal Solid:1", "OffCenterHopper") = OffCenterHopper
```

### Assembly:

User Parameters					
	InletDepth		in	36 in	36.000000
	InletWidth		in	36 in	36.000000
	InletHeight		in	6 in	6.000000
	MidsectionHeight		in	30 in	30.000000
	OutletDepth		in	16 in	16.000000
	OutletWidth		in	16 in	16.000000
	OutletHeight		in	6 in	6.000000
	OffsetLeft		in	13 in	13.000000
	OffsetFront		in	12 in	12.000000
	OffCenterHopper		True/False	False	
	FlangeWidth		in	2 in	2.000000

Skeletal Solid:

Model Parameters					
	OutletWidth	Outlet Sketch	in	16 in	16.000000
ersv	OutletDepth	Outlet Sketch	in	16 in	16.000000
	OutletHeight	Outlet	in	6 in	6.000000
nc	d3	Outlet	deg	0.0 deg	0.000000
Vx/E	MidsectionHeight	Work Plane1	in	30 in	30.000000
7	InletWidth	MidSection Sketch	in	36 in	36.000000
ersv	InletDepth	MidSection Sketch	in	36 in	36.000000
	d7	MidSection	ul	0 ul	0.000000
nc	d8	MidSection	deg	90 deg	90.000000
Vx/E	d9	MidSection	ul	0 ul	0.000000
7	d10	MidSection	deg	90 deg	90.000000
ersv	InletHeight	Inlet	in	6 in	6.000000
	d12	Inlet	deg	0.0 deg	0.000000
nc	OffsetLeft	MidSection Sketch	in	10 in	10.000000
Vx/E	OffsetFront	MidSection Sketch	in	10 in	10.000000
	<b>User Parameters</b>				
	OffCenterHopper		True/False	False	

### More Coding Examples – MidOpening with Flanges

Maybe you want to add an opening in the Front side of the hopper that consist of a CUT and 2 FLANGES. We want to be able to toggle them ON/OFF as needed. The following code uses IF/THEN statements to allow us to do this. Again, keeping this very simple for beginning iLogic usage, we will spell out everything and not get into complicated coding. This uses and IF statement but with an AND condition stating that IF 2 different Boolean parameters are set to either TRUE or FALSE, or some combination of the 2, THEN make 2 different features set to TRUE or FALSE. Read the code and it will make sense.

```
'MidOpening Settings
If MidOpening = True And MidOpeningFlanges = True Then
Feature.IsActive("front:1", "Cut1") = True
Feature.IsActive("front:1", "Flange2") = True
Else If MidOpening = False Then
Feature.IsActive("front:1", "Cut1") = False
Feature.IsActive("front:1", "Flange2") = False
Else If MidOpening = True And MidOpeningFlanges = False Then
Feature.IsActive("front:1", "Cut1") = True
Feature.IsActive("front:1", "Flange2") = False
End If
```

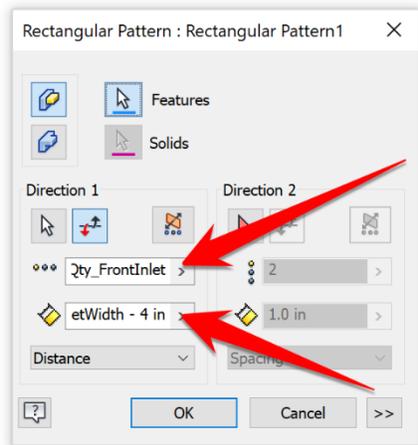
### More Coding Examples – Holes in Inlet/Outlet Flanges

We can add holes in the Sheet Metal Parts and create equations at the part level to drive the number and spacing, but we do NOT want to have to go into each part to drive those holes. We want to do it from the Assembly level. A simple parameter mapping code like below will handle this operation. “front:1” is the part:occurrence. You would do this for all of the parts but the code exist in the Assembly rule.

```
'Set Flange Widths on all SM parts
Parameter("front:1", "FlangeWidth") = FlangeWidth
Parameter("right:1", "FlangeWidth") = FlangeWidth
Parameter("back:1", "FlangeWidth") = FlangeWidth
Parameter("left:1", "FlangeWidth") = FlangeWidth

'Set Holes for "front" Sheetmetal part
Parameter("front:1", "InletWidth") = InletWidth
Parameter("front:1", "HoleQty_FrontInlet") = HoleQty_FrontInlet
Parameter("front:1", "OutletWidth") = OutletWidth
Parameter("front:1", "HoleQty_FrontOutlet") = HoleQty_FrontOutlet
```

The equation at the part level to drive the holes is listed below:



*HoleQty\_FrontInlet*

This is just parameter that we will drive from the assembly

*InletWidth - 4 in*

This is a parameter minus 4” (2” on each side of Flange) and the Type is set to DISTANCE. This ensures we will get evenly spaced holes based on the quantity we defined

above, while still leaving 2” from end of Flange to 1<sup>st</sup> and last holes.

### More Coding Examples – Offset the Hopper

This example of code is actually existing in the Skeletal model instead of the hopper assembly. I set a True/False option in the assembly to either offset or center up the hopper inlet/outlet.

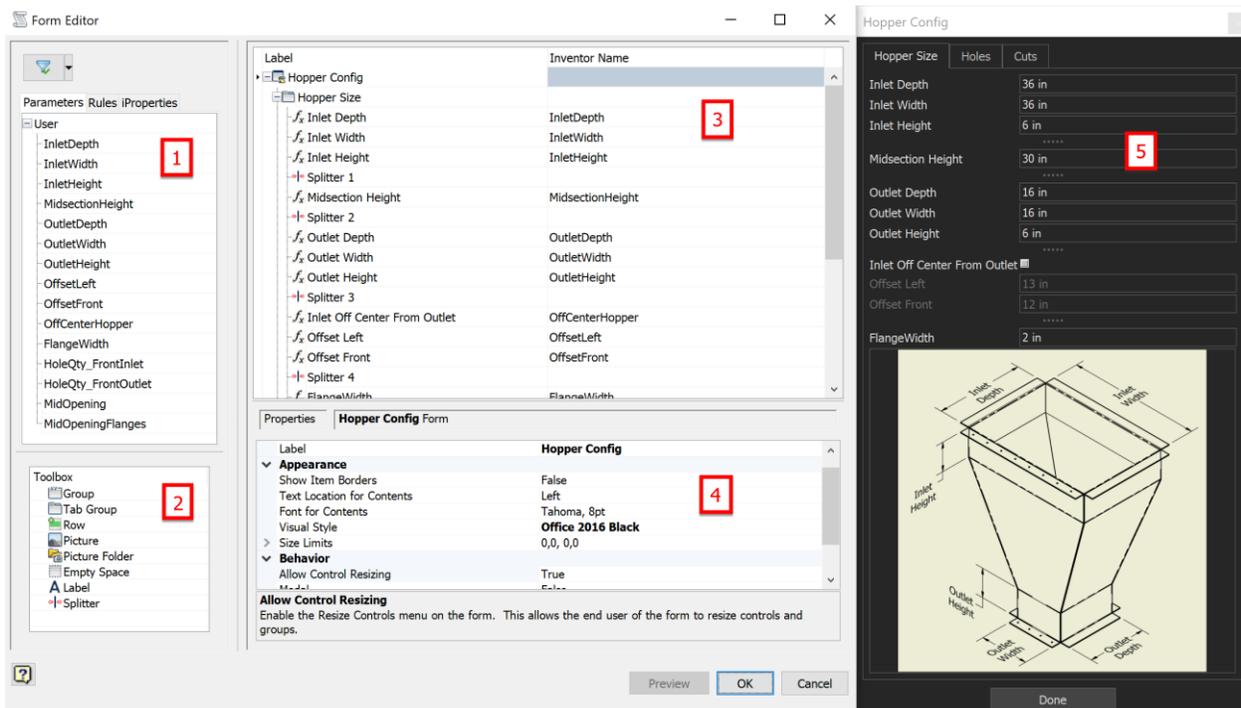
```
If OffCenterHopper = "True" Then
    OffsetLeft = OffsetLeft
    OffsetFront = OffsetFront
Else If OffCenterHopper = "False" Then
    OffsetLeft = (InletWidth / 2) - (OutletWidth / 2)
    OffsetFront = (InletDepth / 2) - (OutletDepth / 2)
End If
```

## Learn how to drive the assembly

Now we just need to build a form that will put these rules in action. We will use the iLogic Forms builder to do this. These forms can be overly elaborate or painfully plain. It really depends on your needs and the time you have to develop.

### What is in the form builder?

1. Parameters, Rules and iProperties – This area allows you to use the named parameters you used in the parent file that you are creating the rule in, as well as run other rules in the file and set various iProperties from this custom form you are creating.
2. The Toolbox – This area is a set of options to organize the form with groups, pictures, labels and splitters.
3. Form Contents – This is the list of parameters, toolbox items, etc that make up the form contents and the structure you have defined.
4. Properties – Each item you place into the Form Contents has a set of properties associated to it. Use this area to define text formats, image sizes, toggle options, etc.
5. Form Preview – This area updates as you build your custom iLogic Form making sure you include all of items that you want to see and making sure they are organized how you want to be able to drive the assembly.



After you build your custom iLogic Form, you will want to test, test and test again. If you include several toggle options or dropdown values, you will want to test ALL scenarios to make sure the model will not “blow up” on you. Some of you just silently laughed a little because you know what I am talking about and have seen this first-hand. Test it!

Hopefully, you will be able to put these techniques into practice at your place of business and you can harness the power of using Adaptivity and iLogic together to automate your designs. Feel free to access the reference material below, as they all have become invaluable assets to my personal collection of links when I am developing iLogic code.

### **Additional AU Reference Material for Consideration (in no particular order)**

MA1936: Advanced iLogic: Code Encapsulation (Chad Salmela, AU 2013)

[http://au.autodesk.com/au-online/classes-on-demand/class-catalog/2013/product-design-suite/ma1936?\\_ga=2.251379910.1589372453.1539965843-658101757.1532377382#chapter=0](http://au.autodesk.com/au-online/classes-on-demand/class-catalog/2013/product-design-suite/ma1936?_ga=2.251379910.1589372453.1539965843-658101757.1532377382#chapter=0)

IM124871: Inventor iLogic: Beyond the Basics (Jon Balgley, AU 2017)

<http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2017/inventor-professional/im124871#chapter=0>

DV1673-L: The Decoding Lab: iLogic Intermediate Session (Steve Olson, AU 2013)

<http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2013/product-design-suite/dv1673-l>

IM120928: Adding Intellect to Inventor with Intelligent Modeling (Jason Hunt, AU 2017)

<http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2017/inventor-professional/im120928>

CP2104: iLogic for Everyone: Five Autodesk® Inventor® iLogic Rules Everyone Can Use (Steve Olson, AU 2012)

<http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2012/autodesk-inventor-products/illogic-for-everyone-five-autodesk-inventor-illogic-rules-everyone-can-use#chapter=0>

MD6936: The Logical Choice: Creating iLogic Templates and iLogic Title Block Automation (Mark Flayler, AU 2017)

<http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2014/inventor/md6936#chapter=0>

SD17917: iLogic and the Inventor API (Brian Ekins, AU 2016)

<http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2016/inventor-professional/sd17917#chapter=0>